Lenovo Network

Application Guide

for Lenovo Cloud Network Operating System 10.6

Lenovo.™

Note: Before using this information and the product it supports, read the general information in the *Safety information and Environmental Notices* and *User Guide* documents on the Lenovo *Documentation* CD, and the *Warranty Information* document that comes with the product.

Contents

Preface

This *Application Guide* describes how to configure and use the Lenovo Cloud Network Operating System 10.6 software on the following Lenovo RackSwitches:

- Lenovo RackSwitch G8272. For documentation on installing the switch physically, see the *Lenovo RackSwitch G8272 Installation Guide*.

- Lenovo RackSwitch G8296. For documentation on installing the switch physically, see the *Lenovo RackSwitch G8296 Installation Guide*.

- Lenovo RackSwitch G8332. For documentation on installing the switch physically, see the *Lenovo RackSwitch G8332 Installation Guide*.

- Lenovo ThinkSystem NE1032T RackSwitch. For documentation on installing the switch physically, see the *Lenovo ThinkSystem NE1032T RackSwitch Installation Guide*.

- Lenovo ThinkSystem NE1032 RackSwitch. For documentation on installing the switch physically, see the *Lenovo ThinkSystem NE1032 RackSwitch Installation Guide*.

- Lenovo ThinkSystem NE1072T RackSwitch. For documentation on installing the switch physically, see the *Lenovo ThinkSystem NE1072T RackSwitch Installation Guide*.

- Lenovo ThinkSystem NE10032 RackSwitch. For documentation on installing the switch physically, see the *Lenovo ThinkSystem NE10032 RackSwitch Installation Guide*.

- Lenovo ThinkSystem NE2572 RackSwitch. For documentation on installing the switch physically, see the *Lenovo ThinkSystem NE2572 RackSwitch Installation Guide*.

Who Should Use This Guide

This guide is intended for network installers and system administrators engaged in configuring and maintaining a network. The administrator should be familiar with Ethernet concepts, IP addressing, Spanning Tree Protocol, and SNMP configuration parameters.

Application Guide Overview

This guide will help you plan, implement, and administer the Cloud NOS (CNOS) software. Where possible, each section provides feature overviews, usage examples, and configuration instructions. The following material is included:

Part 1: Getting Started

This material is intended to help those new to CNOS products with the basics of switch management. This part includes the following chapters:

- Chapter 1, "Using the Command Line Interface", describes the CNOS command-line interface modes, commands, keyboard shortcuts, and aliases.

- Chapter 2, "Switch Administration", describes how to access the switch to configure the switch, and view switch information and statistics. This chapter discusses a variety of manual administration interfaces, including local management via the switch console, and remote administration via Telnet or Secure Shell.

- Chapter 3, "System License Keys", describes how to install additional features on the switch.

- Chapter 4, "Switch Software Management," describes how to update the CNOS software operating on the switch and how to convert from CNOS to ENOS.

Part 2: Securing the Switch

This material contains information about implementing security protocols on the switch. This part includes the following chapters:

- Chapter 5, "Securing Administration," describes methods for using Secure Shell for administration connections, and configuring end-user access control.

- Chapter 6, "AAA Protocols," describes different secure administration methods for remote administrators. This includes using RADIUS, Terminal Access Controller Access-Control System Plus (TACACS+) and Authentication, Authorization, and Accounting (AAA).

- Chapter 7, "Access Control Lists," describes how to use filters to permit or deny specific types of traffic, based on a variety of source, destination, and packet attributes.

Part 3: Switch Basics

This material contains information about setting up features on the switch. This part includes the following chapters:

- Chapter 8, "Interface Management", describes how to configure the switch interfaces, like the ethernet or management ports.

- Chapter 9, "Forwarding Database", describes how a Layer 2 device can be configured to learn and store MAC addresses and their corresponding ports.

- Chapter 10, "VLANs", describes how to configure Virtual Local Area Networks (VLANs) for creating separate network segments, including how to use VLAN tagging for devices that use multiple VLANs.

- Chapter 11, "Ports and Link Aggregation", describes how to group multiple physical ports together to aggregate the bandwidth between large-scale network devices.

- Chapter 12, "Spanning Tree Protocol", describes how to use the Rapid Per VLAN Spanning Tree Plus (Rapid PVST+) and Multiple Spanning Tree Protocol (MSTP) to build a loop-free network topology.

- Chapter 13, "Virtual Link Aggregation Groups", describes using Virtual Link Aggregation Groups (VLAGs) to form LAGs spanning multiple VLAG-capable aggregator switches.

- Chapter 14, "Quality of Service", discusses Quality of Service (QoS) features, including IP filtering using class maps, Differentiated Services, and IEEE 802.1p priority values.

- Chapter 15, "CEE", discusses using various Converged Enhanced Ethernet (CEE) features such as Priority-based Flow Control (PFC), Enhanced Transmission Selection (ETS) and Data Center Bridging Capability Exchange (DCBX).

- Chapter 16, "Secure Mode", describes the difference between secure mode and legacy mode, what enabling secure mode means, and how to enable and disable it.

Part 4: IP Routing

This part includes the following chapters:

- Chapter 17, "Basic IP Routing", describes how to configure the switch for IP routing using IP subnets, BFD, DHCP Relay and VRF.

- Chapter 18, "Routed Ports", describes how to configure a switch port to forward Layer 3 traffic.

- Chapter 19, "Address Resolution Protocol", describes how to use the Address Resolution Protocol (ARP) protocol to map an IPv4 address to a MAC address.

- Chapter 20, "Internet Protocol Version 6", describes how to configure the switch to use IPv6.

- Chapter 21, "Internet Group Management Protocol", describes how CNOS implements Internet Group Management Protocol (IGMP) Snooping to conserve bandwidth in a multicast-switching environment.

- Chapter 22, "Border Gateway Protocol", describes Border Gateway Protocol (BGP) concepts and features supported in CNOS.

- Chapter 23, "Open Shortest Path First", describes key Open Shortest Path First (OSPF) concepts, and how they are implemented in CNOS, and provides examples of how to configure your switch for OSPF support.

- Chapter 24, "Route Maps", describes route maps that are used to define route policy by permitting or denying certain routes based on a configured set of rules.

Part 5: High Availability Fundamentals

This part includes the following chapters:

- Chapter 25, "Basic Redundancy", describes how the switch supports redundancy through LAGs and VLAGs.

- Chapter 26, "Virtual Router Redundancy Protocol", describes how the switch supports high-availability network topologies using Virtual Router Redundancy Protocol (VRRP).

- Chapter 27, "Layer 2 Failover", describes how to configure and use network adapter teaming for Layer 2 LAG failover.

Part 6: Network Management

This part includes the following chapters:

- Chapter 28, "Link Layer Discovery Protocol", describes how Link Layer Discovery Protocol (LLDP) helps neighboring network devices learn about each others' ports and capabilities.

- Chapter 29, "Service Location Protocol", describes the Service Location Protocol (SLP) that allows the switch to provide dynamic directory services.

- Chapter 30, "Simple Network Management Protocol", describes how to configure the switch for management through a Simple Network Management Protocol (SNMP) client.

- Chapter 31, "Telemetry", describes the CNOS Network Telemetry Agent and how to use the data it provides to fine-tune your network.

Part 7: Hyperconverged Infrastructure

This part includes the following chapters:

- Chapter 32, "Network Virtualization Gateway", describes how to integrate VMware NSX with your switch.

- Chapter 33, "Network Policy Agent", explains how to use the CNOS network policy agent plugin that works with the Nutanix Virtual Domain Module.

Part 8: Monitoring

This part includes the following chapters:

- Chapter 34, "Port Mirroring", discusses tools to copy selected port traffic to a remote monitor port for network analysis.

- Chapter 35, "Sampled Flow", discusses using Sampled Flow (sFlow) for monitoring traffic.

Part 9: Appendices

This part includes the following appendices:

- Appendix A, "Getting help and technical assistance", provides details on where to go for additional information about Lenovo and Lenovo products.

- Appendix B, "Notices", contains safety and environmental notices.

Additional References

Additional information about installing and configuring your switch is available in the following guides:

- *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*

- *Lenovo Network Release Notes for Lenovo Cloud Network Operating System 10.6* for your switch

- *Lenovo Network Python Programming Guide for Lenovo Cloud Network Operating System 10.6*

- *Lenovo Network REST API Programming Guide for Lenovo Cloud Network Operating System 10.6*

Typographic Conventions

The following table describes the typographic styles used in this book.

Table 1. *Typographic Conventions*

Typeface or Symbol	Meaning	Example
ABC123	This type is used for names of commands, files, and directories used within the text. It also depicts on-screen computer output and prompts.	View the readme.txt file. Switch#
ABC123	This bold type appears in command examples. It shows text that must be typed in exactly as shown.	Switch# **ping**
\<ABC123\>	This italicized type appears in command examples as a parameter placeholder. Replace the indicated text with the appropriate real name or value when using the command. Do not type the brackets. This also shows book titles, special terms, or words to be emphasized.	To establish a Telnet session, enter: Switch# **telnet** *\<IP address\>* Read your *User's Guide* thoroughly.
{}	Command items shown inside brackets are mandatory and cannot be excluded. Do not type the brackets.	Switch# **cp {ftp|sftp}**
[]	Command items shown inside brackets are optional and can be used or excluded as the situation demands. Do not type the brackets.	Switch# **configure [device]**
|	The vertical bar (**|**) is used in command examples to separate choices where multiple options exist. Select only one of the listed options. Do not type the vertical bar.	Switch# **cp {ftp|sftp}**
\<AaBb123\>	This block type depicts menus, buttons, and other controls that appear in graphical interfaces.	Click the **\<Save\>** button.

Part 1: Getting Started

This section discusses the following topics:

-
-
-

Chapter 1. Using the Command Line Interface

Lenovo Cloud Network Operating System uses an industry-standard command line interface (CLI). Like any switch CLI, there are subtle differences between the CNOS CLI and the CLI on switches from other vendors.

The following subjects are discussed in this chapter:

- "CLI Command Modes" on page 34
- "Command Line Interface Shortcuts" on page 35
- "Command Aliases" on page 37

CLI Command Modes

The CLI has three major command modes listed in order of increasing privileges, as follows:

● User EXEC Mode: `Switch>`

This is the initial mode of access. By default, on console sessions password checking is disabled for this mode.

● Privileged EXEC mode: `Switch#`

This mode is accessed from User EXEC Mode. This mode can be accessed using the following command: **enable**

● Configuration Mode: `Switch(config)#`

This mode allows you to make changes to the running configuration. If you save the configuration, the settings survive a reload of the switch. Several sub-modes can be accessed from the User EXEC Mode. This mode can be accessed using the following command: **configure [device]**

Each mode provides a specific set of commands. Most lower-privilege mode commands are accessible when using a higher-privilege mode.

Note: The word "`Switch`" is a generic term used throughout the *Application Guide* to indicate the hostname of the switch when issuing commands. Depending on the Lenovo RachSwitch or ThinkSystem, the word "`Switch`" will be replaced with one of the following:

Switch Type	Prompt
RackSwitch G8272	`G8272`
RackSwitch G8296	`G8296`
RackSwitch G8332	`G8332`
ThinkSystem NE1032 RackSwitch	`NE1032`
ThinkSystem NE1032T RackSwitch	`NE1032T`
ThinkSystem NE1072T RackSwitch	`NE1072T`
ThinkSystem NE10032 RackSwitch	`NE10032`
ThinkSystem NE2572 RackSwitch	`NE2572`

Command Line Interface Shortcuts

The following shortcuts allow you to enter commands quickly and easily.

CLI List and Range Inputs

For VLAN and port commands that allow an individual item to be selected from within a numeric range, lists and ranges of items can now be specified. For example, the vlan command permits the following options:

```
Switch(config)# vlan 1,3,1094           (access VLANs 1, 3, and 1094)
Switch(config)# vlan 1-20               (access VLANs 1 through 20)
Switch(config)# vlan 1-5,90-99,1090-1094  (access multiple ranges)
Switch(config)# vlan 1-5,19,20,1090-1094  (access a mix of lists and ranges)
```

The numbers in a range must be separated by a dash: *<start of range>-<end of range>*

Multiple ranges or items are permitted using a comma: *<range or item 1>,<range or item 2>*

Do not use spaces within list and range specifications.

Ranges can also be used to apply the same command option to multiple items. For example, to access multiple ports with one command:

```
Switch(config)# spanning-tree mst 1-4 cost 4096      (instances 1 through 4)
```

Command Abbreviation

Most commands can be abbreviated by entering the first characters which distinguish the command from the others in the same mode. For example, consider the following full command:

```
Switch(config)# display mac address-table interface ethernet 1/12
```

It can be abbreviated as follows:

```
Switch(config)# disp ma ad i e 1/12
```

Tab Completion

By entering the first letter of a command at any prompt and pressing **<Tab>**, the ISCLI displays all available commands or options that begin with that letter. Entering additional letters further refines the list of commands or options displayed. If only one command fits the input text when **<Tab>** is pressed, that command is supplied on the command line, waiting to be entered.

If multiple commands share the typed characters, when you press **<Tab>**, the ISCLI completes the common part of the shared syntax.

Line Editing

The following case-insensitive keystroke commands are available for editing command lines:

Command	Behavior
<Ctrl + A>	Moves the cursor to the beginning of the line.
<Ctrl + B>	Moves the cursor one character to the left.
<Ctrl + D>	Deletes the character at the cursor.
<Ctrl + E>	Moves the cursor to the end of the line.
<Ctrl + F>	Moves the cursor one character to the right.
<Ctrl + K>	"Kills" all text to the right of the cursor, putting it into a buffer.
<Ctrl + L>	Clears the screen, leaving the current line intact at the top.
<Ctrl + N>	Move to the next command in the command history.
<Ctrl + P>	Move to the previous command in the command history.
<Ctrl + T>	Swaps the character at the cursor with the character to the left of the cursor.
<Ctrl + U>	Clears all text from the command line.
<Ctrl + W>	Deletes from the cursor to the start of the "word."
<Ctrl + Y>	"Yanks" the text from the kill buffer.
<Esc + B>	Moves the cursor backwards one "word."
<Esc + C>	Capitalizes the first letter of the "word" or the character where the cursor is pointing.
<Esc + D>	Deletes to the end of the word to the right of the cursor.
<Esc + F>	Moves the cursor forwards one "word."
<Esc + L>	Changes the text to lowercase from the cursor to the end of the "word."
<Esc + U>	Changes the text to uppercase from the cursor to the end of the "word."

Command Aliases

Command aliasing enables you to change the names of commands in the CLI.

Defining Aliases

To define an alias, enter:

```
Switch(config)# alias <command alias> <current command>
```

For example, to use the command show to invoke the display command, enter:

```
Switch(config)# alias show display
```

Removing Aliases

To remove an alias, enter:

```
Switch(config)# no alias <command alias>
```

To remove all aliases, enter:

```
Switch(config)# no alias all
```

Displaying Aliases

To see the list of aliases configured to your system, enter:

```
Switch(config)# display alias
CLI alias information:
====================
show  : display
abc   : display
```

Note: The alias command does not do validation checking. If you enter an invalid command for an alias to invoke, you will get an error message.

Rules for Using Aliases

The following rules apply when you are defining an alias:

- An alias must be an alphanumeric string that starts with an alphabetic character. There can be no spaces or punctuation characters in an alias name. There *can* be dashes and spaces in the *command* being aliased. For example, the following command aliases the string dsi to display sys-info:

```
Switch(config)# alias dsi display sys-info
```

- You cannot "escape" non-alphanumeric characters with a backslash or with quotes. For example, you will get an error message if you enter:

```
Switch(config)# alias show\ sys-info display sys-info
```

- You can have multiple aliases for the same command, but you cannot have multiple commands mapped to the same alias. For example, if you enter:

```
Switch(config)# alias show display
Switch(config)# alias abc display
```

The aliases show and abc will both invoke the display command. However, if you enter:

```
Switch(config)# alias show display
Switch(config)# alias show enable
```

The show alias will invoke the enable command.

- You *can* use an alias to invoke a multiple word command. For example, you can enter:

```
Switch(config)# alias ssi display sys-info
```

The ssi alias will now invoke the command display sys-info.

- You *cannot* nest aliases. For example, if you enter:

```
Switch(config)# alias show display
Switch(config)# alias ssi show sys-info
```

The ssi command will return an error message.

- You cannot alias an argument of a command. For example, if you try entering:

```
Switch(config)# alias show display
Switch(config)# alias si sys-info
```

The command show si will return an error message because the switch is trying to parse it as display si.

- If you use the name of an existing command as an alias name, it will override the existing command. For example, if you enter:

```
Switch(config)# alias qos display
```

The qos command will behave as if you had entered display. To fix this, enter:

```
Switch(config)# no alias <command>
```

In the case of fixing the qos command to its original function, you would enter:

```
Switch(config)# no alias qos
```

- An alias does not support multiple command lines. For example, if you enter:

```
Switch(config)# alias dvdu display version\n display user
```

You will get an error message.

- You cannot concatenate aliases. For example, if you enter:

```
Switch(config)# alias display show
Switch(config)# alias pa port-aggregation
Switch(config)# show pa 1
```

You will get an error message after you enter show pa.

- The maximum number of aliases that can be configured on a switch is 128.
- The following are reserved words that cannot be used as an alias name:

  - alias
  - all
  - bfd
  - configure
  - disable
  - display

  - enable
  - end
  - exit
  - logout
  - name
  - no

  - python
  - quit
  - reload
  - remove
  - restart
  - save

Chapter 2. Switch Administration

Your RackSwitch is ready to perform basic switching functions right out of the box. Some of the more advanced features, however, require some administrative configuration before they can be used effectively.

The extensive Lenovo Cloud Network Operating System for the switch provides a variety of options for accessing the switch to perform a variety of configurations and to view switch information and statistics.

This chapter discusses the various commands used to administer the switch:

- "Administration Interfaces" on page 42
- "Industry Standard Command Line Interface" on page 43
- "Establishing a Connection" on page 44
- "Zero Touch Provisioning" on page 51
- "DHCP IP Address Services" on page 55
- "Switch Login Levels" on page 62
- "Ping" on page 64
- "Traceroute" on page 69
- "Network Time Protocol" on page 72
- "Domain Name Server Client" on page 77
- "System Logging" on page 79
- "Idle Disconnect" on page 87
- "Python Scripting" on page 88
- "REST API Programming" on page 89

Administration Interfaces

Cloud NOS provides a variety of user interfaces for administration. These interfaces vary in character and in the methods used to access them. Some are text-based and some are graphical; some are available by default, while others require configuration; some can be accessed by local connection to the switch, while others are accessed remotely using various client applications. For example, administration can be performed using any of the following:

- A built-in, text-based command-line interface (CLI) and menu system for switch access via a serial port connection or an optional Telnet or SSH session

- SNMP support for access through third party commercial and open source network management applications.

The specific interface chosen for an administrative session depends on your preferences, the switch configuration, and the available client tools.

In all cases, administration requires that the switch hardware is properly installed and turned on (see the *Lenovo RackSwitch Installation Guide*).

Industry Standard Command Line Interface

The Industry Standard Command Line Interface (ISCLI) provides a simple and direct method for switch administration. Using a basic terminal, you can issue commands that allow you to view detailed information and statistics about the switch, and to perform any necessary configuration and switch software maintenance.

You can establish a connection to the ISCLI in any of the following ways:

- Serial connection via the serial port on the switch (this option is always available)

- Telnet connection over the network

- SSH connection over the network

Establishing a Connection

The factory default settings permit initial switch administration through the built-in serial port. The switch can also be initially configured through the OOB management port that gets a default IP address (192.168.50.50/24); in this case, the user is able to log in via SSH into the port and perform initial configuration.

Remote access using the network requires the accessing terminal to have a valid, routable connection to the switch interface. The client IP address may be configured manually, or an IP address can be provided automatically to the switch using a service such as DHCP (see "DHCP IP Address Services" on page 55). An IPv6 address can also be obtained using IPv6 stateless address configuration.

Note: Throughout this manual, *IP address* is used in places where either an IPv4 or IPv6 address is allowed. IPv4 addresses are entered in dotted-decimal notation (for example, 10.10.10.1), while IPv6 addresses are entered in hexadecimal notation (for example, 2001:db8:85a3::8a2e:370:7334). In places where only one type of address is allowed, *IPv4 address* or *IPv6 address* is specified.

Using the Switch Management Interface

To manage the switch through the management interface, you must configure it with an IP interface. Configure the IP address and network mask and default gateway address:

1. Log onto the switch.

2. Enter Global Configuration mode.

```
Switch> enable
Switch# configure device
Switch(config)#
```

3. Configure a management IP address and network mask:

   ● IPv4 configuration:

   ```
   Switch(config)# interface mgmt 0
   Switch(config-if)# ip address <IPv4 address>/<IPv4 network mask length>
   Switch(config-if)# exit
   ```

   ● IPv6 configuration:

   ```
   Switch(config)# interface mgmt 0
   Switch(config-if)# ipv6 address <IPv6 address>/<IPv6 network mask length>
   Switch(config-if)# exit
   ```

4. Configure the appropriate default gateway:

   ● IPv4 configuration:

   ```
   Switch(config)# vrf context management
   Switch(config-vrf)# ip route 0.0.0.0 0.0.0.0 <default gateway IPv4 address>
   Switch(config-vrf)# exit
   ```

- IPv6 configuration:

```
Switch(config)# vrf context management
Switch(config-vrf)# ipv6 route ::/0 <default gateway IPv6 address>
Switch(config-vrf)# exit
```

Once you configure a management IP address for your switch, you can connect to the management port and use a Telnet or an SSH client from an external management station to access and control the switch. The management port provides out-of-band management.

Note: To use a telnet client, you must first enable telnet access with the command:

```
Switch(config)# feature telnet
```

Other Ways to Manage the Switch Using IP

Besides using the out-of-band management port to administer the switch, you can manage the switch using an in-band connection over the data ports. The following options are available for configuring in-band management:

- Switched Virtual Interface (SVI)

- L3 routed ports

"Switch Virtual Interfaces" on page 202 contains rules and more details about using an SVI, while "Configuring a Routed Port" on page 408 contains more details about configuring routed ports. The following section contains examples of each.

Configuring a Switched Virtual Interface for Management

A Switched Virtual Interface is a VLAN that has an IP address assigned directly on it via the command:"

```
Switch(config)# interface vlan <VLAN number>
```

The VLAN must already exist before you configure the VLAN interface, and the VLAN must be allowed on any data ports you want to use to manage the switch. Along with configuring the VLAN interface, if you want to connect to the switch via a remote IP subnet, configure an in-band default gateway.

The following is an example of configuring an SVI and associated default gateway.

1. Log onto the switch.

2. Enter configuration mode and then create the desired VLAN that will be used by the SVI

```
Switch> enable
Switch# configure device
Switch(config)# vlan <VLAN number>
Switch(config)# exit
```

3. Create the SVI and configure the IP address and network mask.

```
Switch(config)# interface vlan <VLAN number>
Switch(config-if)# ip address <IPv4 or IPv6 address>/<IPv4 or IPv6 prefix length>
Switch(config-if)# exit
```

4. Configure the in-band default gateway (optional).

- IPv4 configuration:

```
Switch(config-if)# ip route 0.0.0.0/0 <default gateway IPv4 address>
```

- IPv6 configuration:

```
Switch(config-if)# ipv6 route ::/0 <default gateway IPv6 address>
Switch(config-vrf)# exit
```

You must carry the VLAN being used for management on at least one of the
in-band data ports, to permit management of the switch via this path.

Using the Switch Ethernet Ports in Routed Port Mode for Management

You also can configure in-band management directly on any of the switch Ethernet
data ports by setting the physical interface to Routed Port mode. To allow in-band
management via the Routed port feature use the following procedure:

1. Log onto the switch.

2. Enter interface mode and configure an ethernet interface as routed port.

```
Switch> enable
Switch# configure device
Switch(config)# interface ethernet <chassis number>/<port number>
Switch(config-if)# no bridge-port
```

3. Configure the interface IP address and network mask on this physical Ethernet
interface.

- IPv4 configuration:

```
Switch(config-if)# ip address <IPv4 address>/<IPv4 prefix length>
Switch(config-if)# exit
```

- IPv6 configuration:

```
Switch(config-if)# ipv6 address <IPv6 address>/<IPv6 prefix length>
Switch(config-if)# exit
```

4. (Optional) Configure the in-band default gateway.

- IPv4 configuration:

```
Switch(config)# ip route 0.0.0.0/0 <default gateway IPv4 address>
```

● IPv6 configuration:

```
Switch(config)# ipv6 route ::/0 <default gateway IPv6 address>
```

Once you configure the IP address and have a network connection, you can use a Telnet or an SSH client from an external management station to access and control the switch. Once the default gateway is enabled, the management station and the switch do not need to be on the same IP subnet to communicate.

The switch supports an industry standard command-line interface (ISCLI) that you can use to configure and control the switch over the network using a Telnet or an SSH client. You can use the ISCLI to perform many basic network management functions. In addition, you can configure the switch for management using an SNMP-based network management system.

For more information, see the documents listed in "Additional References" on page 28.

Using Telnet

A Telnet connection offers the convenience of accessing the switch from a workstation connected to the network. Telnet access provides the same options for user and administrator access as those available through the console port.

By default, Telnet access is disabled. Use the following command to enable or disable Telnet access:

```
Switch(config)# [no] feature telnet
```

Once the switch is configured with an IP address and gateway, you can use Telnet to access switch administration from any workstation connected to the management network.

To establish a Telnet connection with the switch, run the Telnet client on your workstation, use Telnet as the protocol type and the switch's IP address as the hostname.

You will then be prompted to enter a password as explained in "Switch Login Levels" on page 62.

By default, Telnet uses TCP port 23 of the remote host to establish a connection from the switch. When initializing a Telnet session, you can specify the TCP port of the remote host by using the following command on the switch:

```
Switch# telnet <switch IPv4 address> port <1-65535>
```

Note: The specified port will be used only for the current Telnet session. Future sessions will not use the selected port.

By default, Telnet clients will connect to the local Telnet server using TCP port 23 on the switch. To configure the TCP port used by a Telnet client when establishing a connection to the switch, use the following command:

```
Switch(config)# telnet server port <1-65535>
```

Using Secure Shell

Although a remote network administrator can manage the configuration of a switch via Telnet, this method does not provide a secure connection. The Secure Shell (SSH) protocol enables you to securely log into another device over a network to execute commands remotely. As a secure alternative to using Telnet to manage switch configuration, SSH ensures that all data sent over the network is encrypted and secure.

By default, SSH access is enabled. Use the following command to enable or disable SSH access:

```
Switch(config)# [no] feature ssh
```

The switch can do only one session of key/cipher generation at a time. Thus, an SSH client will not be able to log in if the switch is doing key generation at that time. Similarly, the system will fail to do the key generation if an SSH client is logging in at that time.

The supported SSH encryption and authentication methods are:

● Server Host Authentication: Client RSA-authenticates the switch when starting each connection

● Key Exchange: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group14-sha1

● Encryption: aes128-ctr, aes192-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com

● MAC: hmac-sha2-256, hmac-sha2-512, hmac-sha1-etm@openssh.com, hmac-sha2-256-etm@openssh.com

● User Authentication: Local password authentication, TACACS+

Lenovo Cloud Network Operating System implements the SSH version 2.0 standard and is confirmed to work with SSH version 2.0 compliant clients such as the following:

● OpenSSH_6.7p1 for Linux

● Secure CRT Version 7.3.4(build 839)

● Putty SSH release 0.63

Using SSH with Password Authentication

Once the IP parameters are configured, you can access the command line interface using an SSH connection.

To establish an SSH connection with the switch, run the SSH client on your workstation, use SSH as the protocol type and the switch's IP address as the hostname.

You will then be prompted to enter a password as explained in "Switch Login Levels" on page 62.

Using SSH with Server Key Authentication

SSH can also be used for switch authentication based on asymmetric cryptography. Server encryption keys can be generated on the switch and used to authenticate incoming login attempts based on the client's private encryption key pairs. After a predefined number of failed server key authentication attempts, a login error will appear and the SSH session will be disconnected.

To set up server key authentication:

1. Disable SSH:

```
Switch(config)# no feature ssh
```

Note: SSH settings cannot be modified if SSH is enabled.

2. Generate an SSH key:

   ● DSA:

   ```
   Switch(config)# ssh key dsa [force]
   ```

   ● RSA:

   ```
   Switch(config)# ssh key rsa [force]
   ```

   Note: You can also configure the length of the RSA key by using the following command:

   ```
   Switch(config)# ssh key rsa length <768-2048>
   ```

3. Configure a maximum number of failed server key authentication attempts before the SSH session will be disconnected:

   ```
   Switch(config)# ssh login-attempts <1-10>
   ```

   Note: The default number of failed attempts is 3.

4. Re-enable SSH:

   ```
   Switch(config)# feature ssh
   ```

Once the server key is configured on the switch, a client can use SSH to log in from a system where the private key pair is set up.

Using Simple Network Management Protocol

CNOS provides Simple Network Management Protocol (SNMP) version 1, 2, and 3 support for access through any network management software, such as Switch Center or Lenovo XClarity.

Note: The SNMP read function is enabled by default. For best security practices, if SNMP is not needed for your network, disable this function prior to connecting the switch to the network.

To access the SNMP agent on the switch, the read and write community strings on the SNMP manager must be configured to match those on the switch.

The read and write community strings on the switch can be configured using the following commands:

- read-only access community string:

```
Switch(config)# snmp-server community <community string (1-32 characters)> ro
```

- read-write access community string:

```
Switch(config)# snmp-server community <community string (1-32 characters)> rw
```

The SNMP manager must be able to reach any one of the IP interfaces on the switch.

For the SNMP manager to receive the SNMPv1 traps sent out by the SNMP agent on the switch, configure the trap host on the switch with the following command:

```
Switch(config)# snmp-server host <IP address> traps version 1 <trap host community string>
```

For more information on SNMP usage and configuration, see Chapter 30, "Simple Network Management Protocol."

Zero Touch Provisioning

Zero Touch Provisioning (ZTP) enables a switch to automatically provision itself using the resources available on the network without manual intervention. When a switch with ZTP enabled starts up, it locates a DHCP server which provides the switch with an interface IPv4 address and a gateway IPv4 address. The switch then obtains the IP address of a TFTP server from which it will download the necessary boot file. The next step is for the switch to run the boot file.

On the switch, ZTP will trigger when any of the following conditions are met:

- a switch boots with no startup configuration (only the default configuration)

- the startup configuration is erased and the switch is reloaded

- ZTP is forcedly enabled from the CLI

Note: ZTP will not be triggered if it is forcedly disabled from the CLI.

During the boot process, if the switch does not find a startup configuration and ZTP is enabled, the switch will enter ZTP mode. When forcedly enabled from the CLI, the switch enters ZTP mode regardless of the presence of a startup configuration. The switch will search for available DHCP servers and request them to acquire an interface address, a gateway address, the TFTP server address, and the boot file name.

After the information from the DHCP server is obtained, ZTP will download and run the boot file, and then execute the ZTP process according to the boot file. ZTP automatically handles the process of upgrading the switch software image and installing configuration files.

Notes:

- During the boot process, a prompt will appear asking if you want to abort or continue the ZTP process. If you choose to exit ZTP, the switch will continue with its normal boot process, using the default configuration or any startup configuration, if one is present on the switch and ZTP was forcedly enabled from the CLI.

- If ZTP was forcedly enabled and no DHCP server was found during the ZTP process, any previous IPv4 address manually configured of the management interface will be removed.

- If ZTP is canceled during its execution, the switch exits ZTP mode. If an interface IPv4 address was obtained, it will not be released. If any files where downloaded, they will not be deleted.

- Important ZTP events are logged by the switch and are available for display from a console session.

DHCP Discovery

After entering ZTP mode, the switch sends a DHCP discover message on its management interface requesting DHCP offers from the DHCP servers present on the network. The receiving DHCP server replies with a DHCP offer message.

When the DHCP client receives the DHCP offer message, it will request the DHCP server to send the following information:

- an interface IPv4 address

- a gateway IPv4 address

- the TFTP server IP address (using option 66)

- the boot file name (using option 67)

The switch completes the DHCP negotiation process (request and acknowledgement) with the DHCP server, which assigns the switch an IPv4 address. The switch then uses the acquired TFTP server IP address to contact the TFTP server. The boot file name contains the complete file path of the boot file on the TFTP server. The switch then downloads the boot file.

If no DHCP servers reply to the DHCP discover message or if no DHCP offer meets the ZTP requirements, the switch will be unable to complete the DHCP negotiation and an IPv4 address is not assigned (except the default IPv4 address 192.168.50.50/24, but this cannot help the switch finalize the ZTP process). ZTP will try three times to successfully obtain the required information. If it fails the DHCP negotiation three times, the switch exits ZTP mode and continues the normal boot process.

Notes:

- The interface IPv4 address obtained from the DHCP server is kept and used even after the ZTP process over.

- ZTP supports only DHCPv4 and not DHCPv6.

- ZTP supports only TFTP and not FTP, SCP, or other transfer protocols.

- DHCP servers must be configured with options 66 and 67 to ensure that the switch always obtains the TFTP server hostname and the boot file name during the ZTP process.

DHCP options 66 and 67 are enabled by default on the switch. If either of them is intentionally disabled, the ZTP process will result in a failure.

DHCP option 66 provides the IP address of a single TFTP server. To enable or disable DHCP option 66, use the following command:

```
Switch(config)# [no] ip dhcp client request tftp-server-name
```

DHCP option 67 provides the file path of the boot file needed by ZTP. To enable or disable DHCP option 67, use the following command:

```
Switch(config)# [no] ip dhcp client request bootfile-name
```

ZTP Boot File

The boot file is written in YAML format and contains switch models, and under each switch model are several fields that instruct the ZTP process what to do.

The boot file may contain up to three fields under each switch model:

- `img_name` - this instructs ZTP to update the switch software image to the specified image version and configure it as the standby image on the switch

- `configuration` - this instructs ZTP to copy the specified configuration file from the TFTP server and use it as the startup configuration file on the switch

- `script` - this instructs ZTP to copy the script file and execute it on the switch

ZTP checks the boot file for the switch model and execute the appropriate actions according to the fields under the correct switch model.

ZTP supports the execution of Python scripts. If there is a `script` field under the switch model in the boot file, the field has a higher priority than the other two fields (`img_name` and `configuration`) and ZTP will ignore them. ZTP downloads the Python script file to the switch and executes it. The script can also contain instructions to download and install a switch software image and a configuration file.

Note: The Python script file is stored in a temporary folder on the switch and it will be deleted once the switch reloads.

Following is an example of a boot file:

```
G8272:
    img_name     : G8272-10.6.0.1.img
    configuration: netboot_config_file_G8272
    script       : netboot_G8272.py

G8296:
    img_name     : G8296-10.6.0.1.img
    configuration: netboot_config_file_G8296
    script       : netboot_G8296.py

G8332:
    img_name     : G8332-10.6.0.1.img
    configuration: netboot_config_file_G8332
    script       : netboot_G8332.py
```

Note: After the ZTP process is over, the switch will be reloaded if the software image or the startup configuration are updated. If ZTP executes a Python script, the reloading of the switch is decided by the script instead.

Forcedly Enabling or Disabling ZTP

ZTP can be forcedly enabled on the switch even if there is a startup configuration present. It can also be forcedly disabled to not execute even if there is no startup configuration.

ZTP can have one of the following states:

- Default
- Forcedly Enabled
- Forcedly Disabled

To forcedly enable ZTP on the switch, use the following command:

```
Switch(config)# startup zerotouch force enable
```

To forcedly disable ZTP on the switch, use the following command:

```
Switch(config)# startup zerotouch force disable
```

To reset the ZTP to its default setting, use the following command:

```
Switch(config)# no startup zerotouch force
```

To view the current ZTP state, use the following command:

```
Switch# display boot

Current ZTP State: Enable
Current FLASH software:
 active image: version 10.6.0.1, downloaded 18:39:47 UTC Wed Sep 16 2015
 standby image: version 10.6.0.1, downloaded 18:44:40 UTC Wed Sep 16 2015
 Uboot: version 10.6.0.1, downloaded 17:49:51 UTC Thu Jul 30 2015
Currently set to boot software active image
Currently scheduled reboot time: none
Current port mode: default mode
```

To view the ZTP parameters obtained after the ZTP process has executed, use the following command:

```
Switch# display zerotouch

TFTP server: 10.122.3.69
Image: G8xxx-10.6.0.1.img
Configuration: netboot_config_file_G8xxx
Script: netboot_G8xxx.py
```

DHCP IP Address Services

For remote switch administration, the client terminal device must have a valid IP address on the same network as the switch interface. The IP address on the client device may be configured manually, or obtained automatically using IPv6 stateless address configuration, or an IP address may be obtained automatically via DHCP relay as discussed in the next section.

The switch can function as a relay agent for DHCP. This allows clients to be assigned an IP address for a finite lease period, reassigning freed addresses later to other clients. Acting as a relay agent, the switch can forward a client's IP address request to up to five DHCP servers. Additionally, up to five domain specific DHCP servers can be configured for each of up to 10 VLANs.

When a switch receives a DHCP request from a client seeking an IP address, the switch acts as a proxy for the client. The request is forwarded as a UDP unicast MAC layer message to the DHCP servers configured for the client's VLAN or to the global DHCP servers if no domain-specific DHCP servers are configured for the client's VLAN. The servers respond to the switch with a unicast reply that contains the IP default gateway and the IP address for the client. The switch then forwards this reply back to the client.

DHCP is described in RFC 2131 and the DHCP relay agent supported on the switch is described in RFC 1542. DHCP uses User Datagram Protocol (UDP) as its transport protocol. The client sends messages to the server on port 67 and receives messages from the server on port 68.

DHCP Client Configuration

DHCP is enabled by default on the management interface and disabled on all other interfaces. You can enable DHCP only on a maximum of 10 interfaces, including the management interface.

To enable or disable DHCP on an interface (for example ethernet interface 1/12), use the following command:

- for DHCPv4:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# no bridge-port
Switch(config-if)# ip address dhcp
```

- for DHCPv6:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# no bridge-port
Switch(config-if)# ipv6 address dhcp
```

Notes:

- DHCP cannot be enabled on an interface configured as a switch port, only on routing ports.

- Manually configuring an IP address on an interface will disable DHCP for that interface.

DHCPv4 Hostname Configuration (Option 12)

The switch supports DHCPv4 hostname configuration as described in RFC 2132, option 12. DHCPv4 hostname configuration is disabled by default.

The switch's hostname can be manually configured using the following command:

```
Switch(config)# hostname <system network name>
```

Note: If the hostname is manually configured, the switch does not replace it with the hostname received from the DHCPv4 server.

After DHCP configures the hostname on the switch, if the DHCPv4 configuration is disabled, the switch retains the hostname.

To enable or disable DHCP hostname configuration, use the following command on an interface (in this example, ethernet port 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# [no] ip dhcp client request host-name
```

To view the system hostname use the following command:

```
Switch> display hostname
```

Note: The switch prompt also displays the hostname.

DHCPv4 Syslog Server (Option 7)

The switch supports the requesting of the Syslog server IP address from the DHCP server as described in RFC 2132, option 7. The DHCPv4 Syslog server request option is disabled by default.

Note: Manually configured Syslog servers take priority over the DHCPv4 Syslog server.

Up to three Syslog server addresses received from the DHCPv4 server can be used. The Syslog server addresses can be learned over the management port or an ethernet port.

To enable or disable the DHCP Syslog server request, use the following command on an interface (in this example, ethernet port 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# [no] ip dhcp client request log-server
```

To view the Syslog server address, use the following command:

```
Switch> display logging server

Logging server:              enabled
{*2.2.2.1}
        Server severity:     debugging
        Server facility:       local7
        Server vrf:            data
* - Values assigned by DHCP Client.
```

DHCPv4 NTP Server (Option 42)

This option request the DHCP server to provide a list of IP addresses indicating Network Time Protocol (NTP) servers available to the client. The NTP servers are listed in order of preference. The switch supports the requesting of NTP servers as described in RFC 2132, option 42.

By default, the switch does not include this request in DHCPv4 messages. To enable or disable this option on an interface, use the following command (in this example, ethernet port 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# [no] ip dhcp client request ntp-server
```

Note: Any manually configured NTP server will not be overwritten by the NTP servers received via DHCPv4.

To view the list of NTP servers, use the following command:

```
Switch> display ntp peers
```

DHCPv4 Vendor Class Identifier (Option 60)

This option is used by a DHCP client to identify itself to the DHCP server. It is used to define the vendor type and functionality of the DHCP client. The DHCP client can communicate to a server that it uses a specific type of hardware or software by specifying its Vendor Class Identifier (VCI).

The switch supports the identifying of a TFTP server as described in RFC 2132, option 60.

Each switch interface can be configured with a different VCI.

By default, the switch will include this option in DHCPv4 packets. To enable or disable the identification of TFTP servers use the following command (in this example, ethernet port 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# [no] ip dhcp client class-id
```

Note: Depending on the Lenovo RackSwitch, the default VCI is different.

- o for the Lenovo RackSwitch G8272, the default VCI is 'LENOVO G8272'
- o for the Lenovo RackSwitch G8296, the default VCI is 'LENOVO G8296'
- o for the Lenovo RackSwitch G8332, the default VCI is 'LENOVO G8332'
- o for the Lenovo RackSwitch NE2572, the default VCI is 'LENOVO NE2572'
- o for the Lenovo RackSwitch NE10032, the default VCI is 'LENOVO NE10032'
- o for the Lenovo RackSwitch NE1032, the default VCI is 'LENOVO NE1032'
- o for the Lenovo RackSwitch NE1032T, the default VCI is 'LENOVO NE1032T'
- o for the Lenovo RackSwitch NE1072T, the default VCI is 'LENOVO NE1072T'

DHCPv4 Snooping

DHCP snooping provides security by filtering untrusted DHCP packets and by building and maintaining a DHCP snooping binding table.

A trusted port is an interface connected to a legitimate DHCP server. By default, all ports are untrusted. To configure a port as trusted, enter the following Interface mode command:

```
Switch(config-if)# [no] ip dhcp snooping trust
```

By default, DHCP snooping is disabled on all VLANs. You can enable DHCP snooping globally or on one or more VLANs. To enable this feature globally, enter the following command:

```
Switch(config)# [no] ip dhcp snooping
```

To enable DHCP snooping on a specific VLAN, use the following command:

```
Switch(config)# [no] ip dhcp snooping vlan <VLAN>
```

where *VLAN* is the VLAN ID.

Configure the DHCPv4 Snooping Binding Table

The DHCPv4 snooping binding table contains the MAC address, the IP address, lease time, binding type, VLAN number, and port number that correspond to the local untrusted interface on the switch; it does not contain information regarding hosts interconnected with a trusted interface.

The binding table is saved to flash memory every ten minutes. When the system reboots, the binding table is recovered from the flash file. The maximum number of entries in the DHCPv4 snooping binding table is 2048.

Sometimes you may want to manually configure the binding table entries, such as when you need to use a static IP address. Use the following command to configure a binding table entry:

```
Switch(config)# ip dhcp snooping binding <MAC> vlan <VLAN> <IP address>
interface ethernet <slot>/<port> expiry <lease time range>
```

where:

Argument	Description
MAC	The MAC address of the switch
VLAN	The VLAN ID; an integer from 1-4093.
IP address	A valid IPv4 address

Argument	Description
slot	The ethernet slot
port	The ethernet port
lease time range	The lease time range; an integer from 1-4294967295

Configure the DHCPv4 Snooping Syslog

The DHCP snooping daemon creates syslogs when some important events happen, such as a change to a dynamic entry or the timer.

There are two timers in DHCP snooping. One refreshes DHCP snooping binding entries every 60 seconds. The other one saves the binding table to flash every ten minutes. These syslogs are useful for monitoring and adjusting DHCP.

To set the DHCP snooping log level, enter::

```
Switch(config)# logging level dhcp-snp <logging level>
```

where:

Logging Level	Meaning
0	Emergency
1	Alert
2	Critical
3	Error
4	Warning
5	Notification
6	Information
7	Debug

DHCP Snooping Limitations

- DHCP snooping is not supported on a management port.

- DHCP is only supported on Ethernet ports. It is not supported on a portchannel or routing port.

- DHCP snooping does not support LACP or static aggregations.

DHCP Relay Agent

When DHCP clients and associated servers are not on the same physical subnet, a DHCP relay agent can transfer DHCP messages between them. When a DHCP request arrives on an interface, the relay agent forwards the packet to all DHCP server IP addresses configured on that interface. The relay agent forwards replies from all DHCP servers to the host that sent the request. If no DHCP servers are configured on that interface, the relay agent will not forward packets.

DHCP has two versions. DHCPv4 is used to configure hosts with IPv4 addresses, IPv4 prefixes, and other configuration data required to operate in an IPv4 network. DHCPv6 is used to configure hosts with IPv6 addresses, IPv6 prefixes, and other configuration data required to operate in an IPv6 network.

For DHCPv4, you can configure the relay agent to add the relay agent information (option 82) in the DHCPv4 message and then forward it to the DHCPv4 server. The reply from the server is forwarded back to the client after removing option 82.

The DHCP Relay Agent is globally enabled by default. To globally enable or disable DHCP use the following command:

- for DHCPv4:

```
Switch(config)# [no] ip dhcp relay
```

- for DHCPv6:

```
Switch(config)# [no] ipv6 dhcp relay
```

DHCP relay can be configured differently on each ethernet port or VLAN. The maximum number of DHCP servers configured on an interface is 32. To configure DHCP on an interface, use the following steps:

1. Enter the configuration menu for the desired interface (in this example, ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)#
```

2. Configure the DHCP server address:

- for DHCPv4:

```
Switch(config-if)# ip dhcp relay address <IPv4 server address>
```

- for DHCPv6:

```
Switch(config-if)# ipv6 dhcp relay address <IPv6 server address>
```

3. To view the current DHCP settings, use the following command:

- for DHCPv4:

```
Switch> display ip dhcp relay
```

- for DHCPv6:

```
Switch> display ipv6 dhcp relay
```

DHCPv4 Option 82

DHCPv4 option 82 provides a mechanism for generating IP addresses based on the location in the network of the client device. When you enable the DHCPv4 relay agent option on the switch, it inserts the relay agent information option 82 in the packet. The switch then sends a unicast DHCPv4 request packet to the DHCPv4 server. The DHCPv4 server uses the option 82 field to assign an IP address and sends the packet, with the original option 82 field included, back to the relay agent. The DHCPv4 relay agent strips off the option 82 field in the packet and sends the packet to the DHCPv4 client.

The configuration of this feature is optional. The feature helps resolve several issues where untrusted hosts access the network. See RFC 3046 for details.

To configure DHCPv4 option 82, use the following command:

```
Switch(config)# ip dhcp relay information option
```

Switch Login Levels

To enable better switch management and user accountability, two levels or *classes* of user access have been implemented on the switch. The levels of access to CLI management functions and screens increase as needed to perform various switch management tasks. Conceptually, access classes are defined as follows:

- *Network Operators* can only make temporary changes on the switch. These changes will be lost when the switch is reloaded or reset. Operators have access to the switch management features used for daily switch operations. Because any changes an operator makes are undone by a reload of the switch, operators cannot severely impact switch operation.

- *Network Administrators* are the only ones that may make permanent changes to the switch configuration—changes that are persistent across a reload or reset of the switch. Administrators can access switch functions to configure and troubleshoot problems on the device. Because administrators can also make temporary (operator-level) changes as well, they must be aware of the interactions between temporary and permanent changes.

Note: The default (predefined) access classes cannot be removed or their rules modified. Also, new access classes cannot be created.

Access to switch functions is controlled through the use of unique usernames and passwords. Once you are connected to the switch via console, Telnet, or SSH, you are prompted to enter a password. The default username and password combinations for each access level are listed in Table 2.

Note: It is recommended that you change the default switch passwords after initial configuration and as regularly as required under your network security policies.

Table 2. *Default Username and Password Combinations*

User Account	Password	Description and Tasks Performed	Status
oper	oper	The Operator manages all functions of the switch. The Operator can reset ports, except the management port.	Disabled
admin	admin	The Administrator has complete access to all menus, information, and configuration commands on the switch, including the ability to change both the operator and administrator passwords.	Enabled

The following characters are restricted in CNOS and cannot be used in passwords:

"?`&()<>|;'$

For more details, see "End-user Access Control" on page 130.

To display the current role configurations, use the following command:

```
Switch> display role

Role : network-admin
  Description: Predefined network admin role has access to all commands
  on the switch
-----------------------------------------------------------------------
Rule    Perm    Type          Scope              Entity
-----------------------------------------------------------------------
1       permit  read-write

Role : network-operator
  Description: Predefined network operator role has access to all read
  commands on the switch
-----------------------------------------------------------------------
Rule    Perm    Type          Scope              Entity
-----------------------------------------------------------------------
1       permit  read
```

While a network administrator has access to all of the CLI commands, a network operator has a more limited access, only being able to run commands such as:

- **display**
- **end**
- **exit**
- **logout**
- **quit**
- **terminal**
- **enable**
- **disable**
- **ping**
- **ping6**
- **traceroute**
- **traceroute6**
- **ssh**
- **ssh6**
- **telnet**
- **telnet6**
- **where**
- **configure device**

Chapter 2: Switch Administration

Ping

Ping (Poll INternet Gateway) is an administration utility used to test the connectivity between two network IP devices. It also measures the length of time it takes for a packet to be sent to a remote host plus the length of time it takes for an acknowledgement of that packet to be received by the source host.

Ping functions by sending an Internet Control Message Protocol (ICMP) echo request to the specified remote host and waiting for an ICMP reply from that host.

Using this method, ping also determines the time interval between when the echo request is sent and when the echo reply is received. This interval is called round-trip time. At the end of the test, ping will display the minimum, maximum, and average round-trip times, and the standard deviation of the mean.

Besides the round-trip time, ping can also measure the rate of packet loss. This is determined by the number of received echo replies over the number of sent echo requests. It is displayed as a percentage.

The Switch also supports ping for IPv6 addressing.

To perform a standard ping test, use the following commands:

● IPv4:

```
Switch# ping <target IPv4 address> vrf management
```

● IPv6:

```
Switch# ping6 <target IPv6 address> vrf management
```

For example:

```
Switch# ping 10.10.10.1 vrf management

PING 10.10.10.1 (10.10.10.1) from 10.10.10.127: 56(84) bytes of data.
64 bytes from 10.10.10.1: icmp_seq=1 ttl=61 time=0.368 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=61 time=0.280 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=61 time=0.308 ms
64 bytes from 10.10.10.1: icmp_seq=4 ttl=61 time=0.291 ms
64 bytes from 10.10.10.1: icmp_seq=5 ttl=61 time=0.320 ms

--- 10.10.10.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 0.280/0.313/0.368/0.034 ms
```

Note: If no specific VRF instance is configured, the switch uses the default management VRF. In this case, the user can also use the following command:

```
Switch# ping <target IPv4 address>
or
Switch# ping6 <target IPv6 address>
```

Ping Configurable Parameters

Ping can be configured with various parameters, such as specifying the number or size of echo requests, the time interval between each transmission, or the nonresponsive timeout interval for sent packets.

Test Interruption

Ping tests can be manually stopped at any point in the process. When the interruption is detected, ping will stop sending echo requests and display the results based on the packets transmitted up to that point.

To manually terminate a ping test, press **<Ctrl + C>**.

Ping Count

By default, ping transmits a sequence of five echo requests. To configure the number of packets sent during the test, use the following command:

```
Switch# ping <target IPv4 address> count <1-655350>
```

Ping can also be configured to continuously send echo requests until the test is manually interrupted. To achieve this, use the following command:

```
Switch# ping <target IPv4 address> count unlimited
```

For IPv6 addressing, the commands are as follows:

```
Switch# ping6 <target IPv6 address> count <1-655350>
```

```
Switch# ping6 <target IPv6 address> count unlimited
```

Ping Packet Interval

By default, ping does not wait between consecutive echo requests. As soon as a echo reply has been received or the nonresponsive timer has expired, ping will send the next echo request.

To configure a time interval, in seconds, between the transmission of packets, use the following command:

```
Switch# ping <target IPv4 address> interval <1-60>
```

For IPv6 addressing, the command is as follows:

```
Switch# ping6 <target IPv6 address> interval <1-60>
```

Ping Packet Size

By default, ping sends echo requests with a packet size of 56 bytes. Specifying a larger size than the default can help in detecting the loss of big packets.

To configure the packet size, in bytes, use the following command:

```
Switch# ping <target IPv4 address> packet-size <1-65468>
```

For IPv6 addressing, the command is as follows:

```
Switch# ping6 <target IPv6 address> packet-size <1-65468>
```

Ping Source

By default, ping automatically chooses the outgoing interface for echo requests and sends the packets using the IP address of that interface. To check the connectivity of different paths through the network, you can specify the interface used for sending echo requests.

To use a specific interface during the ping test, use the following command:

```
Switch# ping <target IPv4 address> source <source IPv4 address>
```

Note: The *source IPv4 address* is the IP address of the desired switch interface.

You can also choose the interface used for the ping test by directly specifying the desired interface. To achieve this, use the following command (in this example, ethernet port 1/12 is used):

```
Switch# ping <target IPv4 address> interface ethernet 1/12
```

For IPv6 addressing, the commands are as follows:

```
Switch# ping6 <target IPv6 address> source <source IPv6 address>
```

```
Switch# ping6 <target IPv6 address> interface ethernet 1/12
```

Ping DF-Bit

By default, echo requests are fragmented when they are forwarded through the network. Configuring packets not to be fragmented when traversing the network can be help in determining the maximum transmission unit (MTU) of the path.

To enable the non-fragmentation of echo requests, use the following command:

```
Switch# ping <target IPv4 address> df-bit
```

Note: This parameter is configurable only for IPv4 addressing.

Ping Timeout

By default, after sending an echo request, ping waits up to a maximum of two seconds for an echo reply. If this time interval expires and an echo reply is not received, ping will declare that the remote host has timed out and that the sent packet is lost.

To configure the timeout interval, in seconds, use the following command:

```
Switch# ping <target IPv4 address> timeout <1-60>
```

For IPv6 addressing, the command is as follows:

```
Switch# ping6 <target IPv6 address> timeout <1-60>
```

Ping VRF

By default, ping uses the default Virtual Routing and Forwarding (VRF) instance. To configure ping to use a different VRF instance, use the following command:

```
Switch# ping <target IPv4 address> vrf {default|management}
```

Note: You can choose only between the default or management VRF instances.

For IPv6 addressing, the command is as follows:

```
Switch# ping6 <target IPv6 address> vrf {default|management}
```

Ping Interactive Mode

To configure a custom ping test, you can choose what parameters to change by combining the previously presented commands.

Besides this option, you can customize a ping test by using Ping Interactive Mode. In this mode, you can configure additional parameters: the type of service (ToS), the hop limit or time-to-live (TTL) and the data pattern.

Note: Ping Interactive Mode is only available for IPv4 addressing.

To enter Ping Interactive Mode, use the following command:

```
Switch# ping
```

You will be prompted to specify the value of each configurable parameter. If you
do not enter a value, the default will be used.

```
Switch# ping

Vrf context to use [default]: management
Protocol [ip]:
Target IP address: 10.241.1.11
Repeat count [5]: 7
Datagram size [56]: 100
Timeout in seconds [2]: 1
Sending interval in seconds [1]:
Extended commands [n]: yes
Source address or interface:
Type of service [0]:
Set DF bit in IP header? [no]: yes
Data pattern [0xABCD]:
PATTERN: 0xabcd
PING 10.241.1.11 (10.241.1.11) 100(128) bytes of data.
108 bytes from 10.241.1.11: icmp_seq=1 ttl=61 time=0.337 ms
108 bytes from 10.241.1.11: icmp_seq=2 ttl=61 time=0.288 ms
108 bytes from 10.241.1.11: icmp_seq=3 ttl=61 time=0.311 ms
108 bytes from 10.241.1.11: icmp_seq=4 ttl=61 time=0.288 ms
108 bytes from 10.241.1.11: icmp_seq=5 ttl=61 time=0.317 ms
108 bytes from 10.241.1.11: icmp_seq=6 ttl=61 time=0.288 ms
108 bytes from 10.241.1.11: icmp_seq=7 ttl=61 time=0.315 ms

--- 10.241.1.11 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 5997ms
rtt min/avg/max/mdev = 0.288/0.306/0.337/0.022 ms
```

Traceroute

Traceroute is a diagnostic tool used to determine the network route between the switch and a remote device. It displays the network nodes (routers or gateway devices) crossed by a packet until it arrives at the specified destination.

Traceroute sends a sequence of User Datagram Protocol (UDP) packets addressed to a remote device. To determine the intermediate routers between the source and the destination devices, traceroute adjusts the time-to-live (TTL) value, also known as hop limit, of each sequence of sent packets. When a packet crosses a router, its hop limit is decreased by one. If a router detects a hop limit of zero, it discards the packet and sends the source host an Internet Control Message Protocol (ICMP) error message - Time Exceeded.

Traceroute configures the starting sequence of packets with a hop limit of one. The packets reach the first router and their hop limit is reduced from one to zero. The router will not forward the packets, but will instead discard them. Then, it sends an ICMP error message to the source host.

Traceroute sends the next set of packets with a hop limit of two. This time, the first router forwards the packets, reducing their TTL value from two to one. The packets reach the second router, which updates their hop limit to zero and discards them. Then, the seconds router will send the source host an ICMP error message.

Traceroute continues to send packets with increasing hop limit until the targeted remote device receives the packets and returns an ICMP echo reply.

After receiving the echo reply, traceroute uses the returned ICMP messages to create a list of the routers crossed by the packets. It uses the time interval between transmission and reception of packets as the delay (or latency) value for each node.

The Switch also supports traceroute for IPv6 addressing.

To perform a traceroute test, use the following commands:

- IPv4:

```
Switch# traceroute <target IPv4 address>
```

- IPv6:

```
Switch# traceroute6 <target IPv6 address>
```

For example:

```
Switch# traceroute 10.241.1.11

traceroute to 10.241.1.11 (10.241.1.11), 30 hops max, 56 byte packets
 1  10.241.41.1 (10.241.41.1)  1.988 ms  2.117 ms  2.299 ms
 2  10.241.4.254 (10.241.4.254)  1.903 ms  1.914 ms  2.649 ms
 3  10.241.1.33 (10.241.1.33)  1.138 ms  1.195 ms  1.242 ms
 4  10.241.1.11 (10.241.1.11)  1.085 ms !X  1.079 ms !X  1.087 ms !X
```

Traceroute Configurable Parameters

Traceroute is less customizable than ping, providing options only for choosing the outgoing interface or Virtual Routing and Forwarding (VRF) instance.

Test Interruption

Traceroute tests can be manually stopped at any point in the process. When the interruption is detected, traceroute will stop sending UDP packets and display the results based on the packets transmitted up to that point.

To manually terminate a Traceroute test, press **<Ctrl + C>**.

Traceroute Source

By default, traceroute automatically chooses the outgoing interface for sending UDP packets and transmits the packets using the IP address of that interface. To check the connectivity of different paths through the network, you can specify the interface used for sending packets.

To use a certain interface during a traceroute test, use the following command:

```
Switch# traceroute <target IPv4 address> source <source IPv4 address>
```

Note: The *source IPv4 address* is the IP address of the desired switch interface.

For IPv6 addressing, the command is as follows:

```
Switch# traceroute6 <target IPv6 address> source <source IPv6 address>
```

In the case of IPv6 addressing, you can also choose the interface used for the traceroute test by directly specifying the desired interface. To achieve this, use the following command (in this example, ethernet port 1/12 is used):

```
Switch# traceroute6 <target IPv6 address> interface ethernet 1/12
```

Traceroute VRF

By default, traceroute uses the default Virtual Routing and Forwarding (VRF) instance. To configure traceroute to use a different VRF instance, use the following command:

```
Switch# traceroute <target IPv4 address> vrf {default|management}
```

Note: You can choose only between the default or management VRF instances.

For IPv6 addressing, the command is:

```
Switch# traceroute6 <target IPv6 address> vrf {default|management}
```

Traceroute Interactive Mode

To customize a traceroute test, you can also use Traceroute Interactive Mode. In this mode, you can configure additional parameters, such as the timeout interval, the hop limit or time-to-live (TTL), or the destination UDP port number.

Note: Traceroute Interactive Mode is only available for IPv4 addressing.

To enter Traceroute Interactive Mode, use the following command:

```
Switch# traceroute
```

You will be prompted to specify the value of each configurable parameter. If you do not enter a value, the default will be used.

```
Switch# traceroute

Vrf context to use [default]: management
Protocol [ip]:
Target IP address: 10.241.1.11
Source address: 10.241.41.27
Numeric display [n]: yes
Timeout in seconds [2]: 1
Probe count [3]: 5
Maximum time to live [30]: 20
Port Number [33434]: 15477
traceroute to 10.241.1.11 (10.241.1.11), 20 hops max, 56 byte packets
 1  10.241.41.1  0.665 ms  0.870 ms  1.102 ms  1.375 ms  1.611 ms
 2  10.241.4.254  1.459 ms  1.530 ms  1.568 ms  1.607 ms  1.646 ms
 3  10.241.1.33  0.790 ms  0.854 ms  1.654 ms  1.701 ms  1.755 ms
 4  10.241.1.11  0.287 ms !X  0.148 ms !X  0.096 ms !X  0.120 ms !X
0.130 ms !X
```

Network Time Protocol

The Network Time Protocol (NTP) is used to synchronize the internal clocks of network devices. This is helpful for troubleshooting network problems by correlating events on different network devices using, for example, syslog messages.

NTP provides the switch with a mechanism to accurately update its clock to be consistent with the clocks of other network devices within a precision of one millisecond.

NTP uses User Datagram Protocol (UDP) to communicate across the network.

By default, the NTP service is enabled on the switch. To enable or disable the NTP service, use the following command:

```
Switch(config)# [no] feature ntp
```

This can also be achieved by using the following command:

```
Switch(config)# [no] ntp enable
```

When NTP is enabled by using the **ntp enable** command, any configuration settings are saved in a file on the switch. If the protocol is disabled using the **no ntp enable** and then re-enabled, the switch will use that file to restore the NTP configuration. Any NTP commands issued while the protocol is disabled are ignored by the switch.

Note: If the NTP service is disabled by issuing the **no ntp feature** command, the NTP configuration is erased and it will not be restored after re-enabling the service.

By default, NTP will communicate across the interfaces configured in the management Virtual Routing and Forwarding (VRF) instance. To allow NTP to transmit and receive information using other VRF instances, use the following command:

```
Switch(config)# ntp use-vrf {default|management}
```

Note: Only the default or management VRF instances can be used.

NTP Synchronization Retry

An NTP synchronization can be manually triggered by using the following command:

```
Switch# ntp sync-retry
```

NTP Client and Peer

The switch can function as an NTP client and synchronize its internal clock with the clock of a remote NTP server. This synchronization is only one way, as the NTP server does not update its clock to match the one of the NTP client.

The switch supports both IPv4 or IPv6 address-based NTP.

To add or remove a remote NTP server, use the following command:

```
Switch(config)# [no] ntp server <IP address>
```

For the purpose of redundancy in scenarios where an NTP server is unreachable, NTP peers can be configured to communicate with each other. This mechanism is called symmetric-active mode and NTP peers associated in this manner can synchronize their internal clocks. The switch clock is updated using the clock of the remote NTP peer and vice-versa.

To add or remove a remote NTP peer, use the following command:

```
Switch(config)# [no] ntp peer <IP address>
```

Note: An NTP server and an NTP peer cannot share the same IP address. If you configure an NTP server using a specific IP address, you cannot use the same address to set up an NTP peer or vice-versa.

To view the list of currently configured NTP servers or peers, use the following command:

```
Switch> display ntp peers

-----------------------------------------------------------
Peer IP Address                      Serv/Peer
-----------------------------------------------------------
9.110.36.180                         Server (configured)
129.36.17.25                         Server (configured)
10.10.122.36                         Peer (configured)
10.10.122.59                         Peer (configured)
```

Note: The maximum number of NTP servers and peers statically or dynamically configured on the switch is 64.

To view the status of NTP servers and peers, use the following command:

```
Switch> display ntp peer-status

Total peers : 1
* - selected for sync, + - peer mode(active),
- - peer mode(passive), = - polled in client mode
remote            local            st    poll    reach    delay
----------------------------------------------------------------------
=9.110.36.180     9.111.86.200     16    64      0        0.00000
```

NTP Authentication Field Encryption Key

Information (packets) exchanged between the switch and NTP servers or peers can be configured to include encrypted authentication fields. These fields are encoded using a key.

By default, no encrypted fields are included in NTP packets.

To configure encryption authentication fields for NTP servers, use the following command:

```
Switch(config)# ntp server <IP address> key <1-65534>
```

To configure encryption authentication fields for NTP peers, use the following command:

```
Switch(config)# ntp peer <IP address> key <1-65534>
```

NTP Polling Intervals

The switch will periodically poll configured NTP servers or peers, and synchronize its internal clock.

Each individual NTP server or peer can be configured with different poll intervals by specifying a minimum and maximum time limit. The polling interval limits are configured in seconds, using powers of 2.

Each limit can be configured with a minimum value of 4, meaning 2^4 or 16 seconds, and a maximum value of 16, meaning 2^{16} or 65536 seconds (18.2 hours).

To configure the minimum polling interval for NTP servers, use the following command:

```
Switch(config)# ntp server <IP address> minpoll <4-16>
```

To configure the minimum polling interval for NTP peers, use the following command:

```
Switch(config)# ntp peer <IP address> minpoll <4-16>
```

By default, the minimum polling limit has a value of 4 (16 seconds).

To configure the maximum polling interval for NTP servers, use the following command:

```
Switch(config)# ntp server <IP address> maxpoll <4-16>
```

To configure the maximum polling interval for NTP peers, use the following command:

```
Switch(config)# ntp peer <IP address> maxpoll <4-16>
```

By default, the maximum polling limit has a value of 6 (64 seconds).

NTP Preference

During configuration, some NTP servers or peers can be marked as preferred over other devices. Preferred NTP servers or peers take precedence when a clock update is required.

To configure an NTP server as preferred, use the following command:

```
Switch(config)# ntp server <IP address> prefer
```

To configure an NTP peer as preferred, use the following command:

```
Switch(config)# ntp peer <IP address> prefer
```

Dynamic and Static NTP Servers

Using the Dynamic Host Configuration Protocol (DHCP) client, the switch can dynamically learn the addresses of NTP servers.

The DHCP client will add a dynamic server to the list of NTP servers and peers only if the new server is not already statically configured on the switch, or the list has not reached its maximum size of 64 entries.

Static NTP servers take precedence over the ones dynamically learned through DHCP. Any dynamic NTP server entry is removed if a static entry for the same NTP server is configured on the switch.

If the list of NTP servers and peers is full, a statically configured NTP server will replace the oldest dynamic entry in the list. If there are no dynamic entries, the static server will not be added to the list.

The dynamic learning of NTP servers can only be enabled on individual switch interfaces. For more details, see "DHCPv4 NTP Server (Option 42)" on page 57.

NTP Authentication

The switch can be configured to use authentication keys when exchanging packets with an NTP server. NTP authentication ensures that the internal clock is synchronized only using information received from a trusted NTP server.

NTP authentication uses a digital signature that is added to a packet, and it does not encrypt the data contained inside the packet. The packet and the configured authentication key are used to generate a non-reversible magic number which is appended to the packet. The receiving device does the same computation and then compares the results. If authentication succeeds (the two magic numbers match), the packet is used to synchronize the time. If authentication fails (the two magic numbers do not match), the packet is discarded.

By default, NTP authentication is disabled.

To configure an authentication key, use the following command:

```
Switch(config)# ntp authentication-key <1-65534 (key ID)> md5 <key string>
```

Note: You can configure up to 65534 NTP authentication keys. The maximum size of a *key string* is 8 characters. The *key string* is used together with the packet to generate the non-reversible magic number.

To remove an authentication key, use the following command:

```
Switch(config)# no ntp authentication-key <1-65534 (key ID)>
```

To view the configured NTP authentication keys, use the following command:

```
Switch> display ntp authentication-keys
```

Once an authentication key is created, the key can be configured as a trusted key and be used during the NTP authentication process. To enable or disable the use of a specific key in the authentication process, use the following command:

```
Switch(config)# [no] ntp trusted-key <1-65534 (key ID)>
```

To view the list of NTP trusted keys, use the following command:

```
Switch> display ntp trusted-keys
```

To enable or disable NTP authentication, use the following command:

```
Switch(config)# [no] ntp authenticate
```

To check the status of NTP authentication, use the following command:

```
Switch> display ntp authentication-status
```

NTP Authentication Configuration Example

To enable NTP authentication on the switch, use the following steps.

1. Configure an NTP authentication key:

```
Switch(config)# ntp authentication-key 10 md5 MyString
```

2. Configure the authentication key as a trusted key:

```
Switch(config)# ntp trusted-key 10
```

3. Enable NTP authentication:

```
Switch(config)# ntp authenticate
```

Domain Name Server Client

CNOS offers a Domain Name Server (DNS) client. A DNS is a hierarchical, decentralized naming system for computers and services connected in a network. Mainly, it translates easy-to-remember domain names (such as www.example.com) to numerical IP addresses (92.175.123.87 or 2001::ba5) needed to identify and locate the devices and services. This allows the DNS to be rapidly updated when a service or host location is changed in the network, with no direct impact on its users, who go on using the known host name

To enable or disable the DNS service, enter:

```
Switch(config)# [no] ip domain-lookup
```

To configure a system as a DNS server, enter:

```
Switch(config)# [no] ip name-server <IP address> vrf {default|management}
```

where *IP address* is a valid IPv4 or IPv6 address.

To set the default domain name, enter:

```
Switch(config)# [no] ip domain-name <default-domain-name> vrf
{default|management}
```

where *default-domain-name* is a valid domain name.

To add a domain to the list of domain names used by the DNS client, enter:

```
Switch(config)# [no] ip domain-list <domain-name> vrf {default|management}
```

where *domain-name* is a valid domain name.

To add or remove a local mapping service between a specific hostname and IP address, enter:

```
Switch(config)# [no] ip host <hostname> <IP address> vrf
{default|management}
```

where:

Table 3.

Argument	Description
hostname	The host name
IP address	a valid IPv4 or IPv6 address

To see the DNS server running configuration, enter:

```
Switch> display running-config dns
```

To see the DNS server options, such as whether it is enabled or disabled, the default domain, additional domains, name servers ,and IP address to hostname pairs, enter:

```
Switch> display hosts vrf {all|default|management}
```

Notes:

- The maximum number of domain name servers allowed per VRF instance is three.

- The maximum number of domain names allowed in the domain names list per VRF instance is six.

- The maximum number of hostname to IP address mappings allowed per VRF instance is 100.

System Logging

System logging (syslog) is the mechanism through which messages generated by different software components, such as BGP, NTP, or Telnet, are reported by the switch applications. These messages are reported using several types of outputs including the console terminal, virtual teletype (VTY) sessions (Telnet or SSH), log files, or they are sent to remote syslog servers. Logging messages provide operational information about software components, including the status of the application, error reports, and detailed debugging data.

The switch uses the following format when outputting syslog messages:

<Timestamp> <IP/Hostname>(*<OS string>*:*<VRF name>*%*<Syslog ID>*:[*<Process Name>*] *<Message>*

Note: The angle brackets (`<` and `>`) are not present in the syslog message format, but they are used here to differentiate the individual components of the message.

The following parameters are used:

- *Timestamp*

  The time of the message is displayed in the following format:

  `<year>-<month>-<day>-T<hour(1-24)>:<minute>:<second><timezone>`

  For example: `2015-08-01T18:39:59+00:00`

- *IP/Hostname*

  The IP address or hostname is displayed when configured.

  For example: `1.1.1.1` or `Switch`

- *OS string*

  A text string denoting the name of the switch operating system (in this case, `cnos`)

- *VRF name*

  The name of the VRF involved.

- *Syslog ID*

  The syslog ID consists of three components that identify the syslog message:

  <facility>-*<severity level>*-*<mnemonic>*

  o *facility* - the application to which the message refers (such as `BGP` or `NTP`)

  o *severity level* - the severity level of the message (a integer from 0 to 7)

  o *mnemonic* - a text string that uniquely describes a syslog message

- *Process Name*

  The process name is only used when the syslog message is generated by a shared library facility. In this case, the facility parameter within the Syslog ID field identifies the shared library facility, whereas the application process facility identifies the client process, which is using the shared library.

- *Message*

  The syslog message containing detailed event information.

Following are examples of syslog messages logged by process:

```
2017-05-17T04:50:33+00:00 G8272(cnos:data) %NSM-6-INIT: syslog service
initialized
```

```
2017-05-17T04:51:37+00:00 G8272(cnos:management)
%BGP-6-RIB_SCAN_REMAIN_TIME: RIB-SCAN timer is stopped with
remaining-time: 10 seconds
```

Following is an example of a syslog message logged by a shared library
(SECUREIMG) executed in the context of the IMISH process:

```
2017-05-17T21:19:17+00:00 G8272(cnos:data)
%SECUREIMG-6-VALID_IMAGE_DETECTED: [IMISH] Valid image detected
```

Following is an example of a message logged by a sub-application facility (COPP
thread of the QoS process):

```
2017-05-17T04:50:33.345+00:00 switch(cnos:data) %COPP-5-PACKET_LIMIT:
Maximum bandwidth utilization reached (port=eth0, limit=1 Gbps)
```

Syslog Output

Syslog messages are stored in two types of log files: the customer log file and the
platform log file.

The customer log file stores messages generated by the switch applications. This
includes logs that record normal daily operations of the protocols, as well as error
conditions detected. These events are primarily used for troubleshooting and
auditing. The customer log file is persistent across switch reloads and can be
modified or removed. It can be configured to log messages of a certain severity. Its
maximum size can also be configured to a value between 4 kilobytes (KB) and 10
megabytes (MB). The log file consists of 8 individual parts, each with a maximum
size of 1.25 MB. By default, the switch is configured with a log file that has a
maximum size of 10 MB and it stores messages with a severity level of 6 or higher.

The platform log file is used to store system messages generated by the switch
operating system. These messages are mostly used for internal and debugging
purposes, and are typically not of interest to the end users. These messages are
stored in a set of files which are persistent across switch reloads. However, the log
file is permanent and it cannot be removed or modified. It logs any system
messages regardless of their severity and it consists of four individual parts, each
with a maximum size of 1 MB, adding to a maximum total of 4 MB.

To configure the storing of syslog messages in a customer log file on the switch, use
the command:

```
Switch(config)# logging logfile <log name> <0-7 (severity level)> size <4,096-10485760
(bytes)>
```

To delete all customer log file entries, use the following command:

```
Switch# remove logging logfile
```

To disable the logging of messages to the log file, use the following command:

```
Switch(config)# no logging logfile
```

Syslog messages can also be enabled to be sent to devices connected via console or virtual teletype (VTY) sessions (Telnet, SSH), or to remote dedicated syslog servers. By default, the logging of messages is enabled on console sessions for messages with a severity level of 2 or higher, and on VTY sessions for messages with a severity level of 5 or higher.

To enable or disable message logging on console sessions, use the command:

```
Switch(config)# [no] logging console
```

To enable or disable message logging on VTY sessions, use the command:

```
Switch(config)# [no] logging monitor
```

To enable or disable message logging to the current terminal, use the command:

```
Switch(config)# [no] logging terminal
```

Syslog Severity Levels

System Logging has eight severity levels that are attributed with a syslog message. They are commonly associated with an integer from 0 to 7, where 0 is the highest severity level and 7 the lowest.

Following are the different types of syslog severity levels:

- Emergency (0) - the system is unstable
- Alert (1) - a major error has occurred, that requires immediate action
- Critical (2) - an urgent error occurred, that requires immediate action
- Error (3) - a non-urgent error occurred
- Warning (4) - indication that an error might occur if not remedied
- Notice (5) - unusual events, but not error conditions
- Informational (6) - normal operation messages
- Debug (7) - information useful for debugging

When a severity level is configured, the switch will only log messages that have a severity equal or greater than the configured value. For example, when setting a severity level of 4, the switch will log messages that have a severity of 0, 1, 2, 3, or 4. Messages with a severity level of 5, 6, or 7 are not logged.

Each software component (such as OSPF or Telnet) or facility on the switch can be configured with a different severity level. While message logging for a component cannot be disabled, the severity level can be set to emergency (0), logging only the most severe messages for the specified component.

To configure the severity level of a facility, use the following command:

```
Switch(config)# logging level <facility> <0-7 (severity level)>
```

You can configure the severity level of all facilities by using the following command:

```
Switch(config)# logging level all <0-7 (severity level)>
```

Note: Each facility has its own default syslog severity level, which is usually different among facilities.

To view the severity level of a facility, use the following command:

```
Switch(config)# display logging level <facility>
```

To reset the severity level to its default value, use the following command:

```
Switch(config)# [no] logging level <facility>
```

For more details on syslog severity level commands, consult the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

Syslog Time Stamping

Each syslog message is generated with a timestamp that specifies the moment in time when the event of the message occurred.

The timestamp of a syslog message has the following syntax:

<year>-<month>-<day>-T<hour>:<minute>:<second><timezone>

Note: The angle brackets (< and >) are not present in the timestamp syntax, but they are used here to differentiate the individual components of the message.

For example: 2015-08-01T18:39:59+00:00

By default, syslog uses seconds as its base unit for timestamps. System logging can be configured to add fractions of a second (milliseconds or microseconds) to the timestamp of a message. To achieve this, use the following command:

```
Switch(config)# logging timestamp {microseconds|milliseconds|seconds}
```

To reset syslog to use seconds in timestamps, use the following command:

```
Switch(config)# no logging timestamp {microseconds|milliseconds|seconds}
```

Syslog Rate Limit

By default, the switch has a limit of how many messages it can log during a specific time interval. Any messages received above the maximum limit are not logged.

For system messages, the default rate limit is 512 messages every 5 seconds.

For every severity level, the default rate limit is 1024 messages every 10 seconds. Regardless of the software component that generated them, only 1024 messages with a certain severity can be logged during 10 seconds.

For every software component, the default rate limit is 512 messages every 10 seconds.

To configure the rate limit of a specific type of syslog message, use the following command:

```
Switch(config)# logging rate-limit num {<1-4096 (number of messages)>|default}
interval {<1-600 (seconds)>|default} {facility <facility>|level <0-7 (severity level)>|
|system}
```

To display the current syslog rate limit configuration, use the following command:

```
Switch(config)# display logging rate-limit
```

To disable syslog rate limiting, use the following command:

```
Switch(config)# no logging rate-limit {facility <facility>|level <0-7 (severity
level)>|system}
```

For more details on syslog rate limit commands, consult the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

Syslog User Action Logging

You can configure whether user actions are logged to the console or to terminals.

To enable or disable user action logging on the console or on terminals, use the following command:

```
Switch(config)# [no] event-log (console | terminal) enable
```

Syslog Servers

You can configure the logging of messages to a remote dedicated syslog server. This can be statically configured, or dynamically learned via the Dynamic Host Configuration Protocol (DHCP).

By default, the logging of messages to a dedicated syslog server is disabled.

To add a static remote syslog server, use the following command:

```
Switch(config)# logging server <IP address> [<0-7 (severity level)>] [vrf {data|
|management}] [facility <facility>]
```

To remove a static remote syslog server, use the following command:

```
Switch(config)# no logging server <IP address> [vrf {data||management}]
```

The severity level of a static syslog server can be configured by specifying its value when running the above command. If the severity level is not explicitly configured, the default value, debug (7), is used.

Static syslog servers take precedence over dynamically learned ones. If a syslog server is added using DHCP and then a static server is configured on the switch, the dynamic entry may be removed if one of the following conditions is met:

- the static server being added has the same IP address as the currently configured dynamic server
- the maximum number of servers allowed for a given type of IP addressing or address family (IPv4 or IPv6) has been reached

If a syslog server is already statically configured on the switch, it will not be dynamically added to the list.

Each Virtual Routing and Forwarding (VRF) instance on the switch has a limit on the number of configured syslog servers (static or dynamic). The limit is three servers for each type of IP address family, meaning that the switch can have up to three configured IPv4 syslog servers and up to three configured IPv6 syslog servers on each VRF instance.

Note: Only DHCPv4 can be configured to request syslog server addresses from a DHCP server. Thus, only IPv4 syslog servers can be dynamically learned. IPv6 syslog servers can only be added statically.

The dynamic learning of syslog servers can be achieved only on individual switch interfaces. For more details, see "DHCPv4 Syslog Server (Option 7)" on page 56.

Note: The severity level of dynamic syslog servers is debug (7).

As soon as a valid static or dynamic syslog server address is known by the switch, it will immediately start forwarding syslog messages to the remote server.

Console Logging Flood Control

Syslog flooding occurs when one or more software components fail to function correctly and start to continuously generate a large number of syslog messages over a long time interval. Depending on the switch configuration, such syslog messages can be sent to devices connected through console or virtual teletype (VTY) sessions (Telnet, SSH).

Note: By default, the switch sends syslog messages to devices connected through a terminal connection (console, Telnet, or SSH).

Because terminal connections have a limited number of lines that can be displayed on the screen during a session, syslog flooding can cause the loss of control of the switch. The syslog messages appear on the screen faster than commands can be issued and processed correctly by the switch.

In the case where other connection options (for example SNMP or Python Scripting) are not available, to regain control over the switch you must power down the device by disconnecting it from the power source. Syslog flooding can also happen during the switch startup process, before the configuration can be modified or other connection option can be set up.

Console Logging Flood Control enables you to regain control over the terminal during syslog flooding. During a storm of syslog messages press **<Ctrl + C>** three times during a three second time interval. A confirmation message is displayed indicating that syslog messages will not be sent to the console. The message will also inform you that the sending of syslog messages to the console cannot be re-enabled for the following 60 seconds. After this one minute time period has expired, the sending of syslog messages to the console can be re-enabled by pressing **<Ctrl + C>** three times during a three second time interval.

From the moment the confirmation message is displayed, you can access the switch and resolve the problem by correcting the configuration issue. For example, disabling the logging of messages to the console, modifying the severity level of the software component that generated the syslog flood, or modify the configuration of that application.

Duplicate Syslog Message Suppression

Duplicate Syslog Message Suppression prevents the switch from logging more than one copy of a repetitive syslog message. When a sequence of identical syslog messages is detected, the first message is logged and its timestamp is recorded. The following repetitive syslog messages are suppressed for a certain time interval and they are not logged by the switch.

The first occurrence of the repetitive syslog message is always logged at the moment it happens. The following duplicate messages are suppressed during three different time intervals (suppression intervals). After a suppression interval, a summary message is displayed informing about the number of times the syslog message repeated.

Initially, duplicate messages are suppressed for 30 seconds after the first occurrence. If this time interval expires and the syslog message still repeats, it will be suppressed for another 120 seconds (2 minutes). The third time, the suppression interval is increased to 600 seconds (10 minutes). At the end of each suppression interval, the next duplicate syslog message will trigger the summary message. The expiration of a suppression interval does not generate the summary message. If the time interval expires and no duplicate syslog messages are logged, the summary message can be delayed well beyond the suppression interval.

By default, duplicate syslog message suppression is disabled on the switch. To enable or disable it, use the following command:

```
Switch(config)# [no] logging throttle
```

The following is an example of duplicate syslog message suppression and the summary message after the first suppression interval:

```
2016-07-2T7:39:10+00:00 switch(data) %TELNET-3-ERR_CONNECT: Failed to
obtain a VTY for a session: ''tty-server' detected the 'resource not
available' condition' There are no TTYs available

2016-07-2T7:39:45+00:00 switch(data) %TELNET-3-ERR_CONNECT: last message
repeated 2 times
```

The suppression of duplicate syslog messages is interrupted if a new syslog message is detected. The new syslog message is logged immediately, thus canceling the current suppression interval. Other syslog messages from the previous sequence are considered as new and the suppression process starts over.

Core Dump Information

When a core dump occurs, a core dump file is created, and the following notice is added to the system banner after login:

```
======================================================================
+ NOTE: Core dump files exist in FLASH.                              +
+       Use 'display cores' to see a list of all existing core dumps. +
+       Use 'system cores tftp://server[:port][/path] vrf management' +
+       to upload the core dumps for analysis.                       +
+       Use 'remove cores' to erase the dumps from FLASH.            +
======================================================================
```

Idle Disconnect

By default, the switch will disconnect your Telnet session after 10 minutes of inactivity. This function is controlled by the idle timeout parameter, which can be set from 0 to 35791 minutes, where 0 means the session will never timeout.

Use the following command to set the idle timeout value:

```
Switch# terminal session-timeout <0-35791>
```

Note: This command only applies to the current terminal session.

Python Scripting

You can create and execute local Python scripts on switches to make small programs that allow a switch to automatically provision itself, perform fault monitoring, upgrade the image files, or auto-generate configuration files. You can use local scripts as a key part of your auto-provisioning solutions. You can also manage scripts on the switch.

You can implement version control systems, automatically generate alerts, create custom logging tools, and automate the management of network devices. Using Python scripts, you can perform many functions that can be performed through the CLI. In addition to configuration, you can notify users by sending e-mails or updating the syslog.

To enter the Python Programming Shell, use the following command:

```
Switch# python

>>>
```

To exit the Python Programming Shell, use the following command:

```
>>> quit()

Switch#
```

Note: You can also press **<Ctrl + D>** to exit the Python Programming Shell.

See the *Lenovo Network Python Programming Guide for Lenovo Cloud Network Operating System 10.6* for details on how to create and execute Python scripts.

REST API Programming

The Lenovo REpresentational State Transfer (REST) Application Program Interface (API) enables you to remotely configure and manage a Lenovo switch using REST and HyperText Transfer Protocol (HTTP).

The REST API is a JavaScript Object Notation-based (JSON) wrapper around Lenovo's Python Programming interface. It is a component of Configuration, Management, and Reporting (CMR) on CNOS.

To enable and disable REST API on the switch, use the following command:

```
Switch(config)# [no] feature restApi
```

The default settings for REST API is enabled. HTTPS is also enabled by default.

To view the current state of REST API, use the following command:

```
Switch> display restApi server

rest server enabled port: 443
restApi pki rest_mgmt vrf management
restApi pki rest_default vrf default
```

See the *Lenovo REST API Programming Guide for Lenovo Cloud Network Operating System 10.6* for details on how to use the Lenovo REST API.

Chapter 3. System License Keys

License keys determine the number of available features on the switch. Each switch comes with a basic license that provides the use of a limited number of functions. On top of the basic license, optional upgrade licenses can be installed to expand the number of available features.

License keys will be installed only once per feature on a specific switch, regardless of the Lenovo Network OS used. When a new Lenovo NOS is installed on the switch, the license keys will need to be reinstalled.

The switch supports dual boot, meaning it can run both Lenovo Enterprise Network OS (ENOS) and Lenovo Cloud Network Operating System (CNOS). When you install a license key using CNOS it will be also available in ENOS and vice-versa.

This section discusses the following topics:

- "Obtaining License Keys" on page 92
- "Installing License Keys" on page 93
- "Uninstalling License Keys" on page 94
- "Transferring License Keys" on page 95
- "ONIE License Key" on page 96

Obtaining License Keys

License keys or activation keys can be acquired using the *Lenovo System x Features on Demand* (FoD) website:

https://fod.lenovo.com/lkms/angular/app/pages/index.htm#/welcome

You can also use the website to review and manage licenses, and to obtain additional help if required.

Note: If you have a Lenovo ID, you can register at the website.

License keys are provided as files that must be uploaded to the switch. To acquire an activation key, use the FoD website to obtain an Authorization Code. You will need to provide the unique ID (UID) of the specific switch where the key will be installed. The UID is the last 12 characters of the switch serial number. This serial number is located on the Part Number (PN) label and is also displayed during successful login to the device.

You can view the UID of a switch using the following command:

```
Switch> display license host-id

System serial number: Y052MV4CR026
```

When available, download the activation key file from the FoD site.

Installing License Keys

Once an FoD license key file have been acquired, it must be installed on the switch. The example below depicts the use of the switch Command Line Interface (CLI).

To install activation keys, use the following steps:

1. Log into the switch.

2. Enter Global Configuration command mode:

```
Switch> enable
Switch# configure device
Switch(config)#
```

3. You can install the license key by copying it from a remote server or a USB device:

   ● Remote FTP, SCP, SFTP, or TFTP server:

   ```
   Switch(config)# license install {ftp|scp|sftp|tftp} <server address and file
   path> vrf management
   ```

   For example, using a TFTP server:

   ```
   Switch(config)# license install tftp
   tftp://10.120.33.12/fod-keys/fodEE3 vrf management
   ```

   ● USB device:

   ```
   Switch(config)# license install usb1 <file path>
   ```

   For example:

   ```
   Switch(config)# license install usb1 fod-keys/fodEE3
   ```

4. If using a USB device, dismount it:

   ```
   Switch(config)# system eject-usb
   ```

   Note: Repeat this step when installing multiple license keys.

   The license key has now been successfully installed on the switch.

5. To view installed license keys, use the following command:

   ```
   Switch> display license brief
   ```

Uninstalling License Keys

If you wish to remove or disable a feature that is available only through the use of a license key, you can achieve this by using the following command:

```
Switch(config)# no license install <license file>
```

For example:

```
Switch(config)# no license install fodEE3
```

Note: When removing the ONIE license key from the switch, the ONIE software image is also removed.

Transferring License Keys

Licenses keys are based on the unique switch device serial number and are non-transferable.

In the event that the switch must be replaced, a new activation key must be acquired and installed. When the replacement is handled through IBM Service and Support, your original license will be transferred to the serial number of the replacement unit and you will be provided with a new license key.

ONIE License Key

The Open Network Install Environment (ONIE) is a small Linux-based operating system that provides an open install environment for networking devices without operating systems.

ONIE is pre-installed on the NE10032 and the NE2572 switches; it is not pre-installed on other switches, and it can be activated only using an FoD license key. The process of obtaining and installing an activation key is described in .

After you have successfully installed the appropriate FoD license key, you will be able to install ONIE on the switch. For more details on ONIE and how to install and use it, see the *Lenovo ONIE Quick Start Guide*.

Note: ONIE is not supported on the G8332.

Chapter 4. Switch Software Management

The switch software image is the executable code running on the device. A version of the image comes pre-installed on the device. As new versions of the image are released, you can upgrade the software running on your switch. To get the latest version of software supported for your switch, go to the following website:

http://support.lenovo.com

To determine the software version currently used on the switch, use the following switch command in Privileged EXEC mode:

```
Switch# display boot

Current FLASH software:
  active image: version 10.6.2.0, downloaded 02:05:34 UTC Sun May 15 2016
  standby image: empty
  Uboot: version 10.6.2.0, downloaded 02:05:36 UTC Sun May 15 2016
  ONIE: empty
Currently set to boot software active image
Currently scheduled reboot time: none
Current port mode: default mode
```

The typical upgrade process for the software image consists of the following steps:

1. Load a new software image and boot image onto an FTP, SFTP, SCP, or TFTP server on your network.

2. Transfer the new images to your switch.

3. Specify the new software image as the one which will be loaded into switch memory the next time a switch reload occurs.

4. Reload the switch.

   For instructions on the typical upgrade process using the CNOS ISCLI, see "Installing New Software to Your Switch" on page 98.

CAUTION:
Although the typical upgrade process is all that is necessary in most cases, upgrading from (or reverting to) some versions of Lenovo Cloud Network Operating System requires special steps prior to or after the software installation process. Please be sure to follow all applicable instructions in the release notes document for the specific software release to ensure that your switch continues to operate as expected after installing new software.

Installing New Software to Your Switch

The switch can store up to two different CNOS images (called `active` and `standby`) and a special boot software named U-boot. U-Boot is an open source, primary boot loader used to package the instructions needed to boot the switch operating system kernel.

When you install a new CNOS image, it will be placed in the standby image on the switch. When you reload the switch using the standby image, it is swapped with the active image. After the switch reloads, the current active image will become the new standby image, and the current standby image will become the new active image.

Note: U-boot is included in the CNOS image.

To load a new software image to your switch, you will need the following:

● The new software image loaded on a FTP, SCP, SFTP or TFTP server on your network.

● The IP address of the FTP, SCP, SFTP, or TFTP server.

● The name of the new system image or ONIE image.

Note: Before installing an ONIE image on the switch, you need to install the appropriate license key. For more details, see "System License Keys" on page 91.

When the software requirements are met, use the following procedures to download the new software to your switch using the ISCLI.

Installing System Images from a Remote Server

1. In Privileged EXEC mode, enter the following command:

```
Switch# cp {ftp|scp|sftp|tftp} <server address and file path> system-image os vrf
management
```

For example, using the TFTP protocol:

```
Switch# cp tftp tftp://10.120.33.12/nos-images/G8xxx-CNOS-10.6.2.0.imgs
system-image os vrf management
```

The exact form of the server address and the file path of the CNOS image will vary by server. However, the file location is normally relative to the FTP, SCP, SFTP, or TFTP directory (for example, `tftpboot`).

Note: If you use the parameter **all** instead of **os** you will also install the U-boot file included in the CNOS image. You will also get updates and fixes made to the boot and OS images, as well as error messages you can ignore.

2. If required by the FTP, SCP, or SFTP server, enter the appropriate username and password.

3. You will be prompted to confirm the download of the CNOS image:

```
Confirm download operation? (y/n) [n]
```

Once confirmed, the software will begin loading into the switch.

```
Confirm download operation? (y/n) [n] y
Download in progress
...................................................................
Copy Success

Install image...This takes about 90 seconds. Please wait
Valid image detected.
Check image signature succeeded
Extracting image: 100%
Installing system image to slot 2:
Installing image: 100%
Extracting image: 100%
OS image installation succeeded.

Standby image now contains Software Version 10.6.2.0
```

4. The new CNOS version is installed as the standby image. Once the installation is successful, you are asked if you want the switch to boot next time using the new image.

```
Switch is currently set to boot active image.
Do you want to change that to the standby image? (y/n) [n] y
Switch is to be booted with standby image.
```

5. Reboot the switch to run the new CNOS version:

```
Switch# reload
```

The system prompts you to confirm your request. Once confirmed, the switch will reboot to use the new CNOS image.

Note: Any unsaved configuration changes will be lost once the switch starts reloading.

Installing System Images from a USB Device

1. Use the following command to copy a CNOS image from a USB drive to the switch:

   ```
   Switch# cp usb1 <file path> system-image os
   ```

   In this example, a CNOS image is copied from a directory on a USB drive:

   ```
   Switch# cp usb1 nos-images/G8xxx-CNOS-10.6.2.0.imgs system-image os
   ```

   Note: If you use the parameter **all** instead of **os** you will also install the U-boot file included in the CNOS image. You will also get updates and fixes made to the boot and OS images, as well as error messages you can ignore.

2. You will be prompted to confirm the download of the CNOS image:

   ```
   Confirm download operation? (y/n) [n]
   ```

   Once confirmed, the software will begin loading into the switch.

   ```
   Confirm download operation? (y/n) [n] y
   Download in progress
   .....................................................................
   Copy Success

   Install image...This takes about 90 seconds. Please wait
   Valid image detected.
   Check image signature succeeded
   Extracting image: 100%
   Installing system image to slot 2:
   Installing image: 100%
   Extracting image: 100%
   OS image installation succeeded.
   Standby image now contains Software Version 10.6.2.0
   ```

3. The new CNOS version is installed as the standby image. Once the installation is successful, you are asked if you want the switch to boot next time using the new image.

   ```
   Switch is currently set to boot active image.
   Do you want to change that to the standby image? (y/n) [n] y
   Switch is to be booted with standby image.
   ```

4. Dismount the USB drive:

   ```
   Switch(config)# system eject-usb
   ```

5. Reboot the switch to run the new software:

   ```
   Switch# reload
   ```

   The system prompts you to confirm your request. Once confirmed, the switch will reboot to use the new software.

   Note: Any unsaved configuration changes will be lost once the switch starts reloading.

Installing U-boot from a Remote Server

1. In Privileged EXEC mode, enter the following command:

```
Switch# cp {ftp|scp|sftp|tftp} <server address and file path> system-image boot vrf
management
```

For example, using the TFTP protocol:

```
Switch# cp tftp tftp://10.120.33.12/nos-images/G8xxx-CNOS-10.6.2.0.imgs
system-image boot vrf management
```

Note: If you use the parameter **all** instead of **boot** you will also install the CNOS image. You will also get updates and fixes made to the boot and OS images, as well as error messages you can ignore.

The exact form of the server address and the file path of the U-boot file will vary by server. However, the file location is normally relative to the FTP, SCP, SFTP, or TFTP directory (for example, tftpboot).

2. If required by the FTP, SCP, or SFTP server, enter the appropriate username and password.

3. You will be prompted to confirm the download of the U-boot file:

```
Confirm download operation? (y/n) [n]
```

Once confirmed, the software will begin loading into the switch.

```
Confirm download operation? (y/n) [n] y
Download in progress
......................................................................
Copy Success

Install image...This takes about 90 seconds. Please wait
Valid image detected.
Check image signature succeeded
Extracting image: 100%
Installing Boot Loader:
Updating flash: 100%
Boot image installation succeeded.
Boot loader now contains Software Version 10.6.2.0
```

On the next reload, the switch will use the new U-boot file.

Installing U-boot from a USB Device

1. Use the following command to copy a U-boot file from a USB drive to the switch:

```
Switch# cp usb1 <file path> system-image boot
```

In this example, a U-boot file is copied from a directory on the USB drive:

```
Switch# cp usb1 nos-images/G8xxx-CNOS-10.6.2.0.imgs system-image boot
```

Note: If you use the parameter **all** instead of **boot** you will also install the CNOS image.

2. You will be prompted to confirm the download of the software image:

```
Confirm download operation? (y/n) [n]
```

Once confirmed, the software will begin loading into the switch.

```
Confirm download operation? (y/n) [n] y
Download in progress
.........................................................................
Copy Success

Install image...This takes about 90 seconds. Please wait
Valid image detected.
Check image signature succeeded
Extracting image: 100%
Installing Boot Loader:
Updating flash: 100%
Boot image installation succeeded.
Boot loader now contains Software Version 10.6.2.0
```

On the next reload, the switch will use the new U-boot file.

Selecting a Software Image to Run

The switch can store up to two CNOS images. These images are referred to as the active and standby images. The active image is the image that the switch is currently using when rebooting. The standby image is the image that the switch keeps as backup, in case of a failure when booting the active image.

To see what CNOS versions are stored in each image and what image is configured to be used on the next reload, use the following command:

```
Switch# display boot

Current FLASH software:
  active image: version 10.6.2.0, downloaded 11:45:39 UTC Mon Feb 15 2016
  standby image: version 10.6.2.0, downloaded 14:44:26 UTC Tue Feb 9 2016
  Uboot: version 10.6.2.0, downloaded 11:17:04 UTC Wed Nov  4 2015
Currently set to boot software active image
Currently scheduled reboot time: none
Current port mode: default mode
```

You can choose which CNOS image the switch will use on the next reload by running the following command:

```
Switch(config)# startup image {active|standby}
```

For example:

```
Switch(config)# startup image standby

Switch# display boot

Current FLASH software:
  active image: version 10.6.2.0, downloaded 11:45:39 UTC Mon Feb 15 2016
  standby image: version 10.6.2.0, downloaded 14:44:26 UTC Tue Feb 9 2016
  Uboot: version 10.6.2.0, downloaded 11:17:04 UTC Wed Nov  4 2015
Currently set to boot software standby image
Currently scheduled reboot time: none
Current port mode: default mode
```

The switch will now reload using the standby image. After reloading, the switch interchanges the CNOS images. The standby image before the reload becomes the new active image and vice-versa.

Reloading the Switch

You can reboot the switch to make your CNOS image files changes occur.

Normal Reboot

To reboot the switch, use the following command:

```
Switch# reload
```

You are prompted for confirmation:

```
Switch# reload

reboot system? (y/n):
```

After confirming, the switch will halt and perform a reboot. Any opened sessions will be closed. After the reload is complete, you'll have to log in the switch.

```
Switch# reload
reboot system? (y/n): y

Broadcast message from root@switch (Mon Feb 15 15:47:06 2016):

The system ireboot: Restarting system
...
G8xxx login:
```

Note: Any unsaved configurations will be lost once the reboot process starts.

Scheduled Boot

The scheduled boot feature lets you schedule a switch reload in a future time, enabling you to perform switch upgrades during off-peak hours. To set up a scheduled reboot, enter:

```
Switch# reload schedule {monday | tuesday | wednesday | thursday | friday
| saturday | sunday} HH:MM [reason <reason>]
WARNING: There is unsaved configuration!!!
Confirm scheduled reboot ? (y/n): y
```

where:

Parameter	Description
monday	Reboot on Monday.
tuesday	Reboot on Tuesday.
wednesday	Reboot on Wednesday.
thursday	Reboot on Thursday.
friday	Reboot on Friday.

Parameter	Description
saturday	Reboot on Saturday.
sunday	Reboot on Sunday.
HH:MM	Hours and minutes in 24-hour format
reason	(Optional) The reason for the reboot

If there is unsaved information, an error message occurs. To confirm the reboot, enter **y**. Note that any unsaved information will be lost once the reboot starts.

To cancel a scheduled reboot, enter:

```
Switch# reload schedule cancel
```

To display the boot schedule state, enter:

```
Switch# display boot
```

Note: If the switch is rebooted prior to the scheduled reboot time, the previously scheduled reboot will be removed from the configuration, effectively cancelling the scheduled reboot.

Copying Configuration Files

You can also copy running and startup configuration files to or from the switch. This can be done via a remote server or a USB device.

Note: You cannot copy to the switch configuration files that exceed 4 MB in size or that have a line count larger than 100,000 lines.

Copy Configuration Files via a Remote Server

You can use the ISCLI to copy configurations files from or to a remote FTP, SCP, SFTP, or TFTP server.

Use the following command to copy the startup configuration file from a remote server:

```
Switch# cp {ftp|scp|sftp|tftp} [<server address and file path>] startup-config
```

For example, using the TFTP protocol:

```
Switch# cp tftp tftp://10.120.33.12/conf/mystartup.cfg startup-config
```

Use the following command to copy the running configuration file from a remote server:

```
Switch# cp {ftp|scp|sftp|tftp} [<server address and file path>] running-config
```

For example, using the TFTP protocol:

```
Switch# cp tftp tftp://10.120.33.12/conf/mystartup.cfg startup-config
```

Use the following command to copy the startup configuration file to a remote server:

```
Switch# cp startup-config {ftp|scp|sftp|tftp} <server address and file path>
```

For example, using the TFTP protocol:

```
Switch# cp tftp tftp://10.120.33.12/conf/mystartup.cfg startup-config
```

Use the following command to copy the running configuration file to a remote server:

```
Switch# cp running-config {ftp|scp|sftp|tftp} <server address and file path>
```

For example, using the TFTP protocol:

```
Switch# cp tftp tftp://10.120.33.12/conf/myrunning.cfg running-config
```

Copy Configuration Files to a USB Device

You can insert a USB drive into the USB port on the switch. You can copy files to the USB drive.

Use the following command to copy a configuration file to the USB drive:

```
Switch# cp {running-config|startup-config} usb1 <file path>
```

In this example, the running configuration file is copied to a directory on the USB drive:

```
Switch# cp running-config usb1 configuration/myconfig.cfg
```

To dismount a USB device, enter:

```
Switch(config)# system eject-usb
```

Resetting the Switch to the Factory Defaults

To reset the switch to its factory default settings, use the following procedure:

1. Erase the switch start-up configuration:

```
Switch# save erase

Warning: This command will erase the startup-configuration.
Do you wish to proceed anyway? (y/n)  [n] y
```

Enter **y** at the prompt to confirm the erasing process.

2. Reload the switch:

```
Switch# reload
```

Converting the Switch Software Image from CNOS to ENOS

If you want to return to the legacy 8.x ENOS image, you must reinstall it.

Notes:

- For the G8296 and the G8332, Lenovo Cloud Network OS image cannot coexist on the switch with your legacy 8.x Lenovo NOS. All previous configuration will be lost.

- For the G8272, Lenovo offers the capability to store both CNOS and ENOS images on the switch. The first time the switch reboots with ENOS, the CNOS configuration is set to factory default. If you already installed the legacy 8.x ENOS image on your switch, please go to .

- The ThinkSystem switches only run Cloud NOS; they do not run Lenovo Enterprise NOS.

1. Download the ENOS boot image to the switch and enter **y** to confirm the download operation.

```
Switch# cp tftp tftp://<tftp server IP>/G8296-8.4.2.0_Boot.imgs system-image
boot vrf management
Confirm download operation? (y/n) [n] y
Download in progress
..........
Copy Success

Install image...This takes about 90 seconds. Please wait
Check image signature succeeded
Extracting image: 100%
Installing eNOS RFS:
Updating flash: 100%
Extracting image: 100%
Installing eNOS Kernel:
Updating flash: 100%
Extracting image: 100%
Installing eNOS DFT:
Updating flash: 100%
Extracting image: 100%
Installing uboot:
Updating flash: 100%
Boot image installation succeeded.
```

2. Download the ENOS system image to the switch and enter **y** to confirm the download operation:

```
Switch# cp tftp tftp://<tftp server IP>/G8296-8.4.2.0_OS.imgs system-image os
vrf management
Confirm download operation? (y/n) [n] y
Download in progress
..........
Copy Success
Install image...This takes about 90 seconds. Please wait
Check image signature succeeded
Extracting image: 100%
Installing E-NOS image to image1
Installing image to E-NOS image1: 100%
OS image installation succeeded.
```

Notes:

● You can also install the boot and system images using the following commands:

```
Switch# cp tftp tftp://<tftp server IP>/<switch_model>-8.4.2.0_Boot.imgs
system-image all vrf management
```

```
Switch# cp tftp tftp://<tftp server IP>/<switch_model>-8.4.2.0_OS.imgs
system-image all vrf management
```

● If you use the all argument instead of boot or os, an "image installation failed" message will appear for whichever type of image you are not installing. You can ignore these error messages.
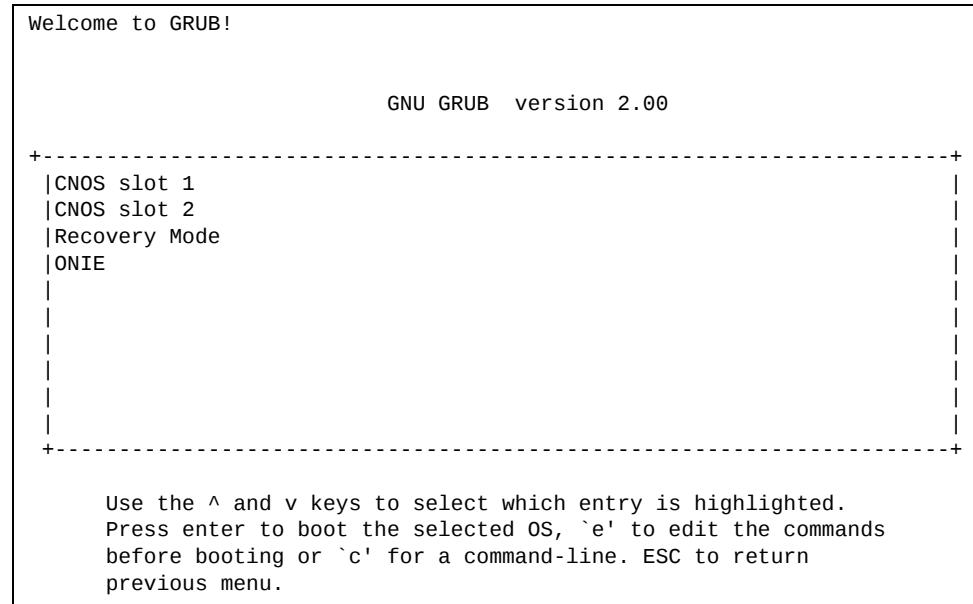
3. Reboot the switch.

```
Switch# reload
```

The NE10032/NE2572 GRUB Menu

The ThinkSystem NE10032/NE2572 RackSwitch does not use a Boot Management menu. Instead, all images are preloaded, and all booting options, such as switching boot images, are included in the GNU GRUB bootloader. To boot CNOS or switch images on the NE10032/NE2572:

1. Boot the switch. Eventually the GNU GRUB bootloader displays. Select the image you want to use (either CNOS slot 1 or CNOS slot 2).

```
Welcome to GRUB!


                        GNU GRUB  version 2.00


+----------------------------------------------------------------------+
|CNOS slot 1                                                           |
|CNOS slot 2                                                           |
|Recovery Mode                                                         |
|ONIE                                                                  |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
+----------------------------------------------------------------------+


     Use the ^ and v keys to select which entry is highlighted.
     Press enter to boot the selected OS, `e' to edit the commands
     before booting or `c' for a command-line. ESC to return
     previous menu.
```

2. The system will automatically load CNOS and prompt you to log in.

NE10032/NE2572 Rescue Mode

The Rescue Mode option allows you to recover from a failed software or boot image upgrade using TFTP or a USB drive.

To enter Rescue Mode, select **Recovery Mode** in the GRUB menu. The following menu appears.

```
Entering Rescue Mode.
Please select one of the following options:
        T) Configure networking and tftp download an image
        U) Install image from USB stick
        F) Run filesystem check
        I) Select which image to boot
        C) Reset configuration to factory default
        Z) Reset the Network Administrator (admin) password
        B) Reset the password required to enter Privileged EXEC mode
        R) Reboot
        E) Exit
```

The Rescue Mode menu allows you to perform the following actions:

- To recover from a failed software or boot image upgrade using TFTP, press **T** and follow the screen prompts. For more details, see "Recovering from a Failed Image Upgrade using TFTP" on page 116.

- To recover using an install image from a USB stick, press **U**.

- To check if the switch is ready to run Cloud NOS software, press **F**. It performs a check to see if the filesystem is optimally partitioned and updates it accordingly.

- To select which image to boot, press **I**.

- To reset the switch configuration to factory defaults, press **C**.

- To reset the Network Administrator account (admin) password, press **Z**.

- To reset the password required to enter Privileged EXEC mode, press **B**.

- To restart the reload process from the beginning, press **R**.

- To exit the Rescue Mode menu, press **E**.

The Boot Management Menu

Note: This section does not apply to the NE10032/NE2572.

The Boot Management menu allows you to switch the NOS image, reset the default user password, or recover from a failed software installation.

You can interrupt the startup process and enter the Boot Management menu from the serial console port. When the system displays the following message, press **Shift + B**. The Boot Management menu displays.

```
The system is going down for reboot NOW!
INIT: reboot: Restarting system

...

Press shift-B for startup menu or shift-R for recovery mode: ..
Running Startup Menu

...

Boot Management Menu
Boot Management Menu
        M - Change boot mode (ENOS vs CNOS)
        I - Change booting image
        C - Change configuration to factory default
        R - Boot in recovery mode (tftp and xmodem download of images to
recover switch)
        P - Reset the Network Administrator (admin) password
        B - Reset the password required to enter privileged exec mode
        Q - Reboot
        E - Exit
Please choose your menu option:
```

The Boot Management menu allows you to perform the following actions:

- For the G8272, to switch between ENOS and CNOS images, enter M and follow the screen prompts. ENOS refers to Lenovo Enterprise Network OS and CNOS refers to Lenovo Cloud Network OS. For further details, see "Switching Between ENOS and CNOS Images Loaded on the G8272" on page 114.

- To change the booting image, enter I and follow the screen prompts.

- To boot in recovery mode, enter R.

- To reset the default user password, enter P and follow the screen prompts.

- To enter the ONIE submenu, enter O.

Note: ONIE is not supported on the G8332.

- To reset the Network Administrator account (admin) password, enter P.

- To reset the password required to enter Privileged EXEC mode, enter B.

- To reload the switch, enter Q. The reloading process will start again.

- To exit the Boot Management menu, enter E. The reloading process continues.

Note: The ONIE submenu will only be available if you have activated ONIE by installing an appropriate license key. For more details, see Chapter 3, "System License Keys".

Switching Between ENOS and CNOS Images Loaded on the G8272

If you have both ENOS and CNOS images loaded on your device, you can switch from one to another by using the Boot Management menu. To do so, follow these steps:

1. Reboot the switch.

```
Switch# reload
```

2. When the phrase "Booting Lenovo NOS" appears, quickly press **B** (**Shift**+**b**) to call the Boot Management Menu.

```
Boot Management Menu
        M - Change boot mode (ENOS vs CNOS)
        I - Change booting image
        R - Boot in recovery mode (tftp and xmodem download of images to
recover switch)
        P - Password reset
        O - ONIE boot menu
        Q - Reboot
        E - Exit
Please choose your menu option:
```

3. Enter **M** to change the boot mode.

```
Please choose your menu option: M
```

4. Enter **E** to enter ENOS Boot Mode or **C** to enter CNOS Boot Mode.

```
Currently booting in CNOS mode
Enter boot mode: ENOS or CNOS: E
Boot Management Menu
        M - Change boot mode (ENOS vs CNOS)
        I - Change booting image
        C - Change configuration block
        R - Boot in recovery mode (tftp and xmodem download of images to
recover switch)
        O - ONIE boot menu
        Q - Reboot
        E - Exit
Please choose your menu option:
```
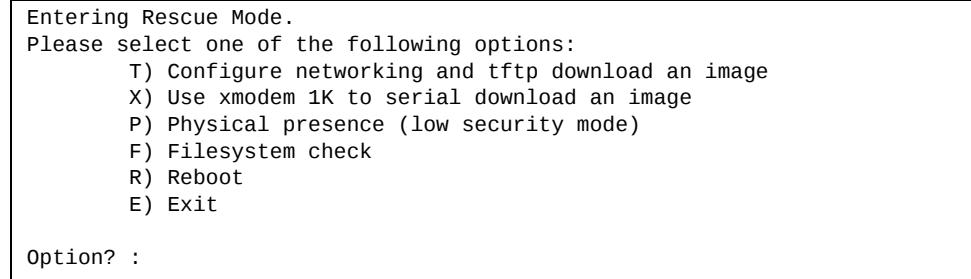
5. At the Boot Management Menu, enter **Q** to reboot.

   Note: All previous CNOS/ENOS configurations are stored and re-applied when switching back to the corresponding OS.

Boot Recovery Mode

The Boot Recovery Mode allows you to recover from a failed software or boot image upgrade using TFTP or XModem download.

To enter Boot Recovery Mode you must select the `Startup in recovery mode` option from the Boot Management Menu.

```
Entering Rescue Mode.
Please select one of the following options:
        T) Configure networking and tftp download an image
        X) Use xmodem 1K to serial download an image
        P) Physical presence (low security mode)
        F) Filesystem check
        R) Reboot
        E) Exit

Option? :
```

The Boot Recovery Mode menu allows you to perform the following actions:

- To recover from a failed software or boot image upgrade using TFTP, press **<T>** and follow the screen prompts. For more details, see "Recovering from a Failed Image Upgrade using TFTP" on page 116.

- To recover from a failed software or boot image upgrade using XModem download, press **<X>** and follow the screen prompts. For more details, see "Recovering from a Failed Image Upgrade using XModem Download" on page 118.

- To enable the loading of a special image, press **<P>** and follow the screen prompts. For more details, see "Physical Presence" on page 119.

- To check if the switch is ready to run Cloud NOS software, press **<F>**. It performs a check to see if the filesystem is optimally partitioned and updates it accordingly.

- To restart the reload process from the beginning, press **<R>**.

- To exit Boot Recovery Mode menu, press **<E>**.

Recovering from a Failed Image Upgrade using TFTP

Use the following procedure to recover from a failed image upgrade using TFTP:

1. Connect a PC to the console port of the switch.

2. Open a terminal emulator program that supports the Telnet protocol (for example, HyperTerminal, SecureCRT, or PuTTY) and input the proper hostname or IP address and the port to connect to the console interface of the switch.

3. Reload the switch and access the Boot Management menu by pressing **<Shift + B>** when the following message appears and the dots are being displayed.

```
Press shift-B for startup menu or shift-R for recovery mode: ...
```

4. Enter Boot Recovery Mode by selecting **<R>**. The Recovery Mode menu will appear.

5. To start the recovery process using TFTP, select **<T>**. The following message will appear:

```
Performing TFTP rescue. Please answer the following questions (enter 'q'
to quit):
```

6. Enter the IP address of the management port:

```
IP addr :
```

7. Enter the network mask of the management port:

```
Netmask :
```

8. Enter the gateway of the management port:

```
Gateway :
```

9. Enter the IP address of the TFTP server:

```
Server addr:
```

10. Enter the file path and the filename of the image:

```
Image Filename:
```

After the procedure is complete, the Recovery Mode menu will be redisplayed.

Following is an example of a successful recovery procedure using TFTP:

```
Entering Rescue Mode.
Please select one of the following options:
        T) Configure networking and tftp download an image
        X) Use xmodem 1K to serial download an image
        P) Physical presence (low security mode)
        F) Filesystem check
        R) Reboot
        E) Exit

Option? : t
Performing TFTP rescue. Please answer the following questions (enter 'q'
to quit):
IP addr :10.241.6.4
Netmask :255.255.255.128
Gateway :10.241.6.66
Server addr:10.72.97.135
Image Filename: G8xxx-CNOS-10.6.2.0.imgs
        Netmask : 255.255.255.128
        Gateway : 10.241.6.66
Configuring management port......................
Installing image G8xxx-CNOS-10.6.2.0.imgs from TFTP server 10.72.97.135

Extracting images ... Do *NOT* power cycle the switch.
Installing Application: Image signature verified.
Installing image as standby image: 100%

Standby image update succeeded
Updating install log. File G8xxx-CNOS-10.6.2.0.imgs installed from
10.72.97.135 at 15:29:30 on 12-3-2015
Please select one of the following options:
        T) Configure networking and tftp download an image
        X) Use xmodem 1K to serial download an image
        P) Physical presence (low security mode)
        F) Filesystem check
        R) Reboot
        E) Exit

Option? :
```

Recovering from a Failed Image Upgrade using XModem Download

Use the following procedure to recover from a failed image upgrade.

1. Connect a PC to the serial port of the switch.

2. Open a terminal emulator program that supports Xmodem download (for example, HyperTerminal, SecureCRT, or PuTTY) and select the following serial port characteristics:

   - Speed: 9600 bps
   - Data Bits: 8
   - Stop Bits: 1
   - Flow Control: None

3. Reload the switch and access the Boot Management menu by pressing **Shift + B** when the following message appears and the dots are being displayed.

   ```
   Press shift-B for startup menu or shift-R for recovery mode: ...
   ```

4. Enter Boot Recovery Mode by selecting R. The Recovery Mode menu will display.

5. Select X for Xmodem download. You will see the following display:

   ```
   Running xmodem rescue.....
   ```

6. When you see the following message, change the Serial Port speed to 115200 bps:

   ```
   Change the baud rate to 115200 bps and hit the <ENTER> key before
   initiating the download.
   ```

7. Press **Enter** to set the system into download accept mode. When the readiness meter displays (a series of "C" characters), start Xmodem on your terminal emulator. You will see a display similar to the following:

   ```
   ... Waiting for the <Enter> key to be hit before the download can start...
   CC
   ```

8. Select the image to download. Xmodem initiates the file transfer. When download is complete, you are asked to change the Serial Port speed back to 9600 bps:

   ```
   Change the baud rate back to 9600 bps, hit the <ENTER> key
   ```

9. Press **Enter** to start installing the image.

The image install will begin. After the procedure is complete, the Recovery Mode menu will be redisplayed.

```
Extracting images ... Do *NOT* power cycle the switch.
Installing Root Filesystem:
Image signature verified.  100%
Installing Kernel:
Image signature verified.  100%
Installing Device Tree:
Image signature verified.  100%
Installing Boot Loader:  100%
Updating install log. File image installed from xmodem at 18:06:02 on
13-3-2015
Please select one of the following options:
        T) Configure networking and tftp download an image
        X) Use xmodem 1K to serial download an image
        P) Physical presence (low security mode)
        F) Filesystem check
        R) Reboot
        E) Exit

Option? :
```

Boot image recovery is complete.

Physical Presence

Use the following procedure to enable the installation of special images on the switch, when a deeper troubleshooting analysis is required:

1. Connect a PC to the console port of the switch.

2. Open a terminal emulator program that supports a serial port connection and select the following serial port characteristics:

   - Default Baud Rate:

     - G8332, G8296, G8272, NE1032, NE1032T, NE1072$ 9600 bps
     - NE10032, NE2572:   115200 bps

   - Data Bits: 8

   - Stop Bits: 1

   - Parity: None

   - Flow Control: None

3. Boot the switch and access the Boot Management menu by pressing **Shift + B** when the following message appears and the dots are being displayed.

```
Press shift-B for startup menu or shift-R for recovery mode: ...
```

4. Enter Boot Recovery Mode by selecting R. The Recovery Mode menu will appear.

5. To begin the Physical Presence procedure, select P. The following warning message will appear and you will be prompted for confirmation (enter y to confirm):

```
WARNING: the following test is used to determine physical presence and if
completed will put the switch in low security mode.

Do you wish to continue y/n? y
```

6. A security test will be performed. The system location (blue) LED will blink a number of times between 1 and 12. Enter that number:

```
Hit a key to start the test. The blue location LED will blink a number of
times.

...........
How many times did the LED blink?
```

7. After entering the correct number, the Recovery Mode menu will reappear. To install a special image, use one of the following procedures:

- TFTP (for details, see page 116)
- XModem Download (for details, see page 118)

Note: You have three attempts to successfully complete the security test. After three incorrect attempts, the switch will reboot.

After the test is completed, the switch will be put in low security mode. This mode will allow you to install special images on the switch. To revert to normal security mode, you must reboot the switch or select P again in the Recovery Mode menu.

ONIE Submenu

In the ONIE (Open Network Install Environment) submenu you can choose from different ONIE operations such as installing and uninstalling NOS images, upgrading ONIE, and using ONIE rescue and recovery mode to troubleshoot a broken system.

Note: ONIE is not supported on the G8332.

For more detailse about ONIE, see "ONIE" on page 122.

Note: The ONIE submenu will only be available if you have activated ONIE by installing an appropriate license key. For more details, see Chapter 3, "System License Keys".

To enter the ONIE submenu you must select the "ONIE submenu" option from the Boot Management Menu.

```
ONIE Menu
    I - Boot ONIE OS installer
    N - Boot NOS mode (system default)
    R - Boot ONIE rescue mode
    U - Boot ONIE self update mode
    D - Boot ONIE OS uninstaller
    E - Exit ONIE menu

Please choose your menu option:
```

The ONIE submenu allows you to perform the following actions:

- To boot the switch in ONIE install mode, enter I.
- To boot the switch using the installed NOS image, enter N.
- To boot the switch in ONIE rescue mode, enter R.
- To boot the switch in ONIE update mode, press U.
- To boot the switch in ONIE uninstall mode, press D.
- To exit the ONIE submenu and return to the Boot Management Menu, enter E.

For more information about the ONIE submenu, see the *ONIE Quick Start Guide* for your switch.

ONIE

The Open Network Install Environment (ONIE) is a small Linux-based operating system that provides an open install environment for networking devices without operating systems.

ONIE enables a network switch ecosystem for end-users to choose among different Network Operating Systems (NOS). Practically, ONIE boots on a switch, discovers NOS installer images available on the local network or USB drive, copies the chosen image to the switch, and provides an environment where the installer can load the NOS onto the switch.

The switch is compatible with ONIE. The NE10032 and the NE2572 switches come with ONIE pre-installed. To use ONIE on the G8296, G8272, NE1032, NE1032T, or NE1072T, you must first install a license key. For more details on how to obtain and install such a license, see Chapter 3, "ONIE License Key".

For more details on ONIE and how to install and use it, see the *Lenovo ONIE Quick Start Guide*.

Note: ONIE is not supported on the G8332.

Part 2: Securing the Switch

This section discusses the following switch security topics:

Chapter 5. Securing Administration

Secure switch management is needed for environments that perform significant management functions across the Internet. Common functions for secured management are described in the following sections:

- "Secure Shell and Secure Copy" on page 126

- "End-user Access Control" on page 130

Note: SNMP read and write functions are enabled by default. For best security practices, if SNMP is not needed for your network, we recommend that you disable these functions prior to connecting the switch to the network (see Chapter 30, "Simple Network Management Protocol").

Secure Shell and Secure Copy

Because using Telnet does not provide a secure connection for managing the switch, Secure Shell (SSH) and Secure Copy (SCP) features have been included for switch management. SSH and SCP use secure tunnels to encrypt and secure messages between a remote administrator and the switch.

SSH is a protocol that enables remote administrators to log securely into the switch over a network to execute management commands. The SSH module consists of a SSH server and a SSH client. By default, the SSH server listens on TCP port number 22. Once a SSH client establishes a connection to the SSH server, the client acts as a virtual terminal through which the remote system is accessed.

SCP is typically used to copy files securely from one machine to another. SCP uses SSH for encryption of data on the network. On the switch, SCP is used to download and upload the switch configuration via secure channels.

SSH is enabled by default and using this features provides the following benefits:

- Identifying the administrator using Name/Password
- Authentication of remote administrators
- Encrypting messages between the remote administrator and the switch
- Secure copy support

Lenovo Cloud Network Operating System implements the SSH version 2.0 standard and is confirmed to work with SSH version 2.0-compliant clients such as the following:

- OpenSSH_5.4p1 for Linux
- Secure CRT Version 5.0.2 (build 1021)
- Putty SSH release 0.60

SSH Encryption and Authentication

The following encryption and authentication methods are supported for SSH:

- Server Host Authentication: Client RSA authenticates the switch at the beginning of every connection
- Key Exchange: RSA, DSA
- Encryption: 3DES-CBC, DES
- User Authentication: Local password authentication, TACACS+

Generating RSA/DSA Host Key for SSH Access

To support the SSH host feature, a RSA or DSA host key is required. The host key is 2048 bits long and is used to identify the switch.

To configure a new RSA or DSA host key, connect to the switch through the console port or via a Telnet session, and enter the following command to manually generate the host key:

- To generate a new DSA key:

```
Switch(config)# ssh key dsa
```

- To generate a new RSA key:

```
Switch(config)# ssh key rsa
```

When the switch reboots, it will retrieve the host key from the FLASH memory.

Note: The switch will perform only one session of key/cipher generation at a time. Thus, a SSH/SCP client will not be able to log in if the switch is performing key generation at that time. Also, key generation will fail if a SSH/SCP client is logging in at that time.

SSH Integration with TACACS+ Authentication

SSH is integrated with TACACS+ authentication. After the TACACS+ server is enabled on the switch, all subsequent SSH authentication requests will be redirected to the specified TACACS+ server for authentication. The redirection is transparent to the SSH client.

Configuring SSH on the Switch

To configure SSH parameters, use to following procedure:

1. Configure the number of unsuccessful SSH login attempts (the default value is 3):

```
Switch(config)# ssh login-attempts <1-10>
```

2. Optionally, you can change the SSH server port (the default port is 22):

```
Switch(config)# ssh server port <1-65535>
```

Note: To change the number of SSH login attempts and server port, the SSH feature must be disabled.

3. Optionally, you can configure a DSA or RSA host key:

```
Switch(config)# ssh key {dsa|rsa}
```

4. Check the current SSH server settings:

```
Switch# display ssh server
```

Using SSH Client Commands

This section shows the format for using some client commands.

To log onto an SSH server:

1. Connect via IPv4 or IPv6:

   - IPv4:

     ```
     Switch# ssh <username>@<server IPv4 address>
     ```

     Example:

     ```
     Switch# ssh admin@205.178.15.157
     ```

   - IPv6:

     ```
     Switch# ssh6 <server IPv6 address>
     ```

     Example:

     ```
     Switch# ssh6 FE80::0202:B3FF:FE1E:8329
     ```

2. Optionally, you can specify the SSH server port used to establish the connection:

   ```
   Switch# ssh <username>@<server IPv4 address> port <1-65535>
   ```

3. Optionally, you can also specify the Virtual Routing and Forwarding (VRF) instance used for the session:

   ```
   Switch# ssh <username>@<server IPv4 address> port <1-65535> vrf {default|
   |management}
   ```

Using Secure Copy

You can use SCP to copy files from or to a remote server. This includes the copying of the running and startup configurations, or technical support files.

Copying a File Using SCP

To copy a file to a remote server using SCP, use the following command:

```
Switch# cp file <filename> scp
```

To copy a file from a remote server using SCP, use the following command:

```
Switch# cp scp file <filename>
```

Copying the Startup Configuration Using SCP

To copy the startup configuration file to a remote server using SCP, use the following command:

```
Switch# cp startup-config scp
```

To copy the startup configuration file from a remote server using SCP, use the following command:

```
Switch# cp scp startup-config
```

Copying the Running Configuration Using SCP

To copy the running configuration file to a remote server using SCP, use the following command:

```
Switch# cp running-config scp
```

Copying Technical Support Files Using SCP

To copy the technical support dump file to a remote server using SCP, use the following command:

```
Switch# cp tech-support scp
```

End-user Access Control

Cloud NOS allows an administrator to define accounts that permit end-users to perform switch operations via CLI commands. Once end-user accounts are configured and enabled, the switch requires username/password authentication.

For example, an administrator can assign a user, who can then log into the switch and perform operational commands (effective only until the next switch reboot).

Considerations for Configuring End-user Accounts

Note the following considerations when you configure end-user accounts:

- A maximum of 100 user IDs are supported on the switch

- CNOS offers end-user support for console, Telnet, and SSH access to the switch

- The length of the username must be between 2 and 28 alphanumeric characters

- Passwords for end-users must be between 8 and 80 alphanumeric characters in length for TACACS+, Telnet, SSH, and console access

Strong Passwords

Cloud NOS requires the use of strong passwords for users to access the switch. Strong passwords enhance security because they make password guessing more difficult.

The following rules must be followed when creating a password for a user:

- Minimum length: 8 characters; maximum length: 80 characters

- Must contain at least one uppercase letter

- Must contain at least one lowercase letter

- Must contain at least one number

- Cannot be same as the username

- When changing passwords no four consecutive characters can be the same as in the old password

A user without a password will not be allowed to log in on the switch.

User Access Control

The end-user access control commands allow you to configure end-user accounts. Any changes to a user will only take place once that user logs out and initiates a new session.

To displays the currently logged in user, use the following command:

```
Switch# display users
```

Setting up Users

Up to 100 users can be configured on the switch. Use the following commands to define a username and set the user password:

1. Create the user:

```
Switch(config)# username <username>
```

Note: The username must be between 2 and 28 alphanumeric characters long.

2. Define a password for the user:

```
Switch(config)# username <username> password <password>
```

Note: The password must be between 8 and 80 alphanumeric characters long and must contain at least one uppercase letter and one number.

3. Check the current user accounts:

```
Switch# display user-accounts
```

The following words are reserved and cannot be used as usernames:

Table 4. *Words Not Permitted as Usernames*

adm	bin	daemon	ftp	ftpuser
games	gdm	gopher	halt	lp
mail	mailnull	man	mtsuser	news
shutdown	sync	sys	uucp	xfs
root	vsca	lighttpd	tcpdump	oprofile
ntp	dhcpd	sshd	nosx	obs
proxy	backup	list	irc	gnats
tss	cyrus	messagebus		

Defining a User's Access Level

Every user account will have a specific access level or role attached. The user role defines what operations the user will be allowed to perform on the switch. The user can have one of two roles:

- `network-admin`: complete read and write access to the switch
- `network-operator`: only read access to the switch

To view the current settings for each role, use the following command:

```
Switch# display role
```

To change a user's role, use the following command:

```
Switch(config)# username <username> role {network-admin|network-operator}
```

Note: If a user is created without specifying an access level, the default user role will be network-operator.

To view the current user roles for each user, use the following command:

```
Switch# display user-accounts
```

Deleting a User

To delete a user:

1. Delete the user:

```
Switch(config)# no username <username>
```

2. Verify if the user has been removed:

```
Switch# display user-accounts
```

The Default User

The user `admin` is automatically installed on the switch by default. This is used to initially set up the switch.

Its default role is `network-admin` and the default password is `admin`.

The `admin` user account cannot be removed, nor its user role changed. However, the password associated with this account can be modified by using the following command:

```
Switch(config)# username admin password <new password>
```

Password History Checking

With this option enabled, when setting up a new password for a user, it will be checked against older passwords used for that account. If the new password matches any of the previously four passwords, it will not be accepted by the switch and another password must be provided.

Note: Password History Checking applies only to the switch local user database. If access to the device is granted through the use of Authentication, Authorization, and Accounting (AAA), password history checking isn't available. For more details about AAA, see "Authentication, Authorization, and Accounting" on page 149.

To enable or disable password history checking, use the following command:

```
Switch(config)# [no] password history-checking
```

By default, password history checking is disabled.

When this option is enabled on the switch, checking new passwords against older ones is enforced under any situation, expect the following:

● during switch initialization

Password history checking will be bypassed to prevent errors that may be caused by this action when the switch configuration is loading.

● during administrative password recovery

Password history checking will be bypassed to ensure that the recovery of the administrator password is successful. For more details about administrative password recovery, see "Administrator Password Recovery" on page 134.

A history of the previous used passwords will be created for each individual user. When deleting a user from the switch local database, its history is also deleted.

Notes:

● When deleting the switch configuration, all password histories stored on the device will also be deleted. This also happens when password history checking is disabled on the switch.

● However, the password histories are persistent across switch reloads and software image upgrades.

Administrator Password Recovery

In case you have changed the password of the `admin` user and have forgotten it, follow these steps to reset the password to the default value:

1. Connect to the switch using the console port.

2. Reload the switch:

```
Switch# reload
```

3. During the reload process, the following message will appear. Press **<Shift + B>** to enter Boot Management.

```
Switch# reload
reboot system? (y/n): y

Broadcast message from root@switch (Mon Feb 15 15:53:53 2016):

The system is going down for reboot NOW!
INIT: reboot: Restarting system


U-Boot 2014.01 (Oct 07 2015 - 10:47:24) - ONIE lenovo

CPU0:  P2020, Version: 2.1, (0x80e20021)
Core:  e500, Version: 5.1, (0x80211051)
Clock Configuration:
       CPU0:1200 MHz, CPU1:1200 MHz,
       CCB:600  MHz,
       DDR:400  MHz (800 MT/s data rate) (Asynchronous), LBC:37.500 MHz
L1:    D-cache 32 KiB enabled
       I-cache 32 KiB enabled
Board: G8xxx
I2C:   ready
DRAM:  4 GiB (DDR3, 64-bit, CL=6, ECC on)
Booting Lenovo NOS....
Press shift-B for startup menu or shift-R for recovery mode: ...
```

4. Select `Password reset` from the Boot Management menu by entering **P**:

```
Boot Management Menu
        M - Change startup mode (E-NOS vs C-NOS)
        I - Change booting image
        R - Startup in recovery mode (tftp and xmodem download of images
to recover switch)
        P - Password reset
        Q - Reboot
        E - Exit
Please choose your menu option: P
```

5. You are prompted for confirmation:

```
Confirm resetting system password: y/N? y
Password reset requested. Please reboot to complete the process
```

6. Reload the switch by selecting the 'Reboot' option. Press **<Q>**:

```
Boot Management Menu
        M - Change startup mode (E-NOS vs C-NOS)
        I - Change booting image
        R - Startup in recovery mode (tftp and xmodem download of images
to recover switch)
        P - Password reset
        Q - Reboot
        E - Exit
Please choose your menu option: q
```

7. After the reload is complete, you will be able to log into the switch using the default user admin with the default password admin.

8. You can change the default password by using the following command:

```
Switch(config)# username admin password <new password>
```

Chapter 6. AAA Protocols

Secure switch management is needed for environments that perform significant management functions across the Internet. The following are some of the functions for secured IPv4 management and device access:

- "RADIUS" on page 138
- "TACACS+" on page 141
- "Lightweight Directory Access Protocol" on page 144
- "Authentication, Authorization, and Accounting" on page 149
- "Public Key Infrastructure" on page 155

RADIUS

Cloud NOS supports the RADIUS (Remote Authentication Dial-in User Service) method to authenticate and authorize remote administrators for managing the switch. This method is based on a client/server model. The Remote Access Server (RAS)—the switch—is a client to the back-end database server. A remote user (the remote administrator) interacts only with the RAS, not the back-end server and database.

RADIUS authentication consists of the following components:

- A protocol with a frame format that utilizes UDP over IP (based on RFC 2865 and 2866)
- A centralized server that stores all the user authorization information
- A client: in this case, the switch

The switch—acting as the RADIUS client—communicates to the RADIUS server to authenticate and authorize a remote administrator using the protocol definitions specified in RFC 2865 and RFC 2866. Transactions between the client and the RADIUS server are authenticated using a shared key that is not sent over the network. In addition, the remote administrator passwords are sent encrypted between the RADIUS client (the switch) and the back-end RADIUS server.

RADIUS Basics

RADIUS offers the following advantages:

- RADIUS uses UDP-based speed-oriented transport
- RADIUS separates accounting, while authentication and authorization are treated together

How RADIUS Authentication Works

The RADIUS authentication process follows these steps:

1. A remote administrator connects to the switch and provides a user name and password.

2. Using Authentication/Authorization protocol, the switch sends request to authentication server.

3. The authentication server checks the request against the user ID database.

4. Using RADIUS protocol, the authentication server instructs the switch to grant or deny administrative access.

RADIUS Authentication Features in Cloud NOS

CNOS supports the following RADIUS authentication features:

- Supports RADIUS client on the switch, based on the protocol definitions in RFC 2865 and RFC 2866.

- For the G8272 and G8296 switches, CNOS supports user-configurable RADIUS application port. The default port is 1812 for authentication and 1813 for accounting.

- Supports MS-CHAP-V2 authentication method, based on the definitions in RFC 2759.

- Allows RADIUS secret password of 65 bytes.

- Supports user-configurable RADIUS server retry and time-out values:
  - Time-out value = 1-60 seconds
  - Retries = 0-5

- The switch will time out if it does not receive a response from the RADIUS server in 0-5 retries. The switch will also automatically retry connecting to the RADIUS server before it declares the server down.Allows network administrator to define privileges for one or more specific users to access the switch at the RADIUS user database.

Switch User Accounts

The user accounts in Table 5 can be defined in the RADIUS server dictionary file.

Table 5. *User Accounts and RADIUS*

User Account	Description and Tasks Performed	Password
Operator	The Operator manages all functions of the switch. The Operator can reset ports, except the management port.	`oper`
Administrator	The super-user Administrator has complete access to all commands, information, and configuration commands on the switch, including the ability to change both the user and administrator passwords.	`admin`

RADIUS Attributes for Cloud NOS User Privileges

When the user logs in, the switch authenticates his level of access by sending the RADIUS access request (the client authentication request) to the RADIUS authentication server.

All user privileges, other than those assigned to the Administrator, must be defined in the RADIUS dictionary. The file name of the dictionary is RADIUS vendordependent. The following RADIUS attributes are defined for user privilege levels:

User Name/Access	Vendor Code	Value
Operator	26543	`network-operator`
Admin	26543	`network-admin`

Configuring RADIUS on the Switch

Use the following procedure to configure Radius authentication on your switch.

1. Configure the IPv4/IPv6 addresses of the RADIUS servers, and enable RADIUS authentication and accounting.

```
Switch(config)# radius-server host 10.10.1.1
```

2. Configure the RADIUS secret.

```
Switch(config)# radius-server host 10.10.1.1 key [0|7] <1-65 character secret>
```

Notes:

- Setting up a specific key for an individual RADIUS server will overwrite the globally configured key.

- By default, the RADIUS key is not encrypted. To specify the encryption level of the key, use the following parameters:
  - **0** - the key is unencrypted (clear text)
  - **7** - the key is encrypted

3. If desired, you may change the default UDP port numbers used to listen to RADIUS.

The well-known ports for RADIUS are 1812 for authentication and 1813 for accounting.

```
Switch(config)# radius-server host 10.10.1.1 auth-port <UDP port number>
Switch(config)# radius-server host 10.10.1.1 acc-port <UDP port number>
```

4. Configure the number retry attempts for contacting the RADIUS server, and the timeout period.

```
Switch(config)# radius-server host 10.10.1.1 retransmit 3
Switch(config)# radius-server host 10.10.1.1 timeout 5
```

TACACS+

CNOS supports authentication and authorization with networks using the Terminal Access Controller Access-Control System Plus (TACACS+) protocol. The switch functions as the Network Access Server (NAS) by interacting with the remote client and initiating authentication and authorization sessions with the TACACS+ access server. The remote client is defined as someone requiring management access to the switch either through a data port or the management port.

TACACS+ Basics

TACACS+ offers the following advantages:

- TACACS+ uses TCP-based connection-oriented transport
- TACACS+ offers full packet encryption in authentication requests
- TACACS+ separates authentication, authorization, and accounting

How TACACS+ Authentication Works

TACACS+ authentication follows the steps below:

1. The remote administrator connects to the switch and provides username and password.
2. Using the authentication/authorization protocol, the switch sends a request to authentication server.
3. The authentication server checks the request against the user ID database.
4. Using the TACACS+ protocol, the authentication server instructs the switch to grant or deny administrative access to the remote user.

During a session, if additional authorization checking is needed, the switch checks with a TACACS+ server to determine if the user is granted permission to use a specific command.

Notes:

- A username can have a maximum length of 32 characters, consisting only of lowercase letters and numbers. The username must also start with a letter. Usernames that start with a number are invalid.
- A password can have a maximum length of 255 characters, consisting of lowercase or uppercase letters, numbers, and special characters.

TACACS+ Authentication Features in Cloud NOS

Authentication is the action of identifying a user and is generally done when the user first attempts to log into a device, or tries to gain access to its services. CNOS supports ASCII inbound login to the device.

Note: PAP, CHAP, and ARAP login methods, TACACS+ change password requests, and one-time password authentication are not supported.

Authorization

Authorization is the action of determining a user's privileges on the device and usually takes place after authentication.

The default mapping between TACACS+ authorization levels and CNOS management access levels is shown in Table 6. The authorization levels must be defined on the TACACS+ server.

Table 6. *Default TACACS+ Authorization Levels*

CNOS User Access Level	TACACS+ level
operator	1
admin	15 or 16

If the remote user is successfully authenticated by the authentication server, the switch verifies the privileges of the remote user and authorizes the appropriate access.

Accounting

Accounting is the action of recording a user's activities on the device for the purposes of billing and/or security. It follows the authentication and authorization actions. If the authentication and authorization is not performed via TACACS+, there are no TACACS+ accounting messages sent out.

You can use TACACS+ to record and track software login access, configuration changes, and interactive commands.

The switch supports the following TACACS+ accounting attributes:

- protocol (Telnet or SSH)
- start_time
- stop_time

Configuring TACACS+ Authentication on the Switch

You can configure up to 4 TACACS+ authentication servers. To set a TACACS+ server on the switch, follow these steps:

1. Enable TACACS+ on the switch:

```
Switch(config)# feature tacacs+
```

2. Configure the IP addresses of a TACACS+ server. Optionally, you can specify the TACACS+ server port.

```
Switch(config)# tacacs-server host <IP address> port <1-65535>
```

Note: If you want to configure a fifth TACACS+ server, you must first delete one of the existing four servers by using the following command:

```
Switch(config)# no tacacs-server host <IP address>
```

3. Configure the TACACS+ encryption/decryption key by using one of the following options:

   ● Set up a global key for all TACACS+ servers:

```
Switch(config)# tacacs-server key [0|7] <1-63 character key>
```

   ● Set up a key for each individual TACACS+ server:

```
Switch(config)# tacacs-server host <IP address> key [0|7] <1-63 character key>
```

   Notes:

   ● Setting up a specific key for an individual TACACS+ server will overwrite the globally configured key.

   ● By default, the TACACS+ key is not encrypted. To specify the encryption level of the key, use the following parameters:
      o **0** - the key is unencrypted (clear text)
      o **7** - the key is encrypted

4. Check the current TACACS+ settings and configured servers:

```
Switch> display tacacs-server
```

Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) is a protocol for accessing distributed directory information services over a network. Cloud NOS uses LDAP for authentication and authorization. With an LDAP client enabled, the switch will authenticate a user and determine the user's privilege level by checking with one or more directory servers instead of a local database of users. This prevents customers from having to configure local user accounts on multiple switches; they can maintain a centralized directory instead.

Configure an LDAP Profile

To configure LDAP on the switch:

1. Enable the LDAP feature:

```
Switch(config)# feature ldap
```

2. Configure LDAP profile information and enter LDAP Configuration mode.

```
Switch(config)# ldap-server profile <profile name>
```

where *profile name* is a string no longer than 16 characters.

3. Configure the distinguished name (DN):

```
Switch(config-ldap-profile)# base-dn "<distinguished name>"
```

where *distinguished name* is a string no longer than 128 characters.

4. Set the LDAP binding method:

```
Switch(config-ldap-profile)# bind-mode {prompted|predefined}
```

5. Set the retransmission count for LDAP connections:

```
Switch(config-ldap-profile)# retransmit <count>
```

where *count* is an integer from 1-5.

6. Set the LDAP connection local timeout period, in seconds:

```
Switch(config-ldap-profile)# time <time>
```

where *time* is an integer from 1-60.

7. Set the LDAP transmit mode and security options:

```
Switch(config-ldap-profile)# security {ldaps [igonre]|startTLE [ignore]|
clear}
```

where:

Parameter	Description
ldaps	Use secure LDAP.
startTLS	Use transport layer security (TLS).
clear	Use clear text.
ignore	(Optional) Do not have the LDAP client check the identity of the peer.

8. Set the LDAP server IP address:

```
Switch(config-ldap-profile)# host <IP address> [interface
{ethernet <host>/<port> | mgmt <management interface number>}
```

where *IP address* is the IPv4 address, the IPv6 address, the hostname, or the DNS name of the LDAP server.

Note: The **interface** argument is only used when connecting to an LDAP server IPv6 link local address.

9. (Optional) Set the TCP port to use for LDAP:

```
Switch(config-ldap-profile)# port <port>
```

where *port* is an integer from 1-65535. If no value is set, the switch will use default TCP port 389.

10. (Optional) Set the Public Key Infrastructure (PKI) to use for this LDAP profile:

```
Switch(config-ldap-profile)# pki <PKI name>
```

where *port* is an integer from 1-65535. If no value is configured, the switch will use the global PKI.

11. (Optional) Set a customized LDAP group attribute:

```
Switch(config-ldap-profile)# attribute group <attribute>
```

where *attribute* is a string up to 64 characters long. If no value is configured, the switch will use the default value memberOf.

12. (Optional) Set the local authorization method:

```
Switch(config-ldap-profile)# authorization {rbac|bitmap}
```

where

Parameter	Description
rbac	Use Role-Based Access Control (default).
bitmap	Use permission string bitmap.

Chapter 6: AAA Protocols **145**

13. (Optional) Set a customized LDAP permission name:

```
Switch(config-ldap-profile)# attribute permission-name <name>
```

where *name* is a string up to 32 characters long. If no value is configured, the switch will use the default value LenovoNetworkPermission.

14. Set a customized LDAP attribute permission bitmap or permission role:

```
Switch(config-ldap-profile)# attribute permission-value {bitmap|role}
{admin|oper|deny} <role>
```

where

Parameter	Description
bitmap	LDAP server attribute permission bitmap.
role	LDAP server attribute permission role.
admin	LDAP server attribute administrator permission.
deny	LDAP server attribute deny permission.
oper	LDAP server attribute operator permission.
role	The bitmap or permission role name; a string up to 16 characters long.

15. (Optional) Set a customized LDAP username:

```
Switch(config-ldap-profile)# attribute username <name>
```

where *name* is a string up to 32 characters long. If no value is configured, the switch will use the default value uid.

16. Set the DN for binding with the LDAP server:

```
Switch(config-ldap-profile)# predefined-credential dn <DN>
```

where *DN* is a string up to 255 characters long, for example:

```
Switch(config-ldap-profile)# predefined-credential dn
cn=Manager,dc=my-domain,dc=com
```

17. Set the password for the DN:

```
Switch(config-ldap-profile)# predefined-credential key <password>
```

where *password* is a string up to 64 characters long.

18. Set the group filter string for searching:

```
Switch(config-ldap-profile)# group-filter <words>
```

where *words* is a colon-separated list of group names used to determine user permissions; a string up to 256 characters long.

Note: The group-filter option only works for bitmap security mode.

19. Exit LDAP configuration mode:

```
Switch(config-ldap-profile)# exit
```

Create an LDAP Server Group

1. Create a group of LDAP servers in AAA:

```
Switch(config)# aaa group server ldap <group name>
```

where *group name* is the name of the group of LDAP servers. This command also enters you into LDAP Group Configuration mode.

Note: All group members must use the same protocols.

2. (Optional) Add a server to the current group:

```
Switch(config)# server <server profile name>
```

where *server profile name* is the name of the server profile being added.

3. Set the VRF for this LDAP server group:

```
Switch(config-ldap)# use-vrf {default|management}
```

where:

Parameter	Description
`default`	Use the default VRF (default).
`management`	Use the management VRF.

Only users who log in from an interface that is in this VRF will be allowed to login. Other users will get a timeout error when trying to authenticate with the servers.

Configure Global LDAP Settings

To configure global LDAP settings on the switch:

1. Set the global PKI profile to be used by LDAP:

```
Switch(config)# ldap-server pki <PKI name>
```

where *PKI name* is the name of the global LDAP PKI profile.

2. Specify the server group for console user login authentication:

```
Switch(config)# aaa authentication login console group <group list>
[local|none]
```

where:

Parameter	Description
group list	A list of up to eight groups.
local	(Optional) Use local authentication if the group servers fail.
none	(Optional) Do not require authentication if the group servers fail.

3. Specify the server group for remote user login authentication:

```
Switch(config)# aaa authentication login default group <group list>
[local|none]
```

where:

Parameter	Description
group list	A list of up to eight groups.
local	(Optional) Use local authentication if the group servers fail.
none	(Optional) Do not require authentication if the group servers fail.

4. Enable the display of error messages when authentication fails:

```
Switch(config)# aaa authentication login error-enable
```

View LDAP Settings

To display the running LDAP configuration, enter:

```
Switch(config)# display running-config ldap
```

To display the running AAA configuration, enter:

```
Switch(config)# display running-config aaa [all]
```

The optional **all** parameter displays the entire configuration, including switch defaults; leaving it off displays only the AAA LDAP configuration.

To display the configured groups, enter:

```
Switch(config)# display aaa groups
```

To display the running authentication settings, enter:

```
Switch(config)# display aaa authentication [login error-enable]
```

The optional **login error-enable** parameter displays the status of that authentication option.

Authentication, Authorization, and Accounting

Authentication, Authorization, and Accounting (AAA) allows the switch to use secure protocols to access the switch. The AAA mechanism uses RADIUSor TACACS+ as an underlying protocol. For AAA to work properly, you must configure the RADIUS or TACASCS+ protocol and a RADIUS or TACACS+ server. Once these are set up, all services can be configured through AAA. For more information about configuring RADIUS, see . For more information about configuring TACACS+, see .

AAA uses RADIUS and TACACS+ to provide authentication, authorization, and accounting. After setting up a RADIUS or TACACS+ server, you must configure the switch through AAA to use the respective server.

Authentication is the service that AAA provides to authenticate users that can log into the switch. Authentication is supported at login for either a remote protocol connection like SSH or Telnet (named default), or for the console.

Authorization is the service that AAA provides to give specific users specific rights. Some users may have administrator privileges, while other users may execute only basic level commands. All this can be configured on the server that is used by AAA, which has to be set up to query the server for this information.

Accounting is the service that AAA provides to log all user login and logout actions on the switch. This is useful for administrators that want to keep a clear log of anyone logging in or attempting to do so.

AAA Groups

You can put already defined RADIUS/TACACS+ servers together under a common group using AAA. You can configure all the servers inside a group using a single set of commands. AAA groups have specific parameters that are applied to the service using a certain group and to all the servers configured inside that group.

Note: When configuring an AAA group, each group must contain only servers that were previously defined using the same protocol.

Each server can be a member of multiple AAA groups. When multiple servers are configured in an AAA group, the switch will start querying the first server. If the query fails (there is no response from the first server), the next server is interrogated. Any answer (positive or negative) is considered authoritative for the query and it is final.

Group Lists

When configuring AAA services (authentication, authorization, and accounting), you can set up a list of AAA groups for each service. The *group list* can have up to a maximum of 8 AAA groups, each separated by space. The groups are queried one after the other in the order they were specified. If the query fails (no answer from the AAA group), the next group is interrogated.

Group lists can end with `local` or `none` or both. If all other groups fail, then, if `local` was specified, local authentication will be used, or, if `none` was used, no authentication will be required.

Note: A group list must contain at least one other group besides `local` or `none`.

Configuring AAA Groups

To create and configure an AAA group, follow these steps:

1. Create the AAA group:

```
Switch(config)# aaa group server tacacs+ <group name>
Switch(config-tacacs)#
```

Note: After creating the group, you will enter AAA Group configuration mode. To create a RADIUS group definition, use the following command:

```
Switch(config)# aaa group server radius <group name>
Switch(config-radius)#
```

2. Add the desired TACACS+ servers to the current group:

```
Switch(config-tacacs)# server <IP address>
```

Note: Only servers that were already defined using RADIUS/TACACS+ can be added to the group.

3. Optionally, you can set the Virtual Routing and Forwarding (VRF) instance for the current group:

```
Switch(config-tacacs)# use-vrf {default|management} <VRF instance>
```

4. Check the currently configured AAA groups:

```
Switch> display aaa groups
```

Authentication

Authentication can be configured separately for remote Telnet or SSH connections, or connections via the console port. You can choose to use either the pluggable authentication module (PAM) by specifying a group of servers (see "AAA Groups" on page 149) or local authentication (local database on the switch).

When using PAM, only the users defined on the authentication server can log in with the privilege levels and security configuration given by the server. The user database is kept remote and when the user tries to log in, AAA will query the defined server and provide access.

In addition to PAM, RADIUS offers MS-CHAP-V2 authentication method. However, when using this method, changing the password or domain related operations are not allowed.

Role-based authentication is supported. To avoid authorizing each command separately, the switch sends a privilege level request to the AAA server. Based on the obtained privileged level, the user is allowed or denied various CLI commands.

For Telnet or console connections the administrator does not need to create local users on the switch for them to log in. The user will be created on demand and will be kept indefinitely.

The administrator is not required to create a username on the switch for a user that logs in using Secure Shell (SSH) via RADIUS/TACACS+.

A maximum number of failed login attempts can be set. When this number is reached, the user is locked out of the switch. Only the system administrator cannot be locked out. Any other user can be unlocked by using the following command in Privileged EXEC mode:

```
Switch# remove aaa local user lockout username <username>
```

Configuring AAA Authentication

To configure authentication, follow these steps:

1. Configure authentication for console sessions:

   - Configure authentication to use an AAA group:

     ```
     Switch(config)# aaa authentication login console group <group list>
     ```

   Note: Use the following command to enable MS-CHAP-V2 authentication option:

     ```
     Switch(config)# aaa authentication login mschapv2 enable
     ```

   - Configure authentication to use the local database:

     ```
     Switch(config)# aaa authentication login console local
     ```

● Disable authentication:

```
Switch(config)# aaa authentication login console none
```

2. Configure authentication for Telnet or SSH sessions:

   ● Configure authentication to use an AAA group:

```
Switch(config)# aaa authentication login default group <group list>
```

   ● Configure authentication to use the local database:

```
Switch(config)# aaa authentication login default local
```

   ● Disable authentication:

```
Switch(config)# aaa authentication login default none
```

3. Configure the maximum number of failed login attempts:

```
Switch(config)# aaa local authentication attempts max-fail <1-25>
```

4. Optionally, you can configure the display of error messages for failed login attempts:

```
Switch(config)# aaa authentication login error-enable
```

5. Check the current AAA authentication settings:

```
Switch> display aaa authentication
```

   In case no TACACS+ servers are reachable, you can configure local authentication by using the following commands:

   ● for console sessions:

```
Switch(config)# aaa authentication login console group <group list> local
```

   ● for telnet or SSH sessions:

```
Switch(config)# aaa authentication login default group <group list> local
```

   Note: The switch will first try to contact the RADIUS/TACACS+ servers configured in the specified AAA group. If this fails, it will then turn to local authentication.

Authorization

Authorization enables control over the commands that are executed on the switch in Privileged EXEC and Global Configuration command modes. Every command is sent to the TACACS+ server and is executed only if the server authorizes it.

Note: AAA Authorization is available only for remote connections and not for connections using the console port.

Note: AAA Authorization is not supported on RADIUS.

Configuring AAA Authorization

To configure authorization, follow these steps:

1. Configure authorization for switch commands:

   ● Configure authorization to use an AAA group:

   ```
   Switch(config)# aaa authorization commands default group <group list>
   ```

   ● Configure authorization to use the local database:

   ```
   Switch(config)# aaa authorization commands default local
   ```

2. Check the current AAA authorization settings:

   ```
   Switch> display aaa authorization
   ```

Accounting

To enable accounting, select the group of servers where accounting information will be stored. Local accounting can also be enabled. For TACACS+ only login and logout information will be logged.

Configuring AAA Accounting

To configure accounting, follow these steps:

1. Configure accounting to use one of the following:

   - an AAA group:

   ```
   Switch(config)# aaa accounting {default|console} group <group list>
   ```

   - the local database:

   ```
   Switch(config)# aaa accounting {default|console} local
   ```

2. Check the current AAA accounting settings:

   ```
   Switch> display aaa accounting
   ```
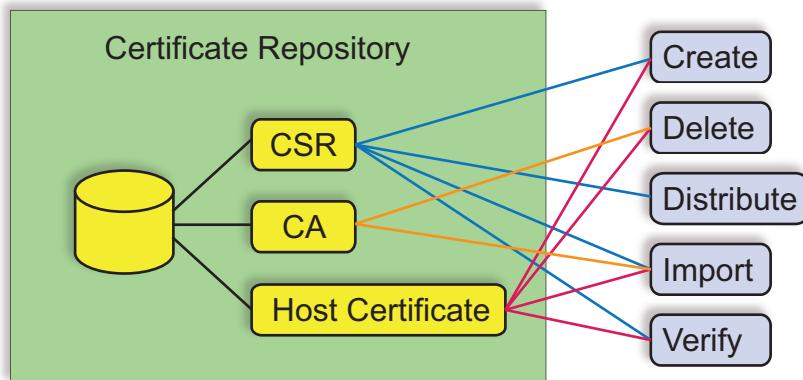
Public Key Infrastructure

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption. The purpose of a PKI is to facilitate the secure electronic transfer of information for network activities such as e-commerce, internet banking and confidential email. It is required for activities where simple passwords are an inadequate authentication method. It provides more rigorous proof to confirm the identity of the parties involved in communication and validates the information being transferred.

PKI Components

Figure 1 shows the components of a PKI.

Figure 1. The Architecture of a PKI

The PKI is comprised of the following components:

- Certificate-related operations:

  o Create
  You can create a public and private key pair, including associate subject name, serial number, and validity period with that public key.

  o Delete
  You can delete certificates.

  o Distribute
  You can export the certificate to an outside server. For example,you need to export the Certificate Signing Request (CSR) to an outside server to get signed.

  o Import
  You can import a certificate to a switch.

  o Verify
  You can verify an imported switch before it gets used by other modules.

- The certificate repository

  A database based on VRF containing the host certificates and CSR.

- Truststore

  A database for Certificate Authority (CA) certificates.

Implementing a PKI System

Follow these steps to set up a PKI profile.

1. Name the PKI:

```
Switch(config)# pki <name>
```

where *name* is the name of the PKI; an alphanumeric string up to 16 characters long. This puts you into PKI Configuration Mode.

Note: The maximum number of PKI profiles allowed is 10.

2. Using SFTP, import your CAs:

```
Switch(config-pki)# ca import <SFTP_URL> vrf {default|management}
```

where *SFTP_URL* is the location of the CA file. The typical format of this URL is:

```
sftp://username@address:directory/filename
```

3. Obtain a host certificate.

- To generate a host certificate on the switch, enter the following:

```
Switch(config-pki)# host-cert generate
Country Name (2 letter code) [US]:
State or Province Name (full name) [CA]:
Locality Name (eg, city) [Santa Clara]:
Organization Name (eg, company) [Lenovo Networking Operating System]:
Organizational Unit Name (eg, section) [Network Engineering]:
Common Name (eg, FQDN or YOUR name) []:
Email (eg, email address) []:
Validate time (days) [180]:
```

The values in square brackets ([]) are the default values.

- To import a certificate, enter the following commands:

```
Switch(config-pki)# host-cert private-key import <SFTP_URL> vrf
{default|management}
Switch(config-pki)# host-cert certificate import <SFTP_URL> [vrf
{default|management}]
```

where *SFTP_URL* is the location of the CA file. The typical format of this URL is:

```
sftp://username@address:directory/filename
```

4. Obtain a CSR.

- To generate a CSR on the switch, enter the following:

```
Switch(config-pki)# csr generate
Country Name (2 letter code) [US]:
State or Province Name (full name) [CA]:
Locality Name (eg, city) [Santa Clara]:
Organization Name (eg, company) [Lenovo Networking Operating System]:
Organizational Unit Name (eg, section) [Network Engineering]:
Common Name (eg, FQDN or YOUR name) []:
Email (eg, email address) []:
Validate time (days) [180]:
```

The values in square brackets ([]) are the default values.

- To import a signed CSR, enter the following command:

```
Switch(config-pki)# csr import <SFTP_URL> [vrf {default|management}]
```

where *SFTP_URL* is the location of the CA file. The typical format of this URL is:

```
sftp://username@address:directory/filename
```

- To export a CSR, enter the following command:

```
Switch(config-pki)# csr export <SFTP_URL> [vrf {default|management}]
```

where *SFTP_URL* is the location of the CA file. The typical format of this URL is:

```
sftp://username@address:directory/filename
```

5. (Optional) To enable a PKI for the REST server on the switch, enter:

```
Switch(config)# restApi pki <PKI_name> vrf {management|default}
```

where:

Argument	Definition
PKI_name	The name of the PKI; an alphanumeric string up to 16 characters long
management	Use the management VRF
default	Use the default VRF

Removing PKI Components

To delete a PKI, enter:

```
Switch(config)# no pki <PKI_name>
```

where *PKI_name* is the name of the PKI. If the PKI is in use, an error message will display, and the PKI will not be deleted.

To delete a CA, enter:

```
Switch(config)# pki <PKI_name>
Switch(config-pki)# ca delete <Subject>
```

where:

Argument	Definition
PKI_name	The name of the PKI; an alphanumeric string up to 16 characters long
Subject	The CA subject, in the form of: `C=US, ST=MD, L=Baltimore, CN=Test/emailAddress=test@example.com` where: ● C=*<country>* – for example, `US` ● ST=*<state or province name>* – for example, `CA` ● L=<locality name>, ● CN=*<common name>* ● emailAddress=*<email>* – for example, `test@example.com`

To remove a host certificate and the private key associated with it, enter:

```
Switch(config-pki)# host-cert delete
```

Viewing PKI Components

To view all PKIs, enter:

```
Switch# display pki
```

To view a host certificate associated with a specific PKI, enter:

```
Switch# display pki <PKI_name> host-certificate [base64]
```

where:

Argument	Definition
PKI_name	The name of the PKI; an alphanumeric string up to 16 characters long
base64	(Optional) Display the host certificate in base 64.

To view a CSR associated with a specific PKI, enter:

```
Switch# display pki <PKI_name> csr [base64]
```

where:

Argument	Definition
PKI_name	The name of the PKI; an alphanumeric string up to 16 characters long
base64	(Optional) Display the CSR in base 64.

To view a CA associated with a specific PKI, enter:

```
Switch# display pki <PKI_name> ca [base64|brief]
```

where:

Argument	Definition
PKI_name	The name of the PKI; an alphanumeric string up to 16 characters long
base64	(Optional) Display the CA in base 64.
brief	(Optional) Only display the CA issue, subject, and date information.

Chapter 7. Access Control Lists

An Access Control Lists (ACL) is a list of filters that permit or deny traffic for security purposes. They can also be used with Quality of Service (QoS) to classify and segment traffic to provide different levels of service to different traffic types. Each filter defines the conditions that must match for inclusion in the filter and the actions that are performed when a match is made.

This chapter discusses the following topics:

Supported ACL Types

Lenovo Cloud Network Operating System supports the following types of ACLs:

- Internet Protocol version 4 (IPv4) ACLs:

  IPv4 ACLs classify packets using Layer 3 and Layer 4 header fields of IPv4 packets. The following command creates an IPv4 ACL:

  ```
  Switch(config)# ip access-list <IP ACL name>
  ```

  Note: ACLs are not supported with IPv6.

- Media Access Control (MAC) ACLs:

  MAC ACLs classify packets using Layer 2 fields. The following command creates a MAC ACL:

  ```
  Switch(config)# mac access-list <MAC ACL name>
  ```

- Address Resolution Protocol (ARP) ACLs:

  An ARP ACL filters ARP request and reply packets using various ARP fields. The following command creates an ARP ACL:

  ```
  Switch(config)# arp access-list <ARP ACL name>
  ```

Based on where they are applied, ACLs can be classified as follows:

- Port-Based ACLs (PACLs)

  A PACL can be applied to all packets received on a switched port. The PACLs can be MAC or IPv4 PACLs.

- Router ACLs (RACLs)

  A RACL can be applied to all packets received on a routed port (ingress RACL) or all packets going out of a routed port (egress RACL) on a physical port or an SVI. The RACLs can be MAC or IPv4 RACLs.

- VLAN ACLs (VACLs)

  A VACL is an ACL that is applied to all ports belonging to a VLAN rather than a single port. VACLs can be MAC or IPv4 ACLs.

Summary of Packet Classifiers

ACLs allow you to classify packets according to a variety of content in the packet header, such as the source address, destination address, source port number, and destination port number. Once classified, packet flows can be identified for more processing.

IPv4 ACLs, MAC ACLs, and ARP ACLs, allow you to classify packets based on the following:

- IPv4 ACLs:
  - AH Packet
  - any protocol packet
  - EIGRP protocol packet
  - ESP packet
  - GRE packet
  - IANA assigned protocol number (0-255)
  - ICMP Packet
  - IGMP packet
  - IPv4 Encapsulation packet
  - KA9Q NOS compatible IP over IP tunneling
  - OSPF packet
  - IPCOMP packet
  - PIM packet
  - TCP Packet
  - UDP Packet
  - Source IPv4 address and subnet mask
  - Destination IPv4 address and subnet mask
  - Type of Service value
  - TCP/UDP application source host and mask
  - TCP/UDP source port as shown in Table 7
  - TCP/UDP application destination host and mask
  - TCP/UDP destination port as shown in Table 7
  - TCP flag value as shown in Table 8
- MAC ACLs:
  - Source MAC address
  - Destination MAC address
  - Ethernet type (ARP, IPv4, MPLS, RARP etc.)
  - Ethernet Priority (IEEE 802.1p Priority)
  - VLAN number and mask

- ARP ACLs:
  - Source IPv4 address and subnet mask
  - Destination IPv4 address and subnet mask
  - Source MAC address
  - Destination MAC address

Table 7. *TCP/UDP Applications*

Port	TCP/UDP Application	Port	TCP/UDP Application	Port	TCP/UDP Application
20	ftp-data	79	finger	179	bgp
21	ftp	80	http	194	irc
22	ssh	109	pop2	220	imap3
23	telnet	110	pop3	389	ldap
25	smtp	111	sunrpc	443	https
37	time	119	nntp	520	rip
42	name	123	ntp	554	rtsp
43	whois	143	imap	1645/1812	Radius
53	domain	144	news	1813	Radius
69	tftp	161	snmp	1985	Accounting
70	gopher	162	snmptrap		hsrp

Table 8. *TCP Flag Values*

Flag	Value
URG	0x0020
ACK	0x0010
PSH	0x0008
RST	0x0004
SYN	0x0002
FIN	0x0001

Summary of ACL Actions

An ACL must have a set of rules to classify the packet. The first token of each rule indicates the action to be performed on those packets that match the rule.

The following actions are supported:

- All ACL types support Permit and Drop actions

- VACLs also support Redirect actions

Note: All ACLs have an implicit default deny action to drop those packets that do not match any of the ACL rules. This default action applies to all IPv4 packets in an IPv4 ACL, and to any packet for MAC and ARP ACLs on the respective port or VLAN.

Configuring Port ACLs (PACLs)

A PACL is configured if an ACL is assigned to a Layer 2 switched interface or port channel. The following are sample PACL configurations:

```
Switch(config)# interface ethernet 1/5
Switch(config-if)# bridge-port
Switch(config-if)# ip port access-group <IP ACL name> in
Switch(config-if)# exit
```

```
Switch(config)# interface ethernet 1/10
Switch(config-if)# bridge-port
Switch(config-if)# mac port access-group <MAC ACL name>
Switch(config-if)# exit
```

Configuring Router ACLs (RACLs)

A RACL is configured if an ACL is assigned to a Layer 3 routed interface, port channel, or SVI. The following is a sample PACL configuration applied on a physical routed port:

```
Switch(config)# interface ethernet 1/5
Switch(config-if)# no bridge-port
Switch(config-if)# ip access-group <IP ACL name> {in|out}
Switch(config-if)# exit
```

While a PACL, a RACL, or both can be configured to an interface, only one of them will be active on that interface, depending upon whether that interface is a Layer 2 or a Layer 3 interface. However, two ACLs of the same type cannot be configured on the same interface.

A packet can match different ACL types at the same time. When that happens, actions are taken according to the order of precedence.

To apply an IPv4 ACL to an SVI, enter:

```
Switch(config)# interface vlan <VLAN ID (1-4093)>
Switch(config-if)# ip access-group <IPv4 ACL> {in|out}
[apply-routed-packets-only]
```

where:

Parameter	Description
no	(Optional) Remove the specified ACL.
IPv4 ACL	The name of the ACL to be applied to the specified VLAN
in	Apply this ACL to ingress IPv4 packets.
out	Apply this ACL to egress IPv4 packets.
apply-routed-packets-only	(Optional) Only match those IPv4 packets that are routedin the specified VLAN by the IPv4 ACLs already applied on L3 VLAN interfaces.

To remove an IPv4 ACL from an SVI, enter:

```
Switch(config)# interface vlan <VLAN ID (1-4093)>
Switch(config-if)# no ip access-group <IPv4 ACL> {in|out}
[routed-packets-only]
```

where:

Parameter	Description
no	(Optional) Remove the specified ACL.
IPv4 ACL	The name of the ACL to be applied to the specified VLAN

Parameter	Description
in	Remove this ACL from ingress IPv4 packets.
out	Remove this ACL from egress IPv4 packets.
routed-packets-only	(Optional) Only remove matching for those IPv4 packets that are routed in the specified VLAN by the IPv4 ACLs already applied on L3 VLAN interfaces.

Configuring VLAN ACLs (VACLs)

A VACL is an ACL that can be assigned to a VLAN rather than to a switch port as with IPv4 ACLs. This is particularly useful in a virtualized environment where traffic filtering and metering policies must follow virtual machines (VMs) as they migrate between hypervisors.

VACLs can be configured for all user-created VLANs except for reserved VLANs.

Individual VACL filters are configured in a similar fashion to IPv4 ACLs.

To create or change a VACL, follow these steps:

1. From Configuration Mode, enter VLAN Access Map Configuration Mode:

```
Switch(config)# vlan access-map <ACL name>
Switch(config-access-map)#
```

2. Enter an action for the VACL to take.

   To drop or forward matching packets, use the following command:

```
Switch(config-access-map)# action {drop|forward}
```

   To redirect matching packets, use the following command:

```
Switch(config-access-map)# action redirect {ethernet <chassis number/port
number>|port-aggregation <LAG number>}
```

   Note: Because the redirected packet will be subjected to the usual VLAN membership check, make sure the interface to where the packet is redirected has the same VLAN membership as the VACL.

3. Enter a match for the VACL to use.

   ● To configure a rule to match an IP ACL, use the following command:

```
Switch(config-access-map)# match ip address <IP ACL name>
```

   ● To configure a rule to match a MAC ACL, use the following command:

```
Switch(config-access-map)# match mac address <MAC ACL name>
```

4. Once a VACL filter is created, it can be assigned to or removed from a list of VLANs.

   ● To assign a VACL filter to a VLAN, use the following command:

```
Switch(config)# vlan filter <VACL name> vlan-list <VLAN ID (1-4093)>
```

   ● To remove a VACL filter from a VLAN, use the following command:

```
Switch(config)# no vlan filter <VACL name> vlan-list <VLAN ID (1-4093)>
```

In this example, traffic from VLAN 3 ports is forwarded.

```
Switch(config)# vlan access-map myVACL
Switch(config-access-map)# action forward
Switch(config-access-map)# exit

Switch(config)# vlan filter myVACL vlan-list 3
```

ACL Order of Precedence

Some types of ACLs cannot be attached multiple times on the same bind point, such as attaching two different PACLs on the same interface or two different VACLs on the same VLAN. If this happens, the second ACL will be not effective and remains useless. Therefore, such configurations are blocked.

When a packet is received on a port, the ACLs are applied in the following order:

1. PACLs and RACLs are applied on physical ports and port channels.

   ACLs are attached on physical ports as Port ACL and Routed Port ACLs as shown:

   - Port ACL:

     ```
     Switch(config)# interface ethernet 1/1
     Switch(config-if)# bridge-port
     Switch(config-if)# ip port access-group testAcl in
     ```

   - Routed Port ACL:

     ```
     Switch(config)# interface ethernet 1/2
     Switch(config-if)# no bridge-port
     Switch(config-if)# ip access-group testAcl in
     ```

2. Ingress VACL:

   ```
   Switch(config)# vlan filter testVlanAcl vlan-list 10-100
   ```

3. Ingress RACLs applied on Logical L3 interface (L3 VLAN interface):

   ```
   Switch(config)# interface vlan 3
   Switch(config-if)# no bridge-port
   Switch(config-if)# ip access-group testAcl in
   ```

4. ARP ACL

5. Egress router ACL

   Although the packets are processed in the order mentioned here, the ACLs are further grouped and are prioritized within each group as follows:

   - **Group 1**
     - (Highest priority) ACLs attached to physical port (port ACL and routed port ACL)
     - (Middle priority) VLAN ACLs (VLAN access-map)
     - (Lowest priority) ACLs attached to the VLAN interface.

   - **Group 2**
     - (Highest Priority) MAC ACLs
     - (Lowest priority) ARP ACLs

   When a packet matches ACLs in both groups, the ACL that has drop or redirect actions would be effective. If none of the matched ACLs have deny or redirect actions, all the actions in those ACLs will be applied to the packet.

Creating and Modifying ACLs

Each ACL is defined by a group of rules. You can create the following types of ACLs:

- **IPv4 ACLs**

  Each command defines one filter rule for matching traffic criteria. Each filter rule must also include an action (permit or deny the packet). For example:

  ```
  Switch(config)# ip access-list myACL
  Switch(config-acl)# permit udp any lt 301 10.2.2.1 0.0.0.255
  ```

  This example configures ACL 'myACL' to permit UDP traffic from any host, that has a UDP source port smaller than 301, and that goes to hosts that belong to the network 10.2.2.0/24. All other traffic will be dropped by the implicit deny rule.

- **MAC ACLs**

  Each command defines one filter rule for matching traffic criteria. Each filter rule must also include an action (permit or deny the packet). For example:

  ```
  Switch(config)# mac access-list myMAC-ACL
  Switch(config-mac-acl)# permit 001b.6384.45e6 0000.0000.ffff
  8c89.a502.93af 0000.0000.ffff cos 2
  ```

  This example configures MAC ACL 'myMAC-ACL' to permit traffic from MAC host 001B.6384.45E6 with source wildcard 0000.0000.FFFF that goes to MAC host 8C89.A502.93AF with destination wildcard 0000.0000.FFFF using Class of Service (CoS) 2. All other traffic will be dropped by the implicit deny rule.

- **ARP ACLs**

  Each command defines one filter rule for matching traffic criteria. Each filter rule must also include an action (permit or deny the packet). For example:

  ```
  Switch(config)# arp access-list myARP-ACL
  Switch(config-arp-acl)# permit ip any mac 0007.e94d.4a85 0000.0000.ffff
  ```

  This example configures ARP ACL 'myARP-ACL' to permit traffic from any IP source address with source MAC address 0007.E94D.4A85 and source wildcard 0000.0000.FFFF. All other traffic will be dropped by the implicit deny rule.

Creating an IPv4 ACL

To create an IPv4 ACL, follow these steps:

1. From Configuration Mode, enter ACL Configuration Mode for the ACL:

```
Switch(config)# ip access-list <ACL name>
Switch(config-acl)#
```

Note: The ACL name is case-sensitive.

2. Assign rules to the ACL using the following syntax:

```
Switch(config-acl)# [<sequence number>] {deny|permit} <protocol> <source IP address>
<source wildcard IP mask> [<operand> <port number>] <destination IP address> <destination wildcard
IP mask> [<operand> <port number>] [<bit to match on>]
```

Note: You can enter more than one rule for an ACL.

3. Verify the IPv4 ACL configuration:

```
Switch(config)# display ip access-lists <ACL name>
```

Note: To change a filtering rule, you must first remove it using the **no** command in front of the command that created the group:

```
Switch(config-acl)# no [<sequence number>] {deny|permit} <protocol> <source IP
address> <source wildcard IP mask> [<operand> <port number>] <destination IP address> <destination
wildcard IP mask> [<operand> <port number>] [<bit to match on>]
```

You can also use the following command to remove a filtering rule:

```
Switch(config-acl)# no <sequence number>
```

Removing an IPv4 ACL

To remove an ACL, enter:

```
Switch(config)# no ip access-list <ACL name>
```

Resequencing an IPv4 ACL

Sometimes you may want to fit a rule between two existing numbers. Use the **resequence** command in Configuration Mode to change the numbering scheme but keep the existing rules in order:

```
Switch(config)# resequence ip access-list <ACL name> <starting number>
<increment>
```

For example:

```
Switch(config)# resequence ip access-list myACL 10 5
```

The rules configured for ACL 'myACL' will be reassigned sequence numbers starting with number 10. Following rules will have a sequence number equal to the previous rule's sequence number plus an increment of 5. In other words, the first rule in the ACL will have a sequence number of 10, the second rule 15, the third rule 20, and so forth.

Creating a MAC ACL

To create a MAC ACL, follow these steps:

1. From Configuration Mode, enter MAC ACL Configuration Mode for the ACL:

```
Switch(config)# mac access-list <ACL name>
Switch(config-mac-acl)#
```

Note: The ACL name is case-sensitive.

2. Assign rules to the ACL using the following syntax:

```
Switch(config-mac-acl)# [<sequence number>] {deny|permit} {{<source MAC address>
[<source wildcard MAC mask>]}|any|host <source MAC address>} {{<destination MAC address>
[<destination wildcard MAC mask>]}|any|host <destination MAC address>} {<protocol>|cos
<cos>|vlan <VLAN ID (1-4093)>} [cos <cos>] [vlan <VLAN ID (1-4093)>]
```

Note: You can enter more than one rule for an ACL.

3. Verify the MAC ACL configuration:

```
Switch(config)# display mac access-lists <ACL name>
```

Note: To change a filtering rule, you must first remove it using the **no** command in front of the command that created the group:

```
Switch(config-mac-acl)# no [<sequence number>] {deny|permit} {{<source MAC
address> [<source wildcard MAC mask>]}|any|host <source MAC address>} {{<destination MAC
address> [<destination wildcard MAC mask>]}|any|host <destination MAC address>} {<protocol>|
cos <cos>|vlan <VLAN ID (1-4093)>} [cos <cos>] [vlan <VLAN ID (1-4093)>]
```

You can also use the following command to remove a filtering rule:

```
Switch(config-acl)# no <sequence number>
```

Removing a MAC ACL

To remove a MAC ACL, use the following command:

```
Switch(config)# no mac access-list <ACL name>
```

Resequencing a MAC ACL

If you want to fit a rule between two existing numbers, use the **resequence** command to change the numbering scheme but keep the existing rules in order:

```
Switch(config)# resequence mac access-list <ACL name> <starting number> <increment>
```

For example:

```
Switch(config)# resequence mac access-list myACL 10 5
```

The rules configured for ACL myACL will be reassigned sequence numbers starting with 10. Following rules will have a sequence number equal to the previous rule's sequence number plus an increment of 5; the first rule will have a sequence number of 10, the second rule 15, the third rule 20, and so forth.

Creating an ARP ACL

To create an ARP ACL, follow these steps:

1. From Configuration Mode, enter ARP ACL Configuration Mode for the ACL:

```
Switch(config)# arp access-list <ACL name>
Switch(config-arp-acl)#
```

Note: The ACL name is case-sensitive.

2. Assign rules to the ACL using the following syntax:

```
Switch(config-arp-acl)# [<sequence number>] {deny|permit} [request|response]
ip {<source IPv4 address>|any|host <single source address> {mac {<source MAC address>
<source wildcard MAC mask>|any|host <source MAC address>}}}
```

Note: You can enter more than one rule for an access list.

3. Verify the ARP ACL configuration:

```
Switch(config)# display arp access-lists <ACL name>
```

Note: To change a filtering rule, you must first remove it using the no command in front of the command that created the group:

```
Switch(config-arp-acl)# no [<sequence number>] {deny|permit} [request|
|response] ip {<source IPv4 address>|any|host <single source address> {mac {<source MAC
address> <source wildcard MAC mask>|any|host <source MAC address>}}}
```

You can also use the following command to remove a filtering rule:

```
Switch(config-acl)# no <sequence number>
```

Removing an ARP ACL

To remove an ARP ACL, use the following command:

```
Switch(config)# no arp access-list <ACL name>
```

Resequencing an ARP ACL

Sometimes you may want to fit a rule between two existing numbers. Use the **resequence** command in Configuration Mode to change the numbering scheme but keep the existing rules in order:

```
Switch(config)# resequence arp access-list <ACL name> <starting number> <increment>
```

For example:

```
Switch(config)# resequence arp access-list myACL 10 5
```

The rules configured for ACL myACL will be reassigned sequence numbers starting with number 10. Following rules will have a sequence number equal to the previous rule's sequence number plus an increment of 5. In other words, the first rule in the ACL will have a sequence number of 10, the second rule 15, the third rule 20, and so forth.

Remarks and ACLs

A *remark* is comment text added to an ACL to make the ACL easier to understand. The following rules apply to ACL remarks:

- Remarks are supported for IPv4, MAC, and ARP access lists.

- All remarks are single line.

- The maximum length of a remark must not exceed 100 characters. Leading and trailing spaces are not counted in the 100 character limit.

- Remarks can only contain characters permitted by the CLI. No special characters, such as "?" or newline, are allowed in remarks.

- You can add as many remarks to a single ACL as the sequence number of the ACL permits. The maximum number of ACL filter entries is 2048.

- You can add a remark at any particular sequence location, as long as it does not conflict with an existing entry at that particular location.

- ACL remarks are displayed in the running configuration.

- ACL remarks are persistent across reboots.

Add ACL Remarks

To add remarks to an IPv4 ACL, in the appropriate ACL Configuration mode (IPv4, MAC, or ARP), enter

```
Switch(config-acl)# [<sequence number>] remark <remark>
```

```
Switch(config-mac-acl)# [<sequence number>] remark <remark>
```

```
Switch(config-arp-acl)# [<sequence number>] remark <remark>
```

where:

Parameter	Description
sequence number	(Optional) The sequence number at which you want the remark inserted; an integer from 1-2147483645. If omitted, the remark is automatically appended with a sequence number ten higher than the last sequence number.
remark	The comment text; a string up to 100 characters long.

Remove ACL Remarks

To remove remarks from an ACL, in the appropriate ACL Configuration mode (IPv4, MAC, or ARP), enter:

```
Switch(config-acl)# no [<sequence number>] remark <remark>
```

```
Switch(config-mac-acl)# no [<sequence number>] remark <remark>
```

```
Switch(config-arp-acl)# no [<sequence number>] remark <remark>
```

where:

Parameter	Description
sequence number	(Optional) The sequence number at which you want the remark inserted. If omitted, the remark is automatically appended with a sequence number ten higher than the last sequence number.
remark	The *exact* text of the remark to be removed.

Note: If multiple remarks contain the same text, only the first occurrence of the remark will be removed. To remove a specific duplicate remark, remove it by sequence number, as in:

```
Switch(config-mac-acl)# no [<sequence number>]
```

View ACL Remarks

To view the remarks of a specific ACL, enter:

```
Switch# display access-lists <access list>
```

where *access-list* is the name of the ACL.

Viewing ACL Rule Statistics

ACL Rule statistics display how many packets have matched each ACL rule. Use ACL Rule statistics to check filter performance or to debug the ACL rule filter configuration.

You must enable statistics for each ACL that you wish to monitor:

```
Switch(config)# {arp|ip|mac} access-list myacl
Switch(config-acl)# statistics per-entry
```

ACL Configuration Examples

The following are examples of configuring ACLs.

ACL Example 1

Use this configuration to block traffic to a specific host. In the following example, all traffic that ingresses on ethernet port 1/1 is denied, if it is destined for IP address 100.10.1.1.

1. Create an ACL:

```
Switch(config)# ip access-list myACL1
Switch(config-acl)#
```

2. Configure the ACL:

```
Switch(config-acl)# deny any any host 100.10.1.1
Switch(config-acl)# permit any any any
Switch(config-acl)# exit
```

3. Apply ACL 'myACL1' to ethernet interface 1/1:

```
Switch(config)# interface ethernet 1/1
Switch(config-if)# ip port access-group myACL1 in
Switch(config-if)# exit
```

ACL Example 2

Use this configuration to block traffic from a network destined for a specific host address. In the following example, all traffic that ingresses on ethernet port 1/2 with source IP from network 100.10.2.0/24 and destination IP 200.20.2.2 is denied.

1. Create an ACL:

```
Switch(config)# ip access-list myACL2
Switch(config-acl)#
```

2. Configure the ACL:

```
Switch(config-acl)# deny any 100.10.2.0/24 host 200.20.2.2
Switch(config-acl)# permit any any any
Switch(config-acl)# exit
```

3. Apply ACL 'myACL2' to ethernet interface 1/2:

```
Switch(config)# interface ethernet 1/2
Switch(config-if)# ip port access-group myACL2 in
Switch(config-if)# exit
```

ACL Example 3

Use the following configuration to deny all IGMP packets that ingress on a port.

1. Create an ACL:

```
Switch(config)# ip access-list myACL3
Switch(config-acl)#
```

2. Configure the ACL:

```
Switch(config-acl)# deny igmp any any
Switch(config-acl)# permit any any any
Switch(config-acl)# exit
```

3. Apply ACL 'myACL3' to ethernet interface 1/3:

```
Switch(config)# interface ethernet 1/3
Switch(config-if)# ip port access-group myACL3 in
Switch(config-if)# exit
```

ACL Example 4

Use the following configuration to permit access from any host to hosts with destination MAC address 11:05:00:10:00:00 and MAC wildcard FF:F5:FF:FF:FF:FF, and deny access to all other hosts.

1. Create a MAC ACL:

```
Switch(config)# mac access-list myMAC-ACL4
Switch(config-mac-acl)#
```

2. Configure the MAC ACL:

```
Switch(config-mac-acl)# permit any 11:05:00:10:00:00 FF:F5:FF:FF:FF:FF
Switch(config-mac-acl)# deny any any
Switch(config-mac-acl)# exit
```

3. Apply MAC ACL 'myMAC-ACL4' to ethernet interface 1/4:

```
Switch(config)# interface ethernet 1/4
Switch(config-if)# mac port access-group myMAC-ACL4
Switch(config-if)# exit
```

ACL Example 5

This configuration blocks egress traffic from a specific network to any destination. In this example, all traffic from the network 100.10.2.0/24 to any destination egressing from routed port 3 is denied.

1. Create an ACL:

```
Switch(config)# ip access-list myACL5
Switch(config-acl)#
```

2. Configure the ACL:

```
Switch(config-acl)# deny any 100.10.2.0/24 any
Switch(config-acl)# permit any any any
Switch(config-acl)# exit
```

3. Apply ACL myACL5 to ethernet interface 1/5:

```
Switch(config)# interface ethernet 1/5
Switch(config-if)# ip access-group myACL5 out
Switch(config-if)# exit
```

ACL Example 6

This configuration creates an ARP ACL with multiple matching rules:

1. Create an ARP ACL:

```
Switch(config)# arp access-list myARP-ACL6
Switch(config-arp-acl)#
```

2. Add rules to the ACL:

```
Switch(config-arp-acl)# statistics per entry
Switch(config-arp-acl)# 10  deny ip any mac host 0033.3333.3333
Switch(config-arp-acl)# 20  permit ip any mac 0044.4444.4444
0000.0000.FFFF
Switch(config-arp-acl)# 30  permit ip host 20.20.20.2 mac host
0011.1111.1111
Switch(config-arp-acl)# 50  permit ip host 40.40.40.2 mac 0055.5555.5555
0000.0000.FFFF
Switch(config-arp-acl)# 60  deny ip host 50.50.50.2 mac any
Switch(config-arp-acl)# 70  deny ip 30.30.30.2 0.0.0.255 mac
0022.2222.2222 0000.0000.FFFF
Switch(config-arp-acl)# 80  deny ip 40.40.40.2 0.0.0.255 mac any
Switch(config-arp-acl)# 90  permit ip 50.50.50.2 0.0.0.255 mac host
0055.5555.5555
Switch(config-arp-acl)# exit
```

3. Configure the inspection filter:

```
Switch(config)# ip arp inspection filter myARP-ACL vlan 20,30,40,50
```

ACL Logging

ACL logging provides insight into traffic as it passes the network or is dropped by the network switch. ACL logging can be CPU intensive and can slow traffic going through the switch. There are two main factors that contribute to the CPU load increase from ACL logging:

- process switching of packets that match log-enabled access control entries (ACEs)
- generating and transmitting log messages.

The logging option applies to an individual ACE and causes packets that match the ACE to be logged. Logging includes the packet's source and destination IP addresses.

Using the configuration commands detailed in this section, you can strike a balance between traffic visibility and the corresponding impact on the switch CPU load.

Configure ACL Logging

To set up ACL logging on your switch:

1. Enter ACL configuration mode:

```
Switch(config)# ip access-list <ACL name>
```

2. Enable ACL logging for each ACE:

```
Switch(config-acl)# <sequence number> deny any any any log(log all rejected traffic)
Switch(config-acl)# <sequence number> deny icmp any any log(log rejected ICMP
                                                            traffic)
Switch(config-acl)# <sequence number> deny igmp any any dscp af11 fragments
log                                          (log specific rejected
                                             IGMP packets)
Switch(config-acl)# exit
```

Note: The log keyword appears at the end of every possible ACE configuration. If enabled, ACL logs will be generated when the rules of the ACL match traffic.

3. Set the maximum number of log entries cached in the software:

```
Switch(config)# logging ip access-list cache entries <entries>
```

where *entries* is the maximum number of entries; an integer from 1-1048576. The default value is 8000.

Note: Each entry corresponds to a specific traffic flow that is matched by an ACL rule that has the log option enabled.

4. Set the log update interval, in seconds:

```
Switch(config)# logging ip access-list cache interval <seconds>
```

where *seconds* is the cache interval; an integer from 5-3600. The default value is 300.

Note: The interval configured allows only one packet per interval to be processed no matter how many log-enabled ACEs exist.

5. Set how often log messages are generated and sent after the initial packet match:

```
Switch(config)# logging ip access-list cache threshold <cache interval>
```

where *cache* is the cache interval. The default value is 0.

Note: These commands use a threshold described as a number of packets, not as a time interval. This configured threshold is applied per flow and does not disable the initial match log message or the interval periodic update.

6. Set the ACL log level accepted by the syslog:

```
Switch(config)# logging level acllog <level>
```

where *level* is the ACL log level; an integer from 0-7 that signifies the following:

Level	Meaning
0	emergencies
1	alerts
2	critical
3	errors
4	warnings
5	notifications
6	information
7	debug

The default value is 6 (information).

7. Clear the ACL cache:

```
Switch(config)# remove logging ip access-list cache
```

8. Display the IP access list cache:

```
Switch(config)# display logging ip access-list cache
```

9. Display the status of the IP access list cache:

```
Switch(config)# display logging ip access-list status
```

Part 3: Switch Basics

This section discusses basic switching functions:

- "Interface Management" on page 187
- "Forwarding Database" on page 213
- "VLANs" on page 217
- "Ports and Link Aggregation" on page 243
- "Spanning Tree Protocol" on page 277
- "Virtual Link Aggregation Groups" on page 297
- "Quality of Service" on page 331
- "CEE" on page 353
- "Secure Mode" on page 373

Chapter 8. Interface Management

The term *interface* can refer to a physical switch port or a logical port, a VLAN, or a Link Aggregation Group (LAG).

This section discusses the following topics:

- "Interface Management Overview" on page 188
- "Management Interface" on page 189
- "Physical Ports" on page 191
- "Port Aggregation" on page 200
- "Loopback Interfaces" on page 201
- "Switch Virtual Interfaces" on page 202
- "Basic Interface Configuration" on page 203

Interface Management Overview

Interface Management (IFM) is part of the system infrastructure that acts as a logical interface abstraction for software components (applications) running on the switch. It is provided by the Network Service Module (NSM). NSM follows a server-client model and, as a server, it handles interface interactions with other switch applications, which are clients to the NSM server.

NSM clients include applications like: Neighbor Discovery Protocol (NDP), Routing Information Base (RIB), Spanning Tree Protocol (STP), Link Aggregation Control Protocol (LACP), Internet Group Management Protocol (IGMP), Border Gateway Protocol (BGP), and more. NSM clients subscribe to the services provided by the NSM server to deal with interface events, like a link going up or down. When an interface event occurs, NSM handles the event accordingly and then notifies all of its clients.

All interfaces are assigned a unique internal identifier called an interface index (ifindex). The *ifindex* is a 32-bit integer attributed to the interface by NSM. Once an ifindex has been assigned to an interface, it is forever bound to it.

During CLI interactions with the switch, interfaces are identified by their names, and not by their ifindex.

To view the ifindex of all interfaces, use the following command:

```
Switch> display interface snmp-ifindex
-------------------------------------------------------------------------
Port                     IFMIB Ifindex (hex)
-------------------------------------------------------------------------
Ethernet1/1              410001  (0x00064191)
Ethernet1/2              410002  (0x00064192)
Ethernet1/3              410003  (0x00064193)
Ethernet1/4              410004  (0x00064194)
...
Ethernet1/54/4           410072  (0x000641d8)
loopback0                     8  (0x00000008)
mgmt0                         3  (0x00000003)
po100                    100100  (0x00018704)
po1000                   101000  (0x00018a88)
po2000                   102000  (0x00018e70)
Vlan1                         9  (0x00000009)
Vlan100                      13  (0x0000000d)
Vlan200                      14  (0x0000000e)
Vlan300                      15  (0x0000000f)
```

Note: An interface cannot be configured until it is created. Also, if the interface is removed, its configuration is lost. For example, the configuration of a 40 Gigabit Ethernet (GbE) port will be lost once the port is split into four 10 GbE breakout ports. There are some interfaces created by default on the switch that cannot be removed.

The switch supports both IPv4 and IPv6 addressing on Layer 3 interfaces, such as routed ports, loopback interfaces or Switch Virtual Interfaces. For more details, see Chapter 17, "Basic IP Routing" and Chapter 20, "Internet Protocol Version 6".

Management Interface

The Management Interface is a special physical port on the switch that allows you to perform switch management tasks.

It is a Layer 3 interface and it cannot be configured as a Layer 2 port.

The management interface cannot forward traffic. To keep it separate from other interfaces, the management port is part of the management Virtual Routing and Forwarding (VRF) instance, while other interfaces are part of the default VRF instance.

To configure the management interface, use the following command:

```
Switch(config)# interface mgmt 0
Switch(config-if)#
```

Notes:

- After running the above command, you will enter Interface Configuration mode for the management port.

- By default, the management interface is created by the switch and it cannot be removed.

To view detailed information about the management interface, use the following command:

```
Switch(config)# display interface mgmt 0

Interface mgmt0
  Hardware is Management Ethernet  Current HW addr: a897.dcde.2500
  Physical:a897.dcde.2500  Logical:(not set)
  index 3 metric 1 MTU 1500 Bandwidth 1000000 Kbit
  no bridge-port
  arp ageing timeout 1500
  <UP,BROADCAST,RUNNING,ALLMULTI,MULTICAST>
  VRF Binding: Associated with management
  Speed auto Duplex full
  IPV6 DHCP IA-NA client is enabled.
  inet 10.241.41.27/25 broadcast 10.241.41.127
  inet6 fe80::aa97:dcff:fede:2500/64
  RX
    131249 input packets 5 unicast packets 128053 multicast packets
    3191 broadcast packets 23903151 bytes
  TX
    6749 output packets 0 unicast packets 6749 multicast packets
    0 broadcast packets 939061 bytes
```

Virtual Routing and Forwarding

Virtual Routing and Forwarding (VRF) allows multiple instances of a routing table to work simultaneously inside a router.

These are the advantages of using VRF technology:

- Network paths can be segmented without using multiple devices.

- The routing instances are independent; the same or overlapping IP addresses can be used without conflicting with each other.

- VRF acts like a logical router (LSR), but while a LSR may include many routing tables, a VRF instance uses only a single routing table.

- VRF uses routing tables known as forwarding information bases (FIBs). Each VRF context created is associated with a FIB ID.

By default, the switch has 2 pre-defined VRFs:

- *default* vrf (`FIBID 0`)

- *management* vrf (`FIBID 1`). The management VRF contains only the management interface, being a reserved vrf for management purposes only.

Physical Ports

The following Lenovo switches are 1U rack-mountable aggregation devices:

- RackSwitch G8272
- RackSwitch G8332
- ThinkSystem NE1032T RackSwitch
- ThinkSystem NE1032 RackSwitch
- ThinkSystem NE1072T RackSwitch
- ThinkSystem NE10032 RackSwitch
- ThinkSystem NE2572 RackSwitch

The Lenovo RackSwitch G8296 is a 2U rack-mountable aggregation device.

All of these switches use a wire-speed, non-blocking switching fabric that provides simultaneous wire-speed transport of multiple packets at low latency on all ports.

G8272 Physical Port Capabilities

The G8272 has the following port capabilities:

- Forty-eight 10 Gigabit Ethernet (GbE) Small Form Pluggable Plus (SFP+) ports which also support legacy 1 GbE connections
- Six 40 GbE Quad Small Form Pluggable Plus (QSFP+) ports, each of which can optionally be used as four 10 GbE ports

SPF+ ports, when used in 10 GbE mode, can be populated with optical transceiver modules, or active or passive Direct-Attach Cables (DACs). When used in legacy 1 GbE mode, the ports can be populated with optical or copper transceiver modules.

QSFP+ ports can be populated with optical QSFP+ modules or passive DACs, including those that allow the port to be used as four 10 GbE ports.

By default, the G8272 has the following interface (port) configuration:

- ethernet interfaces 1-48 are configured as 10 GBit/s ports
- ethernet interfaces 49-54 are configured as 40 GBit/s ports

Table 9 shows the available port modes and supported speeds for the G8272.

Table 9. *Available Port Modes and Supported Speeds for the G8272*

Port Mode	Supported Speeds
1x100G	100G, 100G-Auto Negotiation
2x50G	50G, 50G-Auto Negotiation
4x25G	25G, 25G-Auto Negotiation
1x40G	40G
4x10G	10G

G8296 Physical Port Capabilities

The G8296 has the following port capabilities:

- Eighty-six 10 Gigabit Ethernet (GbE) Small Form Pluggable Plus (SFP+) ports which also support legacy 1 GbE connections
- Ten 40 GbE Quad Small Form Pluggable Plus (QSFP+) ports, of which two ports can optionally be used as four 10 GbE ports

SPF+ ports, when used in 10 GbE mode, can be populated with optical transceiver modules, or active or passive Direct-Attach Cables (DACs). When used in legacy 1 GbE mode, the ports can be populated with optical or copper transceiver modules.

QSFP+ ports can be populated with optical QSFP+ modules or passive DACs, including those that allow the port to be used as four 10 GbE ports.

By default, the G8296 has the following interface (port) configuration:

- ethernet interfaces 1-86 are configured as 10 GBit/s ports
- ethernet interfaces 87-96 are configured as 40 GBit/s ports

Table 10 shows the available port modes and supported speeds for the G8296

Table 10. *Available Port Modes and Supported Speeds for the G8296*

Port Mode	Supported Speeds
1x100G	100G, 100G-Auto Negotiation
2x50G	50G, 50G-Auto Negotiation
4x25G	25G, 25G-Auto Negotiation
1x40G	40G
4x10G	10G
1x25G	25G, 25G-Auto Negotiation
1x10G	10G

G8332 Physical Port Capabilities

The G8332 has the following port capabilities:

- Thirty-two 40 GbE Quad Small Form Pluggable Plus (QSFP+) ports, of which ports 2-25 can optionally be used as four 10 GbE ports

QSFP+ ports can be populated with optical QSFP+ modules or passive DACs, including those that allow the port to be used as four 10 GbE ports.

By default, the G8332 has the following interface (port) configuration:

- ethernet interfaces 1-32 are configured as 40 GBit/s ports

Table 11 shows the available port modes and supported speed for the G8332:

Table 11. *Available Port Modes and Supported Speeds for the G8332*

Port Mode	Supported Speed
1x40G	40G
4x10G	10G
1x10G	10G

NE1072T Physical Port Capabilities

The NE1072T has the following port capabilities:

- Forty-eight 1000/10G BASE-T RJ45 ports.

- Six 40 GbE Quad Small Form Pluggable Plus (QSFP+) ports, all of which can optionally be used as four 10 GbE ports

The 10G BASE-T RJ45 ports, when used in 10 GbE mode, must use CAT6 copper cabling. When used in 1000 base T mode, the ports can be populated with CAT5E copper cabling.

QSFP+ ports can be populated with optical QSFP+ modules or passive DACs, including those that allow the port to be used as four 10 GbE ports.

By default, the NE1072T has the following interface (port) configuration:

- ethernet interfaces 1-48 are configured as 10 GBit/s ports

- ethernet interfaces 49-54 are configured as 40 GBit/s ports

Table 12 shows the available port modes and supported speeds for the NE1072T:

Table 12. *Available Port Modes and Supported Speeds for the NE1072T*

Port Mode	Supported Speeds
1x40G	40G
4x10G	10G
1x10G	10G
1x10G-Base-T	1G-Auto Negotiation, 10G-Auto Negotiation, 1G+10G-Auto Negotiation

NE1032T Physical Port Capabilities

The NE1032T has the following port capabilities:

- Twenty-four 1000/10G BASE-T RJ45 ports

- Eight Gigabit Ethernet (GbE) Small Form Pluggable Plus (SFP+) ports which also support legacy 1 GbE connections

The 10G BASE-T RJ45 ports, when used in 10 GbE mode, must use CAT6 copper cabling. When used in 1000 base T mode, the ports can be populated with CAT5E copper cabling.

SPF+ ports, when used in 10 GbE mode, can be populated with optical transceiver modules, or active or passive Direct-Attach Cables (DACs). When used in legacy 1 GbE mode, the ports can be populated with optical or copper transceiver modules.

By default, the NE1032T has the following interface (port) configuration:

● Ethernet interfaces 1-32 are configured as 10 GBit/s ports

Table 13 shows the available port modes and supported speeds for the NE1032T:

Table 13. *Available Port Modes and Supported Speeds for the NE1032T*

Port Mode	Supported Speeds
1x40G	40G
4x10G	10G
1x10G	10G
1x10G-Base-T	1G-Auto Negotiation, 10G-Auto Negotiation, 1G+10G-Auto Negotiation

NE1032 Physical Port Capabilities

The NE1032 has the following port capabilities:

● Thirty-two 10 Gigabit Ethernet (GbE) Small Form Pluggable Plus (SFP+) ports which also support legacy 1 GbE connections

SFP+ ports, when used in 10 GbE mode, can be populated with optical transceiver modules, or active or passive Direct-Attach Cables (DACs). When used in legacy 1 GbE mode, the ports can be populated with optical or copper transceiver modules.

By default, the NE1032 has the following interface (port) configuration:

● ethernet interfaces 1-32 are configured as 10 GBit/s ports

Table 14 shows the available port modes and supported speeds for the NE1032:

Table 14. *Available Port Modes and Supported Speeds for the NE1032*

Port Mode	Supported Speeds
1x40G	40G
4x10G	10G
1x10G	10G
1x10G-Base-T	1G-Auto Negotiation, 10G-Auto Negotiation, 1G+10G-Auto Negotiation

NE2572 Physical Capabilities

The NE2572 has the following port capabilities:

● Forty-eight 25 Gigabit Ethernet (GbE) SFP28 ports supporting 10GbE and 25GbE connections.

● Six additional 100 Gigabit Ethernet (GbE) QSFP28 ports supporting 10GbE, 25GbE, 40 GbE, 50GbE and 100GbE connections.

QSFP28 ports can be populated with Optical QSFP28/QSFP+ modules or Quad Direct Attach Cables (QDACs), including those that allow breakout to four 25GbE SFP28/SFP+ ports.

SFP28 port can be populated with Optical or Copper SFP28/SFP+ transceivers or SFP28/SFP+ Direct Attach Cables (DACs).

The following per-port hardware profiles are available:

● One 40 GbE port

● Two 50 GbE ports

● Four 10 GbE ports

● Four 25 GbE ports

● One 100GbE port

The following ports speed combinations are supported:

● 10 GbE / 40 GbE / 100 GbE

● 10 GbE / 25 GbE / 50 GbE / 100 GbE

The NE2572 does not support 25 GbE or 50 GbE ports with 40 GbE ports at the same time. Table 17 shows which combinations of port speeds can coexist.

Table 15. *Port Speed Combinations Supported on the ThinkSystem NE2572 RackSwitch*

Combination	10G	25G	40G	50G	100G
10G	Yes	Yes	Yes	Yes	Yes
25G	Yes	Yes	**NO**	Yes	Yes
40G	Yes	**NO**	Yes	**NO**	Yes
50G	Yes	Yes	**NO**	Yes	Yes
100G	Yes	yes	Yes	Yes	Yes

Table 16 shows the available port modes and supported speeds for the NE2572:

Table 16. *Available Port Modes and Supported Speeds for the NE2572*

Port Mode	Supported Speeds
1x100G	100G, 100G-Auto Negotiation
2x50G	50G, 50G-Auto Negotiation
4x25G	25G, 25G-Auto Negotiation

Table 16. *Available Port Modes and Supported Speeds for the NE2572 (continued)*

Port Mode	Supported Speeds
1x40G	40G
4x10G	10G
1x25G	25G, 25G-Auto Negotiation
1x10G	10G

NE10032 Physical Capabilities

The NE10032 has the following port capabilities:

- Thirty-two 100 Gigabit Ethernet (GbE) QSFP28 ports supporting 10GbE, 25GbE, 50GbE and 100GbE connections
- One hundred and twenty-eight 10 GbE QSFP28 ports
- One hundred and twenty-eight 25 GbE QSFP28 ports
- Thirty-two 40 GbE QSFP28 ports
- Sixty-four 50 GbE QSFP28 ports
- custom profile
- default profile

QSFP28 ports can be populated with Optical QSFP28/QSFP+ modules or Quad Direct Attach Cables (QDACs), including those that allow breakout to four 25GbE SFP28/SFP+ ports.

By default, the NE10032 has the following interface (port) configuration

- ethernet interfaces 1-32 are configured as 100 GBit/s ports

The following per-port hardware profiles are available:

- One 40 GbE port
- Two 50 GbE ports
- Four 10 GbE ports
- Four 25 GbE ports
- One 100 GbE port

The following ports speed combinations are supported:

- 10 GbE / 40 GbE / 100 GbE
- 10 GbE / 25 GbE / 50 GbE / 100 GbE

The NE10032 does not support 25 GbE or 50 GbE ports with 40 GbE ports at the same time. Table 17 shows which combinations of port speeds can coexist.

Table 17. *Port Speed Combinations Supported on the ThinkSystem NE10032 RackSwitch*

Combination	10G	25G	40G	50G	100G
10G	Yes	Yes	Yes	Yes	Yes
25G	Yes	Yes	**NO**	Yes	Yes
40G	Yes	**NO**	Yes	**NO**	Yes
50G	Yes	Yes	**NO**	Yes	Yes
100G	Yes	yes	Yes	Yes	Yes

Table 18 shows the available port modes and supported speeds for the NE10032:

Table 18. *Available Port Modes and Supported Speeds for the NE10032*

Port Mode	Supported Speeds
1x100G	100G, 100G-Auto Negotiation
2x50G	50G, 50G-Auto Negotiation
4x25G	25G, 25G-Auto Negotiation
1x40G	40G
4x10G	10G

CLI Port Format

During CLI interactions, physical ports are referred to as ethernet interfaces and have the following port numbering scheme:

- 40, 10, or 100 GbE ports: *<chassis number>/<port number>*

    For example, physical port 10:

    ```
    interface ethernet 1/10
    ```

- 10, 25, or 50 GbE breakout ports: *<chassis number>/<port number>/<subport number>*

    For example, breakout port 3 of physical port 52:

    ```
    interface ethernet 1/52/3
    ```

Note: If a port in 25 GbE mode has the name `ethernet 1/25/1`, the same name is associated in the 50 GbE mode for the first 50 GbE port from the QSFP cage, such as port lanes 1-2. If a port in 25 GbE mode has the name `ethernet 1/25/2`, the same name is associated in the 50 GbE mode for the second 50 GbE port from the QSFP cage, such as port lanes 3-4.

In CLI commands, ethernet interfaces can be used in the following ways:

- as a single interface:

```
Switch(config)# interface ethernet 1/10
```

- as a continuos range:

```
Switch(config)# interface ethernet 1/10-24
```

- as a discrete range:

```
Switch(config)# interface ethernet 1/10, ethernet 1/20-22
```

To configure an ethernet interface, use the following command:

```
Switch(config)# interface ethernet <chassis number>/<port number>
Switch(config-if)#
```

For example:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)#
```

Notes:

- After running the above command, you will enter Interface Configuration mode for the specified ethernet port.

- Ethernet interfaces cannot be removed using the CLI.

To view detailed information about an ethernet interface, use the following command:

```
Switch(config)# display interface ethernet <chassis number>/<port number>

Interface Ethernet1/1
  Hardware is Ethernet  Current HW addr: a897.dcde.2503
  Physical:a897.dcde.2503  Logical:(not set)
  index 410010 metric 1 MTU 1500 Bandwidth 10000000 Kbit
  Port Mode is trunk
  <UP,BROADCAST,MULTICAST>
  VRF Binding: Not bound
  Speed 10000 Mb/s Duplex full
  Last link flapped never
  Last clearing of "display interface" counters never
  30 seconds input rate 0 bits/sec, 0 bytes/sec, 0 packets/sec
  30 seconds output rate 0 bits/sec, 0 bytes/sec, 0 packets/sec
  Load-Interval #2: 5 minute (300 seconds)
     input rate 0 bps, 0 pps; output rate 0 bps, 0 pps
  RX
    0 unicast packets  0 multicast packets  0 broadcast packets
    0 input packets  0 bytes
    0 jumbo packets  0 storm suppression packets
    0 giants  0 input error  0 short frame  0 overrun  0 underrun
    0 watchdog  0 if down drop
    0 input with dribble  0 input discard(includes ACL drops)
    0 Rx pause
  TX
    0 unicast packets  0 multicast packets  0 broadcast packets
    0 output packets  0 bytes
    0 jumbo packets
    0 output errors  0 collision  0 deferred  0 late collision
    0 lost carrier  0 no carrier  0 babble
    0 Tx pause
  0 interface resets
```

Port Aggregation

Multiple physical ports can be aggregated to act as a single logical port called a Link Aggregation Group (LAG). Port aggregation provides link redundancy, increased bandwidth, and traffic balancing across the ethernet ports.

By default, no physical ports are aggregated. Thus, no LAGs are configured on the switch.

To create or modify an existing LAG, use the following command:

```
Switch(config)# interface port-aggregation <1-4096>
Switch(config-if)#
```

Note: After running the above command, you will enter Interface Configuration mode for the specified LAG.

To delete a LAG, use the following command:

```
Switch(config)# no interface port-aggregation <1-4096>
```

To view detailed information about a LAG, use the following command:

```
Switch(config)# display interface port-aggregation <1-4096>

Interface po1
  Hardware is AGGREGATE  Current HW addr: a897.dcde.0203
  Physical:(not set)  Logical:(not set)
  index 100001 metric 1 MTU 1500 Bandwidth 0 Kbit
  Port Mode is access
  <UP,BROADCAST,MULTICAST>
  VRF Binding: Not bound
  Members in this port-aggregation:
    Ethernet1/1
  Last clearing of "display interface" counters never
  30 seconds input rate 0 bits/sec, 0 bytes/sec, 0 packets/sec
  30 seconds output rate 0 bits/sec, 0 bytes/sec, 0 packets/sec
  Load-Interval #2: 5 minute (300 seconds)
     input rate 0 bps, 0 pps; output rate 0 bps, 0 pps
  RX
    0 unicast packets  0 multicast packets  0 broadcast packets
    0 input packets  0 bytes
    0 jumbo packets  0 storm suppression packets
    0 giants  0 input error  0 short frame  0 overrun  0 underrun
    0 watchdog  0 if down drop
    0 input with dribble  0 input discard(includes ACL drops)
    0 Rx pause
  TX
    0 unicast packets  0 multicast packets  0 broadcast packets
    0 output packets  0 bytes
    0 jumbo packets
    0 output errors  0 collision  0 deferred  0 late collision
    0 lost carrier  0 no carrier  0 babble
    0 Tx pause
  0 interface resets
```

For more details about port aggregation, see Chapter 11, "Ports and Link Aggregation".

Loopback Interfaces

A loopback interface is a virtual Layer 3 interface. It is usually used by different applications to communicate when they run on the same switch. Loopback interfaces are used exclusively on the switch and they do not pass packets to other interfaces. Any traffic that is sent to a loopback interface is immediately passed back to the switch as if it was received from another device. If a packet is routed to a loopback interface, but it is not destined for that interface, it will be discarded.

It is used to allow Border Gateway Protocol (BGP) sessions to be always up, even when the outgoing switch interface is not available. BGP sessions can use a loopback interface as their termination address.

Open Shortest Path First (OSPF) and other Layer 3 protocols can use the IP address of a loopback interface as their router ID.

To create or configure a loopback interface, use the following command:

```
Switch(config)# interface loopback <0-7>
Switch(config-if)#
```

Note: After running the above command, you will enter Interface Configuration mode for the specified loopback interface.

To delete a loopback interface, use the following command:

```
Switch(config)# no interface loopback <0-7>
```

Notes:

- By default, loopback interface 0 is created on the switch and it cannot be deleted.

- Loopback interfaces cannot be configured as Layer 2 interfaces. They are automatically created as Layer 3 interfaces and this setting is unchangeable.

To view detailed information about a loopback interface, use the following command:

```
Switch(config)# display interface loopback <0-7>

Interface loopback0
  Hardware is Loopback
  index 8 metric 1 MTU 1500 Bandwidth 0 Kbit
  no bridge-port
  arp ageing timeout 1500
  <UP,LOOPBACK,RUNNING>
  VRF Binding: Not bound
  DHCP client is disabled.
  Encapsulation LOOPBACK
    0 packets input 0 bytes
    0 multicast frames 0 compressed
    0 input errors 0 frame 0 overrun 0 fifo
    0 packets output 0 bytes 0 underruns
    0 output errors 0 collisions 0 fifo
```

Switch Virtual Interfaces

A Switch Virtual Interface (SVI) is a VLAN that is used as a virtual Layer 3 interface. When a VLAN is created, its associated SVI is also created. An SVI is deleted when its related VLAN is deleted.

By default, VLAN 1 is created on the switch, and thus, VLAN 1 SVI is also created.

An SVI behaves as an ordinary Layer 3 interface:

● IP addresses can be configured

● Layer 2 commands are not allowed

● It supports all features that can run on Layer 3 interfaces

A VLAN can be associated with only one SVI. For more information about VLANs, see Chapter 10, "VLANs".

Notes:

● You can configure up to a maximum of 256 SVIs.

● Before configuring an SVI interface, you must first create the appropriate Layer 2 VLAN. For more details, see "Creating a VLAN" on page 220.

To configure an SVI, use the following command:

```
Switch(config)# interface vlan <1-4094>
Switch(config-if)#
```

Note: After running this command, you will enter Interface Configuration mode for the specified SVI.

To remove the configuration of a SVI interface, you must delete its associated VLAN. For more details, see "Deleting a VLAN" on page 221.

Note: The SVI associated with VLAN 1 cannot be deleted because VLAN 1 also cannot be removed from the switch.

To view detailed information about an SVI, enter:

```
Switch(config)# display interface vlan <1-4094>

Interface Vlan1
  Hardware is VLAN  Current HW addr: a897.dcde.2501
  Physical:(not set)  Logical:(not set)
  index 9 metric 1 MTU 1500 Bandwidth 0 Kbit
  no bridge-port
  arp ageing timeout 1500
  <UP,BROADCAST,RUNNING,MULTICAST>
  VRF Binding: Not bound
  DHCP client is disabled.
  Last clearing of "display interface" counters never
```

Basic Interface Configuration

This section covers some of the basic options available when configuring an interface, like its port speed, MAC address, or description.

To configure an interface, you must enter its Configuration mode.

To view a brief overview of all interfaces, enter:

```
Switch(config)# display interface brief

---------------------------------------------------------------------------
Ethernet     VLAN  Type Mode   Status  Reason                   Speed Port
Interface                                                              Agg#
---------------------------------------------------------------------------
Ethernet1/1  1     eth  access down    Link not connected       auto  1
Ethernet1/2  1     eth  access down    Link not connected       auto  --
Ethernet1/3  19    eth  access up      none                     10000 --
Ethernet1/4  20    eth  access down    Administratively down    auto  --
Ethernet1/5  --    eth  routed down    Administratively down    auto  --
Ethernet1/6  1     eth  access down    Link not connected       auto  --
Ethernet1/7  1     eth  access down    Link not connected       auto  --
Ethernet1/8  1     eth  access down    Link not connected       auto  --
Ethernet1/9  1     eth  access down    Link not connected       auto  --
Ethernet1/10 --    eth  routed down    Administratively down    auto  --
...
```

To view the basic interface capabilities, enter:

```
Switch(config)# display interface capabilities

Ethernet1/1
  Model:                G8xxx-X10G-SUP
  Type (SFP capable):   SFP+ 10G BASE
  MDIX:                 no
  Speed:                1000/10000
  Duplex:               full
  Trunk encap. type:    802.1Q
  Port-aggregation:     yes
  Flowcontrol:          rx-(off/on),tx-(off/on)
  QOS scheduling:       rx-(8q1t),tx-(8q1t)
  CoS rewrite:          no
  ToS rewrite:          no
  SPAN:                 yes
  Port mode:            Routed,Bridged
...
```

By default, the switch ethernet interfaces are configured as Layer 2 ports. To configure an ethernet interface as a Layer 3 port (routed port), see Chapter 18, "Routed Ports".

To view information about discarded incoming packets, enter:

```
Switch(config)# display interface [<interface>] ingress-discard-details
Ethernet1/1
+---------------------------------------+----------------------------+
|        Counter Description            |          Count             |
|                                       |                            |
+---------------------------------------+----------------------------+
  IPv4 Discards                                         0
  IPv6 Discards                                         0
  STP Discards                                          0
  Policy Discards                                       100
  ACL Drops                                             0
  Receive Drops                                         0
  Vlan Discards                                         33
  IBP/CBP Discards                                      0
  OBM LP                                                0
  OBM HP                                                0
+---------------------------------------+----------------------------+
...
```

where:

Parameter	Description
interface	(Optional) A type of interface followed by the interface name or number (such as ethernet 1/10).
IPv4 Discards	IPv4 packets not sent to the CPU.
IPv6 Discards	IPv6 packets not sent to the CPU.
STP Discards	Packets received when the interface is not in STP forwarding mode.
Policy Discards	Packets discarded because of an input policy on the interface, such as storm control.
ACL Drops	ACL packets dropped.
Receive Drops	Dropped packets where no output port could be determined for the packet, such as: ● VLAN check failed ● MTU check failed ● ACL Drops action is hit ● Traffic was destined to a non-existing route ● Traffic was destined to an IP address for which the ARP entry is incomplete ● Multicast traffic dropped because of a null output interface
VLAN Discards	VLAN-based discards, such as VLAN tagged packets incoming on a port that is not a member of the VLAN.
IBP/CBP Discards	Packets discarded due to either Ingress Back Pressure (reaching ingress buffer threshold for the interface), or because the Cell Buffer Pool is full (no buffer is available).

Parameter	Description
OBM LP	Oversubscription Buffer Management low priority dropped packets.
OBM HP	Oversubscription Buffer Management high priority dropped packets.

No argument gives you the ingress discard details for all interfaces.

To view information about discarded outgoing packets, enter:

```
Switch(config)# display interface [<interface>] egress-discard-details
Ethernet1/1
+---------------------------------------+---------------------------+
|          Counter Description          |           Count           |
|                                       |                           |
+---------------------------------------+---------------------------+
  HOL-blocking Discards                                 0
  MMU Discards                                          0
  Cell Error Discards                                   0
  MMU Aging Discards                                    0
  Other Discards                                        34
+---------------------------------------+---------------------------+
...
```

where:

Parameter	Description
interface	(Optional) A type of interface followed by the interface name or number (such as ethernet 1/10).
HOL-blocking Discards	Packets discarded due to a head of line blocking condition.
MMU Discards	Packets discarded by the memory management unit.
Cell Error Discards	Packet cell purge errors.
MMU Aging Discards	Packets discarded because of an excessive transit delay through the MMU.
Other Discards	Packets dropped because of any other condition. This includes packets dropped by STP not being in a forwarding state.

No argument gives you the egress discard details for all interfaces.

To display all internal hardware counters, enter:

```
Switch(config)# display interface [<interface>] counters-all
```

where *interface* (optional) is a type of interface followed by the interface name or number (such as ethernet 1/10). No argument gives you all internal hardware counters for all interfaces.

To display all interface statistics, enter:

```
Switch(config)# display interface [<interface>] statistics
```

where *interface* (optional) is a type of interface followed by the interface name or number (such as ethernet 1/10). No argument gives you statistics for all interfaces.

Forwarding Error Correction

The NE10032 and the NE2572 switches support forwarding error correction (FEC). This option is disabled by default. To enable FEC, enter:

```
Switch# fec {auto|cl74|cl91|off}
```

where:

Parameter	Description
auto	(Default) Enables and configures FEC automatically based on the port speed of the interface:
cl74	Enables FEC with clause 74 for interfaces configured with 25 Gb/s or 40 Gb/s port speeds.
cl91	Enables FEC with clause 91 for interfaces configured with 50Gb/s port speeds.
off	Disables FEC on the interface.

Note: The command is applicable only for 25G, 40G, 50G and 100G ports.

- auto (default). The FEC is chosen according with speed. 100G: cl91, 25G and 50G: cl74, 40G: off.

- cl91. Clause 91

- cl74. Clause 74

- off. FEC is disabled

To reset FEC to its default value, enter:

```
Switch# no fec
```

Interface Description

To add a short description to an interface, use the following command:

```
Switch(config-if)# description <text string>
```

Note: Interface descriptions can have up to a maximum of 80 characters.

To view the description of an interface, use the following command:

```
Switch(config)# display interface description
```

To reset the interface description to its default value, use the following command:

```
Switch(config-if)# no description
```

Interface Duplex

A duplex system is a point-to-point system consisting of two network devices that can communicate between them in both directions (transmission and reception).

There are three configurable duplex modes:

● full duplex - both devices can communicate with each other simultaneously;

● half duplex - both device can communicate with each other, but not simultaneously;

● auto-negotiation - the duplex mode is automatically decided by the device.

To configure the duplex mode of an interface, enter:

```
Switch(config-if)# duplex {auto|full|half}
```

To view the current duplex for an interface, use the following command:

```
Switch(config)# display interface status

--- ----------------------------------------------------------------------
Port            Name            Status  Vlan    Duplex  Speed   Type
----------------------------------------------------------------------------
Ethernet1/1     --              up      1       full    10000   eth
Ethernet1/2     --              up      trunk   full    10000   eth
Ethernet1/3     --              up      1       full    10000   eth
Ethernet1/4     --              up      1       full    10000   eth
Ethernet1/5     --              up      1       full    10000   eth
Ethernet1/6     --              up      1       full    10000   eth
...
```

By default, the duplex mode is auto-negotiation.

To reset the duplex mode configuration to its default, use the following command:

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

Interface MAC Address

To view the current MAC address of an interface, use the following command:

```
Switch(config)# display interface mac-address

-----------------------------------------------------------------------
Interface              Mac-Address     Burn-in Mac-Address
-----------------------------------------------------------------------
Ethernet1/1            a897.dcde.2503  a897.dcde.2503
Ethernet1/2            a897.dcde.2504  a897.dcde.2504
Ethernet1/3            a897.dcde.2505  a897.dcde.2505
Ethernet1/4            a897.dcde.2506  a897.dcde.2506
Ethernet1/5            a897.dcde.2507  a897.dcde.2507
...
Ethernet1/54           a897.dcde.2538  a897.dcde.2538
mgmt0                  a897.dcde.2500  a897.dcde.2500
po1                    0e00.0000.0001  (not set)
Vlan1                  a897.dcde.2501  (not set)
```

Note: This command is available for every type of switch interface.

Interface Maximum Transmission Unit

The Maximum Transmission Unit (MTU) is the size (in bytes) of the largest data unit that a protocol can send across the interface.

For each network layer, there is a different size limitation:

- G8332, G8296, and G8272:
    - Layer 2 packets: 64 - 9216 bytes
    - Layer 3 IPv4 packets: 576 - 9216 bytes
    - Layer 3 IPv6 packets: 1280 - 9216 bytes
- NE1032, NE1032T, NE1072T, NE10032, and NE2572:
    - Layer 2 packets: 64 - 9216 bytes
    - Layer 3 IPv4 packets: 576 - 9198 bytes
    - Layer 3 IPv6 packets: 1280 - 9198 bytes

To configure the MTU of an interface, use the following command:

```
Switch(config-if)# mtu <64-9216 (bytes)>
```

By default, the MTU is 1500 bytes.

To reset the MTU of an interface to its default value, use the following command:

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

Interface Shutdown

Each individual interface can be administratively brought up or down. To put an interface in the administratively up state, enter:

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

To put an interface in the administratively down state, enter:

```
Switch(config-if)# shutdown
```

To view the link state of an interface, enter:

```
Switch(config)# display interface status

--- --------------------------------------------------------------------
Port            Name            Status  Vlan    Duplex  Speed  Type
----------------------------------------------------------------------
Ethernet1/1     --              up      1       full    10000  eth
Ethernet1/2     --              up      trunk   full    10000  eth
Ethernet1/3     --              up      1       full    10000  eth
...
```

Interface Speed

For physical ethernet interfaces and the management interface, the port speed can be configured to different values than the default setting. Port speed refers to the maximum amount of data (in bits) that can pass across an interface at any given second.

The switch offers some of the following bits per second (BPS) rates:

- 100 BPS (10 Gb/s)
- 1000 BPS (10 Gb/s)
- 10000 BPS (10 Gb/s)
- 100000 BPS (100 Gb/s)
- 25000 BPS (25 Gb/s)
- 40000 BPS (40 Gb/s)
- 50000 BPS (50 Gb/s)
- auto-negotiation - the speed is automatically decided by the device (NE10032/NE2572 only)

Note: Some of the list BPS rates may not be available, depending upon your switch model.

To configure the port speed of a physical ethernet interface or the management interface, enter:

```
Switch(config-if)# speed {<bits/s>|auto}
```

Note: Only the parameters 10, 100, 1000, and auto are allowed for management ports.

By default, the port speed of an interface is auto-negotiation if the port is capable of auto-negotiation.

To reset the port speed to its default value, enter:

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

Flow Control

Flow Control manages the rate of data transmission between two devices, when the transmitter sends traffic faster then the receiver can process. It allows the receiver to control the transmission speed and prevents traffic loss.

Flow Control can be enabled for both egress and ingress traffic.

To enable or disable flow control on an interface, use the following command:

```
Switch(config-if)# flowcontrol {send|receive} {on|off}
```

To view the current flow control configuration, use the following command:

```
Switch(config)# display interface flowcontrol

Port         Send FlowControl  Receive FlowControl  RxPause  TxPause
             admin   oper      admin   oper
-----------  ------- --------  ------- -----------  -------  --------
Ethernet1/1  on      on        off     off                0         0
Ethernet1/2  off     off       off     off                0         0
Ethernet1/3  off     off       off     off                0         0
Ethernet1/4  off     off       off     off                0         0
Ethernet1/5  off     off       off     off                0         0
Ethernet1/6  off     off       off     off                0         0
Ethernet1/7  off     off       off     off                0         0
...
```

By default, flow control is disabled on all interfaces.

To reset flow control to its default setting, use the following command:

```
Switch(config-if)# no flowcontrol
```

Storm Control

Excessive transmission of traffic can result in a network storm. A network storm can overwhelm the local network with constant broadcast or multicast traffic, thus degrading network performance. Common symptoms of a network storm are denial-of-service (DoS) attacks, slow network response times, and network operations timing out.

Storm Control allows the monitoring of incoming traffic levels for a set time interval. During that interval, traffic level (a percentage of the total interface bandwidth) is compared to the configured traffic storm control level. If ingress traffic reaches the storm control level, the switch drops all packets until the time interval ends.

The switch can be configured to limit the available bandwidth for broadcast, multicast, and unknown unicast packets. Unicast packets whose destination MAC address is not in the Forwarding Database (FDB) are considered unknown unicast packets. When an unknown unicast packet is encountered, the switch handles it like a broadcast packet and floods it to all other ports in the VLAN (broadcast domain). A high rate of unknown unicast traffic can have the same negative effects as a broadcast storm.

To configure the bandwidth available on an interface for different types of packets, use the following command:

```
Switch(config-if)# storm-control {broadcast|multicast|unicast} level
<bandwidth percentage>
```

Note: The *bandwidth percentage* value can be defined up to two decimals.

By default, storm control is disabled on all switch interfaces.

To disable storm control on an interface, use the following command:

```
Switch(config-if)# no storm-control {broadcast|multicast|unicast} level
```

Chapter 9. Forwarding Database

The forwarding database (FDB) table is used by a Layer 2 device (switch/bridge) to store the MAC addresses that have been learned and their corresponding ports. The MAC addresses are either dynamically learned from the source addresses of incoming traffic, or manually configured by the system administrator.

When an Ethernet frame is received, the switch dynamically builds the FDB table by using the source MAC address of the received frames. If the FDB table does not contain any information on the destination MAC address of the received frames, it will flood the Ethernet frame out to all ports in the broadcast domain. When the recipient replies, the switch adds its relevant source MAC address and port ID to the FDB table.

The following topics are discussed in this chapter:

MAC Learning

MAC addresses can be learned only when both global and port based learning are enabled.

By default, MAC address learning is enabled.

Interface based MAC learning can only be enabled or disabled on Ethernet ports or Link Aggregation Groups (LAGs). All the Ethernet interfaces belonging to a LAG (static or dynamic) need to have the same MAC learning configuration.

To globally configure MAC learning, use the following commands:

```
Switch(config)# [no] mac-learn disable
```

To configure MAC learning on each switch interface, use the following command:

```
Switch(config-if)# [no] mac-learn disable
```

To display current MAC learning status, use the following command:

```
Switch# display mac address-table learning [interface ethernet <chassis
number/port number>]
```

To clear the FDB table of dynamic MAC entries, use the following command:

```
Switch# remove mac address-table dynamic [address <MAC address>|interface
{ethernet <chassis number/port number>|port-aggregation <LAG number>}] [vlan
<VLAN ID (1-4093)>]
```

Static MAC addresses

To avoid flooding all LAN ports, entries can be manually configured in the MAC address table. These static MAC entries are retained across switch reloads.

A static MAC address cannot be configured with a non-existing VLAN.

To configure a static MAC address, use the following command:

```
Switch(config)# mac address-table static <MAC address> vlan <VLAN ID (1-4093)>
interface {ethernet <chassis number/port number>|port-aggregation <LAG number>}
```

In addition, you can enter a multicast address as a statically configured MAC address. A multicast address can accept more than one interface as its destination.

To delete a static entry from the FDB table, use the following command:

```
Switch(config)# no mac address-table static <MAC address> vlan <VLAN ID (1-4093)>
[interface {ethernet <chassis number/port number>|port-aggregation <LAG number>}]
```

To clear the FDB table of static MAC entries, use the following command:

```
Switch# remove mac address-table static [address <MAC address>|interface
{ethernet <chassis number/port number>|port-aggregation <LAG number>}] [vlan
<VLAN ID (1-4093)>]
```

Note: Static MAC address will override any matching dynamic entry in the database.

To display static MAC address configuration, use the following command:

```
Switch# display mac address-table static [address <MAC address>] [interface
{ethernet <chassis number/port number>|port-aggregation <LAG number>}] [vlan
<VLAN ID (1-4093)>]
```

Aging Time

You can configure the amount of time that a MAC entry remains in the FDB table. This is called *aging time*.

To configure the aging time, in seconds, use the following command:

```
Switch(config)# mac address-table aging-time <0-1,000,000>
```

To configure MAC entries to be stored permanently in the FDB table, you can set the aging time to 0, by using the following command:

```
Switch(config)# mac address-table aging-time 0
```

The default value is 1800 seconds. To reset the aging time to its default value, use the following command:

```
Switch(config)# no mac address-table aging-time
```

To display the MAC entry aging time, use the following command:

```
Switch(config)# display mac address-table aging-time
```

Note: For more information on the FDB available commands, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

Chapter 10. VLANs

This chapter describes network design and topology considerations for using Virtual Local Area Networks (VLANs). VLANs commonly are used to split up groups of network users into manageable broadcast domains, to create logical segmentation of workgroups, and to enforce security policies among logical segments. The following topics are discussed in this chapter:

VLAN Overview

A Local Area Network (LAN) is a group of inter-connected computers and other end devices that share the same limited geographical area, such as a school, residence, or office building. End devices in a LAN are physically separated from other end devices in different LANs.

A Virtual Local Area Network (VLAN) is a group of end devices that is logically separated by function or application, and not by their physical locations. A VLAN has the same attributes as a physical LAN, but you can group together end devices that are part of different LANs.

Setting up VLANs is a way of segmenting networks to increase network flexibility without changing the physical network topology. With network segmentation, each switch port connects to a segment that is a single broadcast domain. When a switch port is configured to be a member of a VLAN, it is added to a group of ports (workgroup) that belong to one broadcast domain.

Ports are grouped into broadcast domains by assigning them to the same VLAN. Frames received in one VLAN can only be forwarded within that VLAN, and unicast, multicast, and broadcast frames are forwarded and flooded only to ports belonging to the same VLAN.

VLAN Configuration

The switch supports up to 4094 VLANs. Each can be identified with a number between 1 and 4094.

VLANs are divided into four ranges as described in Table 19:

Table 19. *VLAN Ranges*

VLAN number	Usage
1	This is the default VLAN. It is created automatically and all switch ports are part of it by default. You cannot modify or delete this VLAN.
2-3999	You can create, modify, and delete any VLAN included in this range.
4000-4093	These VLANs are reserved by default. For more information, see "Reserved VLANs" on page 223
4094	This VLAN is reserved for internal use. You cannot create, modify, or delete this VLAN.
4095	This is reserved and unused in accordance with the IEEE 802.1Q standard.

To view the current VLAN information, use the following command:

```
Switch> display vlan

Flags:
u - untagged egress traffic for this VLAN
t - tagged egress traffic for this VLAN

VLAN    Name                          Status  IPMC FLOOD Ports

======= ============================= ======= ========== ===============
1       default                       ACTIVE  Disabled
                                                          Ethernet1/1(u)
2       VLAN000                       ACTIVE  Disabled
                                                          Ethernet1/1(t)
                                                          Ethernet1/2(u)
3       VN0003                        ACTIVE  Disabled
                                                          Ethernet1/1(t)
                                                          Ethernet1/3(u)
4       VLAN0004                      ACTIVE  Disabled
                                                          Ethernet1/1(t)
                                                          Ethernet1/4(u)
5       VLAN0005                      ACTIVE  Disabled
                                                          Ethernet1/1(t)
                                                          Ethernet1/5(u)
6       VLAN0006                      ACTIVE  Disabled
                                                          Ethernet1/1(t)
                                                          Ethernet1/6(u)
...
```

To view the reserved VLANs, enter:

```
Switch> display vlan reserved
Reserved VLAN range: 4000-4094
```

Note: VLAN 4094 is always reserved for internal use.

Creating a VLAN

You can create VLANs falling in the normal VLAN range.

You can create a single VLAN or a range of VLANs using a single command. The VLAN range can be either continuos (for example, 20-25) or discrete (for example, 20, 22, 25, 30).

The default VLAN (VLAN 1) is created automatically when the switch boots up for the first time and persists across later reloads.

When you create a new VLAN, it will remain unused until you assign a port to it. By default, all switch ports belong to the default VLAN (VLAN 1).

To create a VLAN, use the following command:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)#
```

Note: After running the preceding command, you will enter VLAN Configuration mode for the specified VLAN or range of VLANs.

For example:

```
Switch(config)# vlan 100-105
Switch(config-vlan)#
```

To view the current list of VLANs, use the following command:

```
Switch> display vlan

Flags:
u - untagged egress traffic for this VLAN
t - tagged egress traffic for this VLAN

VLAN    Name                          Status  IPMC FLOOD Ports

======= ============================= ======= ========== ===============
1       default                       ACTIVE  Disabled
                                                          Ethernet1/1(u)
                                                          ...
100     VLAN0100                      ACTIVE  Disabled
                                                          Ethernet1/1(t)
                                                          Ethernet1/2(u)
101     VLAN0101                      ACTIVE  Disabled
                                                          Ethernet1/3(t)
                                                          Ethernet1/4(u)
102     VLAN0102                      ACTIVE  Disabled
                                                          Ethernet1/5(t)
                                                          Ethernet1/6(u)
...
```

Deleting a VLAN

You can delete VLANs falling in the normal VLAN range.

You can delete a single VLAN or a range of VLANs using a single command. The VLAN range can be either continuos (for example, 20-25) or discrete (for example, 20, 22, 25, 30).

Note: You cannot delete the default VLAN or an internally reserved VLAN.

To delete a VLAN, use the following command:

```
Switch(config)# no vlan <VLAN ID (1-4093)>
```

For example:

```
Switch(config)# no vlan 100-105
```

You can delete a VLAN which is already associated to a port. If the VLAN is the access VLAN of a switch access port, the VLAN remains configured as the access VLAN, but the switch access port will not be able to forward any traffic until the VLAN is created again.

If the VLAN is the native VLAN of a switch trunk port, the VLAN remains configured as the native VLAN, but the switch trunk port will not be able to forward any untagged traffic until the VLAN is created again.

If the VLAN belongs to the allowed VLAN list of a switch trunk port, the VLAN remains in the allowed VLAN list, but the switch trunk port will not be able to forward any traffic carried in the VLAN until the VLAN is created again.

Configuring the State of a VLAN

You can change the operational state of a VLAN belonging to the normal VLAN range to be active or suspended. You can change the state of a single VLAN or a range of VLANs using a single command.

If a VLAN is in active state, it will forward traffic through the ports that are part of that VLAN. If the VLAN is in suspended state, it will not pass any traffic.

By default, when a VLAN is created, it will be in the active operational state.

Notes:

- You cannot change the state of the default VLAN (VLAN 1). It will always remain in the active operational state.

- You also cannot change the operational state of an internally reserved VLAN.

To change the operational state of a VLAN, use the following commands:

- to change the state to active:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# state active
```

- to change the state to suspended:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# state suspend
```

To reset the VLAN operational state to its default value, use the following command:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# no state
```

To view the current state of each VLAN, use the following command:

```
Switch> display vlan

Flags:
u - untagged egress traffic for this VLAN
t - tagged egress traffic for this VLAN

VLAN    Name                           Status  IPMC FLOOD Ports


======= ============================== ======= ========== ================
1       default                        ACTIVE  Disabled
                                                          Ethernet1/1(u)
                                                          ...
100     VLAN0100                       ACTIVE  Disabled
                                                          Ethernet1/1(t)
                                                          Ethernet1/2(u)
...
```

You can change the operation state of a VLAN to suspended even if that VLAN is already associated to a switch port. If the VLAN is the access VLAN of a switch access port, the VLAN remains configured as the access VLAN, but the switch access port will not be able to forward any traffic until the VLAN's state is changed to active.

If the VLAN is the native VLAN of a switch trunk port, the VLAN remains configured as the native VLAN, but the switch trunk port will not be able to forward any traffic carried in the VLAN until the VLAN's state is changed to active.

If the VLAN belongs to the allowed VLAN list of a switch trunk port, the VLAN remains in the allowed VLAN list, but the switch trunk port will not be able to forward any traffic carried in the VLAN until the VLAN's state is changed to active.

Reserved VLANs

Some features, such as Layer 3 ports (routed ports), require internal VLANs for their operations. You can reserve a contiguous block of VLANs to guarantee the delivery and operation of features with such requirements. These reserved VLANs cannot be created, deleted, modified, or manipulated.

To change the range of VLANs reserved for internal use, use the command:

```
Switch(config)# system vlan reserve <VLAN_range>
```

To reset the reserved VLAN range to the default (4000-4094), use the command:

```
Switch(config)# no system vlan reserve
```

To display the reserved VLAN range, use the command:

```
Switch# display system vlan reserved
```

Note: When reserving VLANs, consider the following:

- The reserved VLAN numbers must be a contiguous range.

- You cannot create, delete, modify, or manipulate the VLANs in the reserved VLAN range.

- VLAN 4094 is always reserved.

- The default VLAN cannot be used as a reserved VLAN.

- The operation fails if you try to change the range of reserved VLANs or delete those VLANs.

- You cannot reserve VLANs that already exist on switch, you must either delete them or select another range.

Note: We recommend that you configure the Reserved VLAN range before using the switch in production mode. If there are already routed ports configured, you will have to reboot the switch.

Configuring the Name of a VLAN

You can change the name of a VLAN belonging to the normal VLAN range to any alphanumeric string with a maximum length of 32 characters.

The name of a VLAN must be unique to that VLAN. No two VLANs can have the same identical name. The names are case-sensitive. For example, the name '*VLAN-test*' is considered different from '*VLAN-TEST*'.

The name of the default VLAN (VLAN 1) is '*default*'. For all other VLANs the default name used when creating that VLAN is in the format '*VLANxxxx*', where '*xxxx*' represent the four digits of the VLAN number (including leading zeros). For example, the default name of VLAN 2 is '*VLAN0002*'.

Note: You cannot change the name of the default VLAN or an internally reserved VLAN.

To change the name of a VLAN, use the following command:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# name <VLAN name>
```

To reset the name of a VLAN to its default value, use the following command:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# no name
```

To view the current names of each VLAN, use the following command:

```
Switch> display vlan

Flags:
u - untagged egress traffic for this VLAN
t - tagged egress traffic for this VLAN

VLAN    Name                          Status  IPMC FLOOD Ports

======= ============================= ======= ========== ===============
1       default                       ACTIVE  Disabled
                                                          Ethernet1/1(u)
                                                          ...
100     VLAN0100                      ACTIVE  Disabled
                                                          Ethernet1/1(t)
                                                          Ethernet1/2(u)
...
```

Configuring a Switch Access Port

You can configure an ethernet physical interface or a Link Aggregation Group (LAG) as a switch access port. By default, all the switch ports are in access mode.

A port in access mode can be part only of a single VLAN and it will carry traffic just for that one VLAN. That VLAN is called the Access VLAN.

To configure a port as an access port, use the following command (for this example ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# bridge-port mode access
```

To reset a port to its default operational mode, use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# no bridge-port mode
```

By default, the access VLAN for an interface is the default VLAN (VLAN 1). You can configure the access VLAN to be the default VLAN or a VLAN belonging to the normal VLAN range.

Configuring the Access VLAN

You can configure the access VLAN to be an uncreated VLAN. Until that VLAN is created, the switch access port will not be able to forward any traffic. As soon as it is created, the associated port will start forwarding traffic.

To change the access VLAN of a port, use the following command (for this example ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# bridge-port access vlan <VLAN ID (1-4093)>
```

To reset a port's access VLAN to the default access VLAN (VLAN 1), use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# no bridge-port access vlan
```

Note: Any access mode configuration settings are reset when the port's operational mode is changed from access to another mode.

A switch port in access operational mode accepts untagged, priority-tagged, and tagged traffic carried in the access VLAN and discard all tagged ingress traffic carried in other VLANs. Egress traffic out of the switch access port is sent untagged. For more details, see "Native VLAN Tagging" on page 229.

To view the current operational mode of a port, use the following command (for this example ethernet interface 1/12 is used):

```
Switch> display interface ethernet 1/12 bridge-port

Interface Ethernet1/12
  Bridge-port            : enabled
  Bridge-port mode       : access
  Ingress filter         : enable
  Tag Ingress PVID       : disabled
  Acceptable frame types : all
  Default/Native Vlan    : 110
  Configured Vlans       : 110
  Enabled Vlans          : 110
  Egress-Tagged Vlans    : None
```

Configuring a Switch Trunk Port

You can configure an ethernet physical interface or a Link Aggregation Group (LAG) as a switch trunk port. By default, all the switch ports are in access mode.

A port in trunk mode can be part of any number of VLANs and it will carry traffic for all configured VLANs.

To configure a port as a trunk port, use the following command (for this example ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# bridge-port mode trunk
```

To reset a port to its default operational mode, use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# no bridge-port mode
```

Configuring the Allowed VLAN List

The allowed VLAN list is the group of VLANs assigned to a trunk port for which traffic will be forwarded across the trunk link. By default, the allowed VLAN list consists of all VLANs (1-4094). The trunk port will forward traffic only for VLANs that are created on the switch. If a VLAN is in the allowed VLAN list, but it is not created, the trunk port will not forward traffic for that VLAN. If a VLAN that is in the allowed VLAN list is created after the port is configured as a trunk port, traffic for that VLAN will start to be forwarded as soon the VLAN is created.

For example, the currently created VLANs are the default VLAN and VLANs 100-105. When configuring ethernet interface 1/12 as a trunk port, by default its allowed VLAN list will include all VLANs, but will only forward traffic for VLANs 1 and 100-105. If VLAN 106 is created on the switch, ethernet interface 1/12 will start forwarding traffic for VLAN 106 as well.

You can configure an uncreated VLAN to a switch trunk port as a member of the allowed VLAN list. The switch trunk port will not be able to forward traffic carried in that VLAN until it is created.

To change the allowed VLAN list of a port, use the following command (for this example ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# bridge-port trunk allowed vlan <VLAN ID (1-4093)>
```

For example:

```
Switch(config-if)# bridge-port trunk allowed vlan 100-105,300,320
```

You can also add new VLANs to the current allowed VLAN list, remove VLANs from the list, or specify the list to contain all VLANs or none.

Considering the previous example, the allowed VLAN list currently contains VLANs 100-105, 300, and 320. The following command will add VLAN 110 to that list:

```
Switch(config-if)# bridge-port trunk allowed vlan add 110
```

If you want to remove VLAN 110 from the list, use the following command:

```
Switch(config-if)# bridge-port trunk allowed vlan remove 110
```

If you want to add all VLANs to the list, use the following command:

```
Switch(config-if)# bridge-port trunk allowed vlan all
```

If you want to add all VLANs to the list except VLAN 110, use the following command:

```
Switch(config-if)# bridge-port trunk allowed vlan except 110
```

If you do not want any allowed VLANs on the current port, use the following command:

```
Switch(config-if)# bridge-port trunk allowed vlan none
```

To reset a port's allowed VLAN list to its default settings, use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# no bridge-port trunk allowed vlan
```

Configuring the Native VLAN

When a port is configured as a switch trunk port, it will tag traffic packets with their appropriate VLAN number. All packets coming from the allowed VLAN list will be tagged when they are carried across the trunk, except for packets belonging to the native VLAN. Packets coming from the native VLAN are carried across the trunk untagged.

By default, the native VLAN for an interface is the default VLAN (VLAN 1). You can configure the native VLAN to be the default VLAN or a VLAN belonging to the normal VLAN range.

You can configure the native VLAN to be an uncreated VLAN. The switch trunk port will not be able to forward untagged, priority-tagged, or tagged traffic carried in that VLAN until the native VLAN is created and added to its allowed VLAN list.

To change the native VLAN of a port, use the following command (for this example ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# bridge-port trunk native vlan <VLAN ID (1-4093)>
```

To reset a port's native VLAN to the default native VLAN (VLAN 1), use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# no bridge-port trunk native vlan
```

Note: Any trunk mode configuration settings are reset when the port's operational mode is changed from trunk to another mode.

A port configured in trunk operational mode accepts tagged traffic carried in allowed VLANs and discard traffic carried in VLANs excluded from the allowed VLAN list on ingress side. Egress traffic carried in any allowed VLAN except the native VLAN is sent tagged on egress side.

The trunk port accepts untagged, priority-tagged, and tagged traffic carried in the native VLAN on the ingress side if native VLAN tagging is disabled on the port. Egress traffic carried in the native VLAN out of the trunk port is sent untagged on egress side if native VLAN tagging is disabled on the trunk port. Otherwise, the traffic is sent tagged with the native VLAN ID. For mode details, see "Native VLAN Tagging" on page 229.

To view the current operational mode of a port, use the following command (for this example ethernet interface 1/12 is used):

```
Switch> display interface ethernet 1/12 bridge-port

Interface Ethernet1/12
  Bridge-port          : enabled
  Bridge-port mode     : trunk
  Ingress filter       : enable
  Tag Ingress PVID     : disabled
  Acceptable frame types : all
  Default/Native Vlan  : 110
  Configured Vlans     : 100-105,300,320
  Enabled Vlans        : 100-105,300,320
  Egress-Tagged Vlans  : 100-105,300,320
```

Native VLAN Tagging

When a packet enters a switch trunk link, a tag is added to its frame header to represent the VLAN membership of the frame's port. Each frame must be distinguishable as being within exactly one VLAN. A frame entering the trunk link that does not contain a VLAN tag is assumed to be flowing on the native VLAN.

This can lead to a security vulnerability in the network. It is advised to enable the tagging of native VLAN traffic. This ensures that all packets going out of the trunk link will be tagged and it prevents the reception of untagged packets.

If native VLAN tagging is disabled, the switch is vulnerable to VLAN hopping attacks. The basic concept behind a VLAN hopping attack is for an attacking host on a VLAN to gain access to traffic on other VLANs that would normally not be accessible.

To address such potential attacks, native VLAN tagging performs the following:

● on the ingress side, all untagged data traffic is dropped when native VLAN tagging is not explicitly configured for egress traffic

● on the egress side, all traffic is tagged; if traffic belongs to the native VLAN then it is tagged with the native VLAN ID

Native VLAN tagging can be configured globally or on each interface. The switch port must be in trunk operational mode for native VLAN tagging to function.

By default, native VLAN tagging is globally disabled on the switch. Globally enabling or disabling native VLAN tagging affects all trunk ports that do not have native VLAN tagging individually configured, which is the default setting for 802.1Q trunk ports. Until a native VLAN tagging is configured on each trunk port, individual tagging behavior is dependent on the global configuration.

When native VLAN tagging is disabled on a switch trunk port, all untagged ingress packets are accepted and carried in the native VLAN. All tagged packets belonging to VLANs in the allowed VLAN list are accepted as long as that VLAN is active and tagged packets belonging to VLANs not in the allowed VLAN list are discarded. If the ingress packet is priority-tagged, it is regarded as an untagged packet and it is carried in the native VLAN.

If an egress packet belongs to the native VLAN then it is sent untagged. Other tagged packets belonging to VLANs in the allowed VLAN list are sent with the VLAN preserved. If the egress packet is priority-tagged and belongs to the native VLAN, it is sent untagged (the priority tag is removed). Otherwise, the priority-tagged packet is tagged with a new VLAN ID and its priority tag is removed in advance. In other words, the packet is not sent in Q-in-Q format.

When native VLAN tagging is enabled on a switch trunk port, all untagged ingress packets will be discarded unless native VLAN tagging is not explicitly configured for egress traffic. Otherwise, all untagged ingress packets are accepted in the trunk link and carried in the native VLAN. If the ingress packet is priority-tagged, it is accepted in the native VLAN only if tagging is explicitly configured for egress traffic. Otherwise, the packet is discarded.

All egress packets are tagged. If a packet belongs to the native VLAN then it is tagged with the native VLAN ID. If the egress packet is priority-tagged, it is tagged with a new VLAN ID and its priority tag is removed in advance. In other words, the packet is not sent in Q-in-Q format.

Note: Control protocol data units (PDU) are accepted as untagged on the native VLAN of a switch trunk port, even if native VLAN tagging is enabled.

Configuring Native VLAN Tagging

Native VLAN tagging is supported both on physical ethernet interfaces and on Link Aggregation Groups (LAG). A LAG inherits the native VLAN tagging configuration of its first trunk port member. When adding a switch trunk port to a LAG, the port must have the same native VLAN tagging configuration as the LAG. Otherwise, the trunk port will be rejected from the LAG.

To globally enable or disable native VLAN tagging, use the following command:

```
Switch(config)# [no] vlan dot1q tag native
```

If you want to enable native VLAN tagging just for egress traffic, use the following command:

```
Switch(config)# vlan dot1q tag native egress-only
```

Note: If native VLAN tagging is enabled only for egress traffic, all untagged ingress packets are accepted in the switch trunk port and are carried in the native VLAN. Otherwise, the ingress packets are discarded.

Native VLAN tagging can be configured independently on each ethernet interface or LAG. To enable or disable tagging on an interface, use the following command (for this example, ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vlan dot1q tag native enable
```

To enable native VLAN tagging only for egress traffic, use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vlan dot1q tag native enable egress-only
```

To disable native VLAN tagging on an interface, use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vlan dot1q tag native disable
```

To delete native VLAN tagging configuration of an interface, use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# no vlan dot1q tag native
```

Note: This will make native VLAN tagging on that interface dependent on the global configuration.

To view the current native VLAN tagging settings for all interfaces, use the following command:

```
Switch> display vlan dot1q tag native

Tag native vlan global setting: enabled
----------------------------------------------------------
Port Tag Native VLAN Status
----------------------------------------------------------
Eth1 enabled
Eth5 disabled
Eth13 enabled (egress-only)
...
```

Port VLAN ID Ingress Tagging

The Port VLAN ID (PVID) is the default VLAN ID assigned to a switch port to designate the VLAN of which the port is a member. Any packets received by the switch on such a port will have a tag added to their frame containing the PVID of the port. The packets will then be forwarded in the VLAN corresponding to the PVID of the receiving port.

PVID Ingress Tagging offers tunneling support at the ingress port level through the insertion of VLAN tags.

For egress traffic, through the use of Native VLAN Tagging the switch can choose to remove or not the outer VLAN tag of a packet if it is the Native VLAN ID of the egress port.

For ingress traffic, the default action of the switch is to add the PVID of the ingress port to any untagged packets. If the arriving packet is tagged with a VLAN ID, the switch does not add the PVID tag and the existing VLAN ID tag is used as the outer VLAN tag.

When PVID Ingress Tagging is enabled on a switch port, packets received on that port are tagged with the PVID of the port regardless if the packet is already tagged or it is untagged. If the packet is already tagged, its original VLAN ID tag is not removed and the PVID of the receiving port is added over the existing VLAN ID, resulting a double tagged packet.

PVID Ingress Tagging adds the Native VLAN ID as the outer VLAN tag for switch ports configured as trunk ports, and the Access VLAN ID for switch ports configured as access ports.

Notes:

- PVID Ingress Tagging affects only ingress traffic, while Native VLAN Tagging affects only egress traffic.

- PVID Ingress Tagging cannot be enabled on the switch management interface.

By default, PVID Ingress Tagging is disabled. To enable or disable PVID Ingress Tagging on a switch port, use the following command:

```
Switch(config)# interface ethernet <chassis number>/<port number>
Switch(config-if)# [no] bridge-port mode dot1q-tunnel
```

To display the current PVID Ingress Tagging settings, use the following command:

```
Switch> display dot1q-tunnel [interface ethernet <chassis number>/<port number>|
|port-aggregation <1-4096>]
```

IP Subnet VLAN Assignment

The IP subnet VLAN assignment feature lets you configure the switch to assign a VLAN based on the IP subnet for incoming untagged or priority-tagged packets. This feature can also assign a priority to untagged traffic.

IP subnet VLAN mappings are configurable and are maintained as a table similar to Table 20:

Table 20. *Example of IP Subnet VLAN Mappings*

Source IP Address	Subnet Mask	VLAN ID	Priority (Optional)
192.168.1.0	255.255.255.0	2	7
192.168.2.0	255.255.255.0	3	5
192.168.1.5	255.255.255.255	4	6

For example, if the incoming traffic source IP address belongs to subnet 192.168.2.0/24, the traffic will be assigned to VLAN 3 with priority 5. The rules are global and apply to all enabled ports.

The subnet VLAN table is a longest prefix match table. For example, if the source IP address of the received untagged packet is 192.168.1.5, it will match two entries in Table 20. Since 192.168.1.5/32 is the longest-prefix match, this packet will be assigned to VLAN 4 with priority 6.

The IP subnet VLAN assignment function can be controlled per ethernet port or port aggregation, and only takes effect on enabled ports. By default, this function is disabled on all ports. When dealing with a port aggregation, the following rules apply:

● The configuration will apply to all members of the port aggregation.

● All member ports of an aggregation must have the same configuration.

● When ports are removed from an aggregation, the IP subnet VLAN configuration remains on them.

IP subnet VLAN assignment is disabled by default on all ports. To enable this feature on a port, enter:

```
Switch(config)# interface ethernet <chassis>/<port>
Switch(config-if)# vlan classifier subnet-vlan enable
Switch(config-if)# exit
```

To configure IP subnet VLAN assignment, configure the IP subnet based VLAN classification rule:

```
Switch(config)# vlan classifier subnet-vlan ip {<IP address/mask> |
<IP address>/<submask>} vlan <VLAN ID> [priority <priority>]
```

where:

Parameter	Description
IP address	An IPv4 address.
mask	Shorthand mask for an IP address.
submask	If the IPv4 address does not have a submask (/M), you need to use one in the form of an IPv4 address.
VLAN ID	The ID of the target VLAN; an integer from 1-4093.
priority	(Optional) The priority; an integer from 0-7.

To delete the configuration rule based on the IP address or target VLAN ID, enter:

```
Switch(config)# no vlan classifer subnet-vlan [ip {<IP address/mask> |
<IP address>/<submask>} | vlan <VLAN ID>]
```

If no parameters are specified, all rules are deleted.

To enable or disable the subnet VLAN classification on ports, in Interface Configuration mode, enter:

```
Switch(config-ip)# [no] vlan classifer subnet-vlan enable
```

To display the mapping rule base for a specific IP address or VLAN ID, enter:

```
Switch# display vlan classifer subnet-vlan [ip {<IP address/mask> |
<IP address>/<submask>} | vlan <VLAN ID>]
```

If no parameters are specified, all rules are displayed.

To display the IP subnet VLAN state of an interface, enter:

```
Switch# display vlan classifer subnet-vlan state [interface
{ethernet <chassis number>/<port number> | port-aggregation
<port-aggregation (1-4096)>}]}
```

To display the VLAN classification-related information in the running configuration, enter:

```
Switch# display running-config vlan classifer
```

IPMC Flooding

IP Multicast (IPMC) is a method of sending IP traffic to a group of interested devices in a single transmission. When sending a multicast packet, a device uses an IP multicast address from the range:

- for IPv4: `224.0.0.0/4` (`224.0.0.0` - `239.255.255.255`)

- for IPv6: `ff00::/8`

If the switch does not understand multicast addresses then it will flood the unknown traffic to all the members of a VLAN. In this scenario the switch has to filter the packets sent to multicast groups they are not subscribed to.

By default, IPMC flooding is enabled on all VLANs, meaning that unknown IP multicast traffic is forwarded across all VLAN member interfaces. If IPMC flooding is disabled, the switch will discard all unknown multicast IP traffic that has a destination IP address outside the reserved multicast range. For IPv4 addressing that range is 224.0.0.X (224.0.0.0 - 224.0.0.255).

To enable or disable the flooding of unknown IPMC traffic in a VLAN, use the following command:

```
Switch(config-vlan)# [no] flood
```

To enable or disable IPMC flooding for a specific family of IP addresses, use the following command:

- for IPv4:

```
Switch(config-vlan)# [no] flood ipv4
```

- for IPv6:

```
Switch(config-vlan)# [no] flood ipv6
```

To check the IPMC flood configuration of a VLAN, use the following command:

```
Switch> display vlan id <1-4096>

Flags:
u - untagged egress traffic for this VLAN
t - tagged egress traffic for this VLAN
VLAN    Name                     Status  IPMC FLOOD Ports


======= ======================== ======= ========== ====================
1       default                  ACTIVE  Disabled
                                                     Ethernet1/3(u)
                                                     Ethernet1/4(u)
                                                     Ethernet1/5(u)
...
```

Hybrid Bridge-Port Mode

Hybrid bridge-port mode is a trunk bridge-port mode that lets you have more than one egress untagged VLAN. Like a trunk port, a hybrid port can carry multiple VLANs to receive and pass traffic for them. The hybrid bridge-port mode lets you control which VLANs receive tagged vs. untagged egress traffic. Unlike trunk bridge-port mode, the native VLAN is not considered the only VLAN that can send untagged traffic.

Hybrid Bridge-Port Mode Rules

The following rules apply to hybrid bridge-port mode:

- You can configure an ethernet or port-aggregation interface in hybrid mode.

- A hybrid port can have any number of VLANs configured on the interface and can carry traffic for several VLANs simultaneously.

- The hybrid port is a trunk port that can have more than one egress untagged VLAN. Unlike a trunk port, a hybrid port by default allows traffic of all the allowed range of VLANs to pass through untagged; the native VLAN is not the only VLAN that can send untagged traffic.

- VLAN options set for hybrid ports apply only to outgoing traffic.

- A hybrid port has a native VLAN and an allowed VLAN list.

- You can configure either the native VLAN or a VLAN in the normal VLAN range to be the default VLAN. By default, the native VLAN is VLAN 1.

- You can add or remove the default VLAN or VLANs in the normal VLAN range to or from the allowed VLAN list. By default, the allowed VLAN list consists of all VLANs that are not reserved.

- If DOT1Q tunnel mode is not enabled on the bridge hybrid port, the bridge hybrid port will accept tagged traffic carried on allowed VLANs and discards traffic carried in VLANs excluded from the allowed VLAN list on the ingress side.

- If the native VLAN is part of the allowed VLAN list, the bridge hybrid port will accept untagged, priority-tagged, and tagged traffic from the native VLAN on the ingress side.

- Based on egress tagging options, the egress traffic carried by the native VLAN out of the bridge hybrid port is sent untagged or tagged with the native VLAN ID. By default, a hybrid port sends outgoing native VLAN traffic untagged. If egress tagging is set on the native VLAN, the hybrid port sends the traffic from the native VLAN tagged with the native VLAN ID

- You can set the egress traffic type on a list of VLANs included in the allowed VLAN list of a bridge hybrid port.

- All configurations related to hybrid bridge-port mode are lost when you change the hybrid port to another mode.

Configuring a Hybrid Bridge-Port

To configure a port as a hybrid bridge-port:

1. In Interface Configuration mode, set the bridge port to function in hybrid mode:

```
Switch(config-if)# bridge-port mode hybrid
```

2. Set the native VLAN for the hybrid bridge-port:

```
Switch(config-if)# bridge-port hybrid native vlan <VLAN ID>
```

where *VLAN ID* is the ID of the native VLAN.

Note: To set the native VLAN for hybrid mode back to the default VLAN, enter:

```
Switch(config-if)# no bridge-port hybrid native vlan
```

3. Add VLANs to or remove VLANs from the allowed VLAN list for the hybrid bridge-port:

```
Switch(config-if)# bridge-port hybrid allowed vlan {<VLAN_LIST> | add
<VLAN_LIST> | all | except <VLAN_LIST> | none | remove <VLAN_LIST>}
```

where:

Parameter	Description
VLAN LIST	A range or comma-separated list of allowed VLANs.
add	Add the following VLANs to the allowed list.
except	Add all VLANs except the following.
none	No VLANs.
remove	Remove the following VLANs from the allowed list.

Note: To reset the allowed VLAN list for the hybrid bridge-port to the default (all allowed), enter:

```
Switch(config-if)# no bridge-port hybrid allowed vlan
```

4. Set (or unset) the egress traffic type to tagged for specific VLANs:

```
Switch(config-if)# [no] bridge-port vlan egress-tagged vlan <VLAN_LIST>
```

where *VLAN_LIST* is the list of VLANs to use tagged egress traffic. The default is untagged VLAN traffic.

To see the hybrid bridge-port settings of a specific interface, enter:

```
Switch# display interface <interface> bridge-port
```

To see information about all hybrid bridge-ports associated with the specified VLANs, enter:

```
Switch# display interface hybrid vlan <VLAN ID or VLAN_LIST>
```

To get information about hybrid bridge-ports in the running configuration, enter:

```
Switch# display running-config interface {ethernet
<chassis number>/<port number> | port-aggregation <port-aggregation (1-4096)>}
```

To see information about bridging characteristics of the Layer 2 interface, enter::

```
Switch# display interface bridge-port
```

VLAN Topologies and Design Considerations
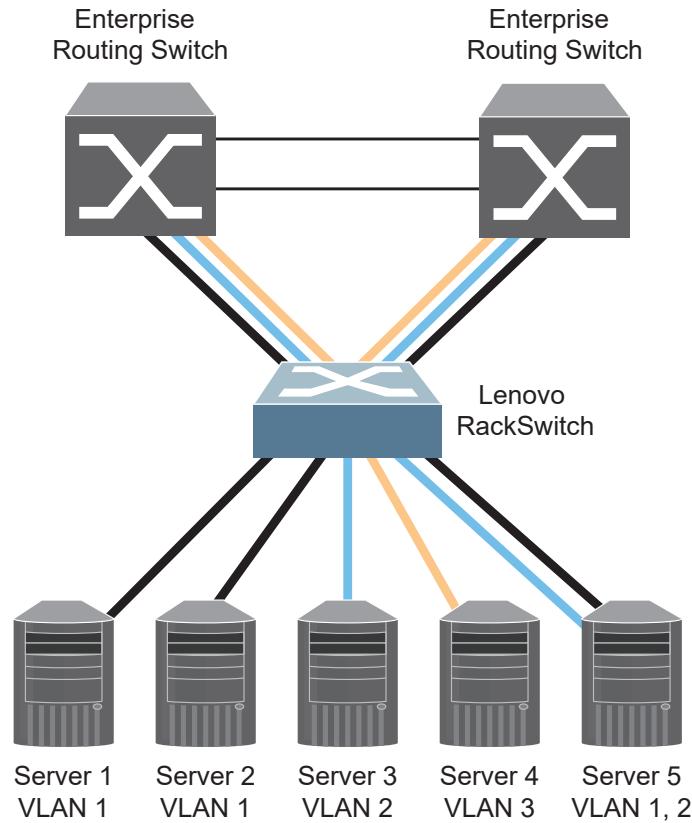
Note the following when working with VLAN topologies:

- By default, the switch software is configured so that trunk operational mode and native VLAN tagging are disabled on all switch ports.

- By default, the switch software is configured so that all switch ports are members of the default VLAN (VLAN 1).

- When using the Multiple Spanning Tree Protocol (MSTP), Spanning Tree Groups (STG) 1 to 64 can include multiple VLANs.

- All ports involved in both aggregation and port mirroring must have the same VLAN configuration. If a port is on a LAG with a mirroring port, the VLAN configuration cannot be changed. For more information about aggregation, see Chapter 11, "Ports and Link Aggregation" and Chapter 34, "Port Mirroring".

Multiple VLANs with Trunk Mode Adapters

Figure 2 illustrates a network topology described in Note: on page 241 and the configuration example on page page 242.

Figure 2. Multiple VLANs with VLAN-Tagged Gigabit Adapters

The features of this VLAN are described in the following table.

Component	Description
Lenovo RackSwitch	This switch is configured with three VLANs that represent three different IP subnets. Five ports (ethernet ports 1/11-1/15) are connected downstream to servers. Two ports (ethernet ports 1/19 and 1/20) are connected upstream to routing switches. Uplink ports are members of all three VLANs and are in trunk mode.
Server 1	This server is a member of VLAN 1 and has presence in only one IP subnet. The associated switch port (ethernet port 1/11) is only a member of VLAN 1, so the port is in access mode.
Server 2	This server is a member of VLAN 1 and has presence in only one IP subnet. The associated switch port (ethernet port 1/12) is only a member of VLAN 1, so the port is in access mode.
Server 3	This server belongs to VLAN 2 and it is logically in the same IP subnet as Server 5. The associated switch port (ethernet port 1/13) is in access mode.
Server 4	A member of VLAN 3, this server can communicate only with other servers via a router. The associated switch port (ethernet port 1/14) is in access mode.
Server 5	A member of VLAN 1 and VLAN 2, this server can communicate with Server 1, Server 2, Server 3 and the upstream switches. It cannot communicate with Server 4. The associated switch port (ethernet port 1/15) is in trunk mode.
Enterprise Routing Switches	These switches must have all three VLANs (VLAN 1, 2 and 3) configured. They can communicate with Server 1, Server 2 and Server 5 via VLAN 1. They can communicate with Server 3 and Server 5 via VLAN 2. They can communicate with Server 4 via VLAN 3. The switch ports are in trunk mode.

Note: Switch trunk port mode is required only on ports that are connected to other switches or on ports that connect to tag-capable end stations, such as servers with trunk mode adapters.

To configure a specific VLAN on a trunk port, the following conditions must be met:

● The port must be in trunk mode.
● The VLAN must be in the trunk port's allowed VLAN range. By default, the range includes all VLANs.
● The VLAN must be un-reserved.
● The VLAN must be created.

The order in which the conditions above are met is not relevant. However, all conditions must be met collectively. When all the conditions are met, the VLAN is enabled on the port. If one of the conditions is broken, the VLAN is disabled.

VLAN Configuration Example

Use the following procedure to configure the example network shown in .

1. Enable trunk mode on server ports that support multiple VLANs and add VLANs to the trunk link.

```
Switch(config)# interface ethernet 1/15
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlans 1,2
Switch(config-if)# exit
```

2. Enable trunk mode on uplink ports that support multiple VLANs.

```
Switch(config)# interface ethernet 1/19
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# exit

Switch(config)# interface ethernet 1/20
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# exit
```

3. Configure server ports that belong to a single VLAN.

```
Switch(config)# interface ethernet 1/13
Switch(config-if)# bridge-port access vlan 2
Switch(config-if)# exit

Switch(config)# interface ethernet 1/14
Switch(config-if)# bridge-port access vlan 3
Switch(config-if)# exit
```

By default, all ports are members of the default VLAN (VLAN 1), so configure only those ports that belong to other VLANs. In this example, ethernet ports 1/11 to 1/12 are by default switch access ports and are members of VLAN 1, and do not require any further configuration.

Chapter 11. Ports and Link Aggregation

Link Aggregation Groups (LAGs) can provide super-bandwidth, multi-link connections between the switch and other LAG-capable devices. A LAG is a group of ports that act together, combining their bandwidth to create a single, larger virtual link. This chapter provides configuration background and examples for aggregating multiple ports together:

Port Configuration Profiles

The following Lenovo switches are 1U rack-mountable aggregation devices:

- RackSwitch G8272

- RackSwitch G8332

- ThinkSystem NE1032T RackSwitch

- ThinkSystem NE1032 RackSwitch

- ThinkSystem NE1072T RackSwitch

- ThinkSystem NE10032 RackSwitch

- ThinkSystem NE2572 RackSwitch

The Lenovo RackSwitch G8296 is a 2U rack-mountable aggregation device.

All of these switches use a wire-speed, non-blocking switching fabric that provides simultaneous wire-speed transport of multiple packets at low latency on all ports.

G8272 Port Configuration

The G8272 has the following port capabilities:

- Forty-eight 10 Gigabit Ethernet (GbE) Small Form Pluggable Plus (SFP+) ports which also support legacy 1 GbE connections

- Six 40 GbE Quad Small Form Pluggable Plus (QSFP+) ports, each of which can optionally be used as four 10 GbE ports

SPF+ ports, when used in 10 GbE mode, can be populated with optical transceiver modules, or active or passive Direct-Attach Cables (DACs). When used in legacy 1 GbE mode, the ports can be populated with optical or copper transceiver modules.

QSFP+ ports can be populated with optical QSFP+ modules or passive DACs, including those that allow the port to be used as four 10 GbE ports.

By default, the G8272 has the following interface (port) configuration:

- ethernet interfaces 1-48 are configured as 10 GBit/s ports

- ethernet interfaces 49-54 are configured as 40 GBit/s ports

The G8272 supports three different port profiles:

- default - forty-eight 10 GBit/s SFP+ ports and six 40 GBit/s QSFP+ ports

- 72x10G - seventy-two 10 GBit/s SFP+ ports

- custom port mode

In 72x10G port mode, each of the six 40 GBit/s ports is split into four 10 GBit/s breakout ports, adding up to a total of seventy-two 10 GBit/s ports.

To change the port mode to the 72x10G profile, use the following command:

```
Switch(config)# hardware profile portmode 72x10G
```

You can configure a custom port profile on the switch, changing an ethernet interface port mode from 40 GBit/s to 10 GBit/s or vice versa, and have different numbers of 10 GBit/s and 40 GBit/s ports than the default configuration. For example, you can configure sixty-four 10 GBit/s ports and two 40 Gbit/s ports.

Note: The maximum number of 40 GBit/s ports configurable on the switch is six.

To create a custom port profile, use the following command:

```
Switch(config)# hardware profile portmode custom 4x 10G ethernet <chassis
number>/<port number>
```

Note: This command will configure the specified port to run in 10 GBit/s mode.

You can specify a single ethernet port or a range of ports. The range can be either continuos (for example, 20-25) or discrete (for example, 20, 22, 25, 30).

For example:

● Configure a single ethernet interface to run in 10 GBit/s mode:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet 1/50
```

● Configure a range of ethernet interfaces to run in 10 GBit/s mode:
   ○ continuos range:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet
1/51-54
```

   ○ discrete range:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet
1/49, ethernet 1/53, ethernet 1/54
```

When a 40 GBit/s port is split into four 10 GBit/s breakout ports, the ethernet interface associated with that port will include four subports.

For example, before changing ethernet port 54 to 10 GBit/s port mode, the port will be displayed as follows:

```
Switch> display interface ethernet 1/54 brief

-------------------------------------------------------------------------
Ethernet      VLAN  Type Mode   Status  Reason              Speed  Port
Interface                                                          Agg#
-------------------------------------------------------------------------
Ethernet1/54  1     eth  access down    Link not connected  auto   --
```

After spliting the port into four 10 GBit/s breakout ports, the port will be displayed as follows:

```
Switch> display interface ethernet 1/54/1-4 brief

-------------------------------------------------------------------------
Ethernet      VLAN Type Mode    Status Reason              Speed Port
Interface                                                        Agg#
-------------------------------------------------------------------------
Ethernet1/54/1 1   eth  access down    Link not connected  auto  --
Ethernet1/54/2 1   eth  access down    Link not connected  auto  --
Ethernet1/54/3 1   eth  access down    Link not connected  auto  --
Ethernet1/54/4 1   eth  access down    Link not connected  auto  --
```

To configure 10 GBit/s breakout ports, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number/subport number>
```

For example:

```
Switch(config)# interface ethernet 1/54/2
```

When configuring a custom port profile on an ethernet interface, you can reset the interface's port mode to its default settings by using the following command:

```
Switch(config)# no hardware profile portmode custom ethernet <chassis number/
/port number/subport number>
```

To reset the switch port configuration to the default profile, use the following command:

```
Switch(config)# hardware profile portmode default
```

To view the current configured port profile, use the following command:

```
Switch# display hardware profile portmode

Current port mode:
        Port Ethernet1/49 is set in 10G mode
        Port Ethernet1/50 is set in 10G mode
        Port Ethernet1/51 is set in 10G mode
        Port Ethernet1/52 is set in 10G mode
        Port Ethernet1/53 is set in 10G mode
        Port Ethernet1/54 is set in 10G mode
```

G8296 Port Configuration

The G8296 has the following port capabilities:

● Eighty-six 10 Gigabit Ethernet (GbE) Small Form Pluggable Plus (SFP+) ports which also support legacy 1 GbE connections

● Ten 40 GbE Quad Small Form Pluggable Plus (QSFP+) ports, of which two ports (87 and 88) can optionally be used as four 10 GbE ports

SPF+ ports, when used in 10 GbE mode, can be populated with optical transceiver modules, or active or passive Direct-Attach Cables (DACs). When used in legacy 1 GbE mode, the ports can be populated with optical or copper transceiver modules.

QSFP+ ports can be populated with optical QSFP+ modules or passive DACs, including those that allow the port to be used as four 10 GbE ports.

By default, the G8296 has the following interface (port) configuration:

● ethernet interfaces 1-86 are configured as 10 GBit/s ports

● ethernet interfaces 87-96 are configured as 40 GBit/s ports

The G8296 supports three different port profiles:

● default - eighty-six 10 GBit/s SFP+ ports and ten 40 GBit/s QSFP+ ports

● 94x10G + 8x40G - ninety-four 10 GBit/s SFP+ ports and eight 40 GBit/s QSFP+ ports

● custom port mode

In 94x10G + 8x40G port mode, two of the ten 40 GBit/s ports are split into four 10 GBit/s breakout ports, adding up to a total of ninety-four 10 GBit/s ports and eight 40 GBit/s ports. The two 40 GBit/s ports that can be split into four 10 GBit/s breakout ports are ethernet ports 87 and 88.

To change the port mode to the 94x10G + 8x40G profile, use the following command:

```
Switch(config)# hardware profile portmode 94x10G+8x40G
```

You can configure a custom port profile on the switch, changing an ethernet interface port mode from 40 GBit/s to 10 GBit/s or vice versa, and have different numbers of 10 GBit/s and 40 GBit/s ports than the default configuration. For example, you can configure ninety 10 GBit/s ports and nine 40 Gbit/s ports.

Note: The maximum number of 40 GBit/s ports configurable on the switch is ten. The minimum number is eight.

To create a custom port profile, use the following command:

```
Switch(config)# hardware profile portmode custom ethernet <chassis number/
/port number>
```

Note: This command will configure the specified port to run in 10 GBit/s mode.

You can specify a single ethernet port or a range of ports. The range can be either continuos (for example, 20-25) or discrete (for example, 20, 22, 25, 30).

For example:

● Configure a single ethernet interface to run in 10 GBit/s mode:

```
Switch(config)# hardware profile portmode custom ethernet 1/88
```

● Configure a range of ethernet interfaces to run in 10 GBit/s mode:
   o continuos range:

```
Switch(config)# hardware profile portmode custom ethernet 1/87-88
```

   o discrete range:

```
Switch(config)# hardware profile portmode custom ethernet 1/87,
ethernet 1/88
```

When a 40 GBit/s port is split into four 10 GBit/s breakout ports, the ethernet interface associated with that port will include four subports.

For example, before changing ethernet port 88 to 10 GBit/s port mode, the port will be displayed as follows:

```
Switch> display interface ethernet 1/88 brief

-------------------------------------------------------------------------
Ethernet      VLAN  Type Mode   Status  Reason               Speed  Port
Interface                                                           Agg#
-------------------------------------------------------------------------
Ethernet1/88  1     eth  access down    Link not connected   auto   --
```

After spliting the port into four 10 GBit/s breakout ports, the port will be displayed as follows:

```
Switch> display interface ethernet 1/88/1-4 brief

-------------------------------------------------------------------------
Ethernet      VLAN  Type Mode   Status  Reason               Speed  Port
Interface                                                           Agg#
-------------------------------------------------------------------------
Ethernet1/88/1 1    eth  access down    Link not connected   auto   --
Ethernet1/88/2 1    eth  access down    Link not connected   auto   --
Ethernet1/88/3 1    eth  access down    Link not connected   auto   --
Ethernet1/88/4 1    eth  access down    Link not connected   auto   --
```

To configure 10 GBit/s breakout ports, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number/subport number>
```

For example:

```
Switch(config)# interface ethernet 1/88/2
```

When configuring a custom port profile on an ethernet interface, you can reset the interface's port mode to its default settings by using the following command:

```
Switch(config)# no hardware profile portmode custom ethernet <chassis number/
/port number/subport number>
```

To reset the switch port configuration to the default profile, use the following command:

```
Switch(config)# hardware profile portmode default
```

To view the current configured port profile, use the following command:

```
Switch# display hardware profile portmode

Current port mode:
        Port Ethernet1/87 is set in 10G mode
        Port Ethernet1/88 is set in 10G mode
```

G8332 Port Configuration

The G8332 has the following port capabilities:

● Thirty-two 40 GbE Quad Small Form Pluggable Plus (QSFP+) ports, of which twenty-four ports (2 to 25) can optionally be used as four 10 GbE ports

QSFP+ ports can be populated with optical QSFP+ modules or passive DACs, including those that allow the port to be used as four 10 GbE ports.

By default, the G8332 has the following interface (port) configuration:

● ethernet interfaces 1-32 are configured as 40 GBit/s ports

The G8332 supports three different port profiles:

● default - thirty-two 40 GBit/s QSFP+ ports

● 96x10G + 8x40G - ninety-six 10 GBit/s SFP+ ports and eight 40 GBit/s QSFP+ ports

● custom port mode

In 96x10G + 8x40G port mode, twenty-four of the thirty-two 40 GBit/s ports are split into four 10 GBit/s breakout ports, adding up to a total of ninety-six 10 GBit/s ports and eight 40 GBit/s ports. The twenty-four 40 GBit/s ports that can be split into four 10 GBit/s breakout ports are ethernet ports from 2 to 25 (included).

To change the port mode to the 96x10G + 8x40G profile, use the following command:

```
Switch(config)# hardware profile portmode 96x10G+8x40G
```

You can configure a custom port profile on the switch, changing an ethernet interface port mode from 40 GBit/s to 10 GBit/s or vice versa, and have different numbers of 10 GBit/s and 40 GBit/s ports than the default configuration. For example, you can configure ninety-two 10 GBit/s ports and nine 40 Gbit/s ports.

Note: The maximum number of 40 GBit/s ports configurable on the switch is thirty-two. The minimum number is eight.

To create a custom port profile, use the following command:

```
Switch(config)# hardware profile portmode custom <port mode> ethernet <chassis
number>/<port number>
```

where:

Argument	Description
port mode	The number of ports x the speed, such as `96x10G` or `10x40G`.
chassis number	The ethernet chassis number.
slot number	The ethernet slot number.

Note: This command will configure the specified port to run in 10 GBit/s mode.

You can specify a single ethernet port or a range of ports. The range can be either continuos (for example, 20-25) or discrete (for example, 20, 22, 25, 30).

For example:

● Configure a single ethernet interface to run in 10 GBit/s mode:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet 1/2
```

● Configure a range of ethernet interfaces to run in 10 GBit/s mode:
  ○ continuos range:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet
1/2-10
```

  ○ discrete range:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet
1/11, ethernet 1/12
```

When a 40 GBit/s port is split into four 10 GBit/s breakout ports, the ethernet interface associated with that port will include four subports.

For example, before changing ethernet port 2 to 10 GBit/s port mode, the port will be displayed as follows:

```
Switch> display interface ethernet 1/2 brief

--------------------------------------------------------------------------------
Ethernet        VLAN   Type Mode   Status  Reason                     Speed   Port
Interface                                                                      Agg#
--------------------------------------------------------------------------------
Ethernet1/2     1      eth  access down    Administratively down      40000   --
```

After spliting the port into four 10 GBit/s breakout ports, the port will be displayed as follows:

```
Switch> display interface brief

-------------------------------------------------------------------------------
Ethernet        VLAN  Type Mode   Status Reason                  Speed   Port
Interface                                                                Agg#
-------------------------------------------------------------------------------
Ethernet1/1     1     eth  access down   Administratively down   40000   --
Ethernet1/2/1   1     eth  trunk  up     none                    10000   --
Ethernet1/2/2   1     eth  access down   Administratively down   10000   --
Ethernet1/2/3   1     eth  access down   Administratively down   10000   --
Ethernet1/2/4   1     eth  access down   Administratively down   10000   --
Ethernet1/3/1   1     eth  access down   Administratively down   10000   --
Ethernet1/3/2   1     eth  access down   Administratively down   10000   --
Ethernet1/3/3   1     eth  access down   Administratively down   10000   --
```

To configure 10 GBit/s breakout ports, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number/subport number>
```

For example:

```
Switch(config)# interface ethernet 1/2/2
```

When configuring a custom port profile on an ethernet interface, you can reset the interface's port mode to its default settings by using the following command:

```
Switch(config)# no hardware profile portmode custom ethernet <chassis number/
/port number/subport number>
```

To reset the switch port configuration to the default profile, use the following command:

```
Switch(config)# hardware profile portmode default
```

To view the current configured port profile, use the following command:

```
Switch# display hardware profile portmode

Current port mode:
Port Ethernet1/2 is set in 10G mode
Port Ethernet1/3 is set in 10G mode
```

NE1072T Port Configuration

The NE1072T has the following port capabilities:

- Forty-eight 1000/10G BASE-T RJ45 ports.

- Six 40 GbE Quad Small Form Pluggable Plus (QSFP+) ports, all of which ports can optionally be used as four 10 GbE SFP+ ports

The 10G BASE-T RJ45 ports, when used in 10 GbE mode, must use CAT6 copper cabling. When used in 1000 base T mode, the ports can be populated with CAT5E copper cabling.

QSFP+ ports can be populated with optical QSFP+ modules or passive DACs, including those that allow the port to be used as four 10 GbE ports.

By default, the NE1072T has the following interface (port) configuration:

- ethernet interfaces 1-48 are configured as 10 GBit/s ports

- ethernet interfaces 49-54 are configured as 40 GBit/s ports

The NE1072T supports three different port profiles:

- default: 48x10G + 6x40G - forty-eight 10 GBit/s SFP+ ports and six 40 GBit/s QSFP+ ports

- seventy-two 10 GBit/s QSFP+ ports

- custom port mode

In 48x10G + 6x40G port mode, there are six 40 GBit/s ports that can be split into four 10 GBit/s ports. These are ethernet ports from 49 to 54.

To change the port mode to the 72x10G profile, use the following command:

```
Switch(config)# hardware profile portmode 72x10G
```

You can configure a custom port profile on the switch, changing an ethernet interface port mode from 40 GBit/s to 10 GBit/s or vice versa, and have different numbers of 10 GBit/s and 40 GBit/s ports than the default configuration. For example, you can configure 52 10 GBit/s ports and 5 40 Gbit/s ports.

Note: The maximum number of 40 GBit/s ports configurable on the switch is six.

To create a custom port profile, use the following command:

```
Switch(config)# hardware profile portmode custom ethernet <chassis number/
/port number>
```

Note: This command will configure the specified port to run in 10 GBit/s mode.

You can specify a single ethernet port or a range of ports. The range can be either continuos (for example, 20-25) or discrete (for example, 20, 22, 25, 30).

For example:

- Configure a single ethernet interface to run in 10 GBit/s mode:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet 1/50
```

- Configure a range of ethernet interfaces to run in 10 GBit/s mode:
  - continuos range:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet
1/49-52
```

  - discrete range:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet
1/49, ethernet 1/50
```

When a 40 GBit/s port is split into four 10 GBit/s breakout ports, the ethernet interface associated with that port will include four subports.

For example, before changing ethernet port 50 to 10 GBit/s port mode, the port will be displayed as follows:

```
Switch> display interface ethernet 1/50 brief

--------------------------------------------------------------------------------
Ethernet       VLAN   Type Mode   Status  Reason                   Speed   Port
Interface                                                                  Agg#
--------------------------------------------------------------------------------
Ethernet1/50   1      eth  access down    Administratively down    40000   --
```

After splitting the port into four 10 GBit/s breakout ports, the port will be displayed as follows:

```
Switch> display interface brief

--------------------------------------------------------------------------------
Ethernet       VLAN   Type Mode   Status  Reason                   Speed   Port
Interface                                                                  Agg#
--------------------------------------------------------------------------------
Ethernet1/1    1      eth  access down    Administratively down    40000   --
Ethernet1/50/1 1      eth  trunk  up      none                     10000   --
Ethernet1/50/2 1      eth  access down    Administratively down    10000   --
Ethernet1/50/3 1      eth  access down    Administratively down    10000   --
Ethernet1/50/4 1      eth  access down    Administratively down    10000   --
```

To configure 10 GBit/s breakout ports, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number/subport number>
```

For example:

```
Switch(config)# interface ethernet 1/50/2
```

When configuring a custom port profile on an ethernet interface, you can reset the interface's port mode to its default settings by using the following command:

```
Switch(config)# no hardware profile portmode custom ethernet <chassis number/
/port number/subport number>
```

To reset the switch port configuration to the default profile, use the following command:

```
Switch(config)# hardware profile portmode default
```

To view the current configured port profile, use the following command:

```
Switch# display hardware profile portmode

Current port mode:
Port Ethernet1/1 is set in 10G mode
Port Ethernet1/50 is set in 40G mode
```

NE1032T Port Configuration

The NE1032T has the following port capabilities:

● Twenty-four 1000/10G BASE-T RJ45 ports

● Eight Gigabit Ethernet (GbE) Small Form Pluggable Plus (SFP+) ports which also support legacy 1 GbE connections

The 10G BASE-T RJ45 ports, when used in 10 GbE mode, must use CAT6 copper cabling. When used in 1000 base T mode, the ports can be populated with CAT5E copper cabling.

SPF+ ports, when used in 10 GbE mode, can be populated with optical transceiver modules, or active or passive Direct-Attach Cables (DACs). When used in legacy 1 GbE mode, the ports can be populated with optical or copper transceiver modules.

By default, the NE1032T has the following interface (port) configuration:

● Ethernet interfaces 1-32 are configured as 10 GBit/s ports

NE1032 Port Configuration

The NE1032 has the following port capabilities:

● Thirty-two 10 Gigabit Ethernet (GbE) Small Form Pluggable Plus (SFP+) ports which also support legacy 1 GbE connections

SFP+ ports, when used in 10 GbE mode, can be populated with optical transceiver modules or active or passive Direct-Attach Cables (DACs). When used in legacy 1 GbE mode, the ports can be populated with optical or copper transceiver modules.

By default, the NE1032 has the following interface (port) configuration:

● ethernet interfaces 1-32 are configured as 10 GBit/s ports

NE2572 Port Configuration

The NE2572 has the following port capabilities:

● Forty-eight 25 Gigabit Ethernet (GbE) SFP28 ports supporting 10GbE and 25GbE connections.

● Six additional 100 Gigabit Ethernet (GbE) QSFP28 ports supporting 10GbE, 25GbE, 40GbE, 50GbE and 100GbE connections.

QSFP28 ports can be populated with Optical QSFP28/QSFP+ modules or Quad Direct Attach Cables (QDACs), including those that allow breakout to four 25GbE SFP28/SFP+ ports.

SFP28 port can be populated with Optical or Copper SFP28/SFP+ transceivers or SFP28/SFP+ Direct Attach Cables (DACs).

Note: The QSFP28 AOC Mellanox Transceiver is not accepted by the Arista DCS-7160-48YC6. The 100G link using the QSFP28 Mellanox AOC Transceiver does not work between the Arista DCS-7160-48YC6 and the NE2572.

The NE2572 supports the following port profiles:

- default: 38x25G + 6x100G - thirty-eight 25 GBit/s SFP+ ports + six 100 GBit/s QSFP+ ports
- 48x10G + 12x50 - forty-eight 10 GBit/s SFP+ ports + twelve 50 GBit/s QSFP+ ports
- 48x10G + 6x100 Gbig/s - forty-eight 10 GBit/s SFP+ ports + six 100 GBit/s QSFP+ ports
- 48x10G + 6x40 Gbig/s - forty-eight 10 GBit/s SFP+ ports + six 40 GBit/s QSFP+ ports
- 48x25G + 12x50 Gbig/s - forty-eight 25 GBit/s SFP+ ports + twelve 50 GBit/s QSFP+ ports
- 72x10G - seventy-two 10 GBit/s SFP+ ports
- 72x25G - seventy-two 25 GBit/s SFP+ ports
- custom port mode

In 48x25G + 6x100G port mode, the six 100 GBit/s ports that can be split into four 10GBit/s or 25GBit/s or two 50Gbit/s breakout ports are ethernet ports from 49 to 54 (included).

To change the port mode to the 48x10G + 6x40G profile, use the following command:

```
Switch(config)# hardware profile portmode 48x10G+6x40G
```

You can configure a custom port profile on the switch, changing an ethernet interface port mode from 100 GBit/s to 25GBit/s or vice versa, and have different numbers of 25 GBit/s and 100 GBit/s ports than the default configuration.

To create a custom port profile, use the following command:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet
<chassis number>/<port range>
```

Notes:

- This command will configure the specified port to run in 10 GBit/s mode.

- For the SFP28 ports (Ethernet ports 1-48), you must specify a four-port ethernet port range. For example, to configure a ethernet interfaces 1/41 through 1/44 to run in 10 GBit/s mode, enter:

```
Switch(config)# hardware profile portmode custom 4x10G ethernet
1/41-44
```

Trying to configure a single Ethernet port such as 1/40 or a discrete range of Ethernet ports, such as 1/40,41,42,43 will result in an error.

- The Quad ports (QSFP28 ports 49-55) do not have the four-port restriction.

When a 100 GBit/s port is split into four 10 GBit/s breakout ports, the ethernet interface associated with that port will include four subports.

For example, before changing ethernet port 1/50 to 10 GBit/s port mode, the port will be displayed as follows:

```
Switch> display interface ethernet 1/50 brief

--------------------------------------------------------------------------------
Ethernet        VLAN  Type Mode   Status  Reason                   Speed  Port
Interface                                                                 Agg#
--------------------------------------------------------------------------------
Ethernet1/50    1     eth  access down    Administratively down    40000  --
```

After spliting the port into four 10 GBit/s breakout ports, the port will be displayed as follows:

```
Switch> display interface brief

--------------------------------------------------------------------------------
Ethernet        VLAN  Type Mode   Status  Reason                   Speed  Port
Interface                                                                 Agg#
--------------------------------------------------------------------------------
Ethernet1/1     1     eth  access down    Administratively down    40000  --
Ethernet1/50/1  1     eth  trunk  up      none                     10000  --
Ethernet1/50/2  1     eth  access down    Administratively down    10000  --
Ethernet1/50/3  1     eth  access down    Administratively down    10000  --
Ethernet1/50/4  1     eth  access down    Administratively down    10000  --
```

To configure 10 GBit/s breakout ports, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number/subport number>
```

For example:

```
Switch(config)# interface ethernet 1/50/2
```

When configuring a custom port profile on an ethernet interface, you can reset the interface's port mode to its default settings by using the following command:

```
Switch(config)# no hardware profile portmode custom ethernet <chassis number/
/port number/subport number>
```

NE10032 Port Configuration

The NE10032 has the following port capabilities:

- Thirty-two 100 Gigabit Ethernet (GbE) QSFP28 ports supporting 10GbE, 25GbE, 50GbE and 100GbE connections.

The QSFP28 ports accept approved QSFP28 transceivers. The QSFP28 optical transceiver provides an MTP cable connector for connecting to external ports.

Note: The QSFP28 AOC Mellanox Transceiver is not accepted by the Arista DCS-7160-48YC6. The 100G link using the QSFP28 Mellanox AOC Transceiver does not work between the Arista DCS-7160-48YC6 and the NE10032.

By default, the NE10032 has the following interface (port) configuration

- ethernet interfaces 1-32 are configured as 100 GBit/s ports

The NE10032 supports the following port profiles:

- default: 32x100G - thirty-two 100 GBit/s QSFP28 ports

- custom port mode

- 128x10G - 128 10 Gbig/s QSFP28 ports

- 128x25G - 128 25 Gbig/s QSFP28 ports

- 32x40G - 32 40 Gbig/s QSFP28 ports

- 64x50G - 64 50 Gbig/s QSFP28 ports

You can configure a custom port profile on the switch, changing one or more Ethernet interfaces from 100 Gbit/s port mode to one of the following modes:.

Parameter	Mode
1x40G	1x40G mode
2x50G	2x50G mode
4x10G	4x10G mode
4x25G	4x25G mode

To create a custom port profile, use the following command:

```
Switch(config)# hardware profile portmode custom <parameter> ethernet <chassis
number>/<port number>
```

The following port speed combinations are supported on the NE10032:

- 10G/40G/100G

- 10G/25G/50G/100G

For example, the NE10032 can operate with 24 ports in 4x10G mode and 8 ports in 4x25G mode, but it cannot operate with 24 ports in 25G mode and 8 ports in 40G mode.

You can specify a single ethernet port or a range of ports. The range can be either continuos (for example, 20-25) or discrete (for example, 20, 22, 25, 30).

For example:

● Configure a single ethernet interface to run in 25 GBit/s mode:

```
Switch(config)# hardware profile portmode custom 4x25G ethernet 1/32
```

● Configure a range of ethernet interfaces to run in 10 GBit/s mode:
  o continuos range:

```
Switch(config)# hardware  profile portmode custom 4x10G ethernet
1/24-29
```

  o discrete range:

```
Switch(config)# hardware profile  portmode custom 4x10G ethernet
1/24, ethernet 1/29
```

When a 100 GBit/s port is split into breakout ports, the ethernet interface associated with that port will include the appropriate number of subports.

For example, before changing ethernet port 1 to 10 GBit/s port mode, the port will be displayed as follows:

```
Switch> display interface ethernet 1/50 brief

-------------------------------------------------------------------------------
Ethernet       VLAN  Type Mode   Status Reason                  Speed   Port
Interface                                                               Agg#
-------------------------------------------------------------------------------
Ethernet1/1    1     eth  access down   Administratively down  100000   --
```

After splitting the port into four 10 GBit/s breakout ports, the port will be displayed as follows:

```
Switch> display interface brief

-------------------------------------------------------------------------------
Ethernet       VLAN  Type Mode   Status Reason                  Speed   Port
Interface                                                               Agg#
-------------------------------------------------------------------------------
Ethernet1/1    1     eth  access down   Administratively down  100000   --
Ethernet1/1/1  1     eth  trunk  up     none                    25000   --
Ethernet1/1/2  1     eth  access down   Administratively down   25000   --
Ethernet1/1/3  1     eth  access down   Administratively down   25000   --
Ethernet1/1/4  1     eth  access down   Administratively down   25000   --
```

To configure 25 GBit/s breakout ports, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number/subport number>
```

For example:

```
Switch(config)# interface ethernet 1/1/2
```

When configuring a custom port profile on an ethernet interface, you can reset the interface's port mode to its default settings by using the following command:

```
Switch(config)# no hardware profile portmode custom ethernet <chassis number/
/port number/subport number>
```

To reset the switch port configuration to the default profile, use the following command:

```
Switch(config)# hardware profile portmode default
```

To view the current configured port profile, use the following command:

```
Switch# display hardware profile portmode

Current port mode:
Port Ethernet1/1 is set in 100G mode
Port Ethernet1/32 is set in 25G mode
```

Aggregation Overview

When using LAGs between two switches, as shown in Figure 3, you can create a virtual link between the switches that operates with combined throughput levels and depends on how many physical ports are included. The physical links in a LAG operate as a single logical connection.

Figure 3. Port LAG

LAGs are also useful for connecting the switch to third-party devices supporting link aggregation, such as Cisco routers and switches with EtherChannel technology (not ISL aggregation technology), or Sun's Quad Fast Ethernet Adapter. LAG technology is compatible with such devices if they are configured manually.

LAG traffic is statistically distributed among the ports in a LAG, based on a variety of configurable options.

Also, since each LAG is comprised of multiple physical links, the LAG is inherently fault tolerant. As long as one connection between the switches is available, the trunk remains active and statistical load balancing is maintained whenever a port in a LAG is lost or returned to service.

A LAG can be statically configured or dynamically aggregated through the use of a network protocol, like the Link Aggregation Control Protocol (LACP). For more details about LACP, see "Link Aggregation Control Protocol" on page 266.

When creating a LAG, you must assign it a number from 1 to 4096. The switch supports the following maximum number of LAGs, either statically configured or dynamically aggregated using LACP:

- G8272: maximum 72 LAGs
- G8296: maximum 96 LAGs
- G8332: maximum 104 LAGs
- NE1032T: maximum 32 LAGs
- NE1032: maximum 32 LAGs
- NE1072T: maximum 72 LAGs
- NE10032: maximum 126 LAGs
- NE2572: maximum 72 LAGs

Creating a LAG

To create a LAG, use the following command:

```
Switch(config)# interface port-aggregation <1-4096>
Switch(config-if)#
```

Note: Upon creating a LAG, you will enter Interface Configuration command mode, where you'll be able to configure the settings of the LAG.

To remove a LAG, use the following command:

```
Switch(config)# no interface port-aggregation <1-4096>
```

Static LAGs

You can create a static LAG that contains multiple physical links. Static LAGs do not require protocols like LACP to run on the switch to maintain the configured aggregation.

When you create a static LAG, the LAG members (switch ports) take on specific settings necessary for the correct operation of the aggregation.

Before configuring a static LAG, consider the following:

- Consider how existing VLANs will be affected by the addition of a LAG.

- Determine which switch ports (up to 32) are to become LAG members (the specific ports making up the LAG).

- Ensure that the chosen switch ports are in the up link state.

- Member ports must be all configured either as switch access ports or as switch trunk ports, and must have the same VLAN membership. For more details, see "Configuring a Switch Access Port" on page 225 and "Configuring a Switch Trunk Port" on page 226.

- LAG member ports must have the same Spanning Tree Protocol (STP) configuration.

- LAG member ports must have MAC address learning and flow control configuration.

- The same maximum transmission unit (MTU) size must be configured on all LAG member ports.

- The same bandwidth, speed, duplex, and auto-negotiation settings must be configured on all LAG member ports.

- Consider how the existing spanning tree will react to the new LAG configuration.

Static LAG Configuration Rules

When configuring a static LAG, you should consider the following configuration rules that determine how a LAG will react in any network topology:

- All links must originate from one logical source device and lead to one logical destination device. Usually, a LAG connects two physical devices together with multiple links. However, in some networks, a single logical device may include multiple physical devices, such as when using VLAGs (see Chapter 13, "Virtual Link Aggregation Groups"). In such cases, links in a LAG are allowed to connect to multiple physical devices because they act as one logical device.

- A physical switch port cannot be member of multiple LAGs. It can belong to only a single LAG.

- Aggregation from third-party devices must comply with Cisco EtherChannel technology.

- Each LAG inherits its port configuration (speed, flow control, native VLAN tagging) from the first member port. As additional ports are added to the LAG, their settings must be changed to match the LAG configuration.

- The maximum number of physical links in a LAG is 32. After reaching this limit, no more links can be added to the LAG.

- When a port becomes a member of a LAG, its MAC address and STP configuration are replaced with the ones of the LAG.

- When a port joins or leaves a LAG, the following parameters are unaffected: interface description, LLDP settings, LACP port priority, and link state.

Configuring a Static LAG

To add a physical port to a static LAG, use the following steps:

1. Configure the ethernet interface's port mode and VLAN membership.

   - as a switch access port:

     ```
     Switch(config)# interface ethernet <chassis number/port number>
     Switch(config-if)# bridge-port mode access
     Switch(config-if)# bridge-port access vlan <VLAN ID (1-4093)>
     Switch(config-if)# exit
     Switch(config)#
     ```

   - as a switch trunk port:

     ```
     Switch(config)# interface ethernet <chassis number/port number>
     Switch(config-if)# bridge-port mode trunk
     Switch(config-if)# bridge-port trunk allowed vlan <VLAN ID (1-4093)>
     Switch(config-if)# exit
     Switch(config)#
     ```

2. Create a LAG and configure the same port mode and VLAN membership as the previously configured ethernet interface.

   - access port mode:

     ```
     Switch(config)# interface port-aggregation <1-4096>
     Switch(config-if)# bridge-port mode access
     Switch(config-if)# bridge-port access vlan <VLAN ID (1-4093)>
     Switch(config-if)# exit
     Switch(config)#
     ```

   - trunk port mode:

     ```
     Switch(config)# interface port-aggregation <1-4096>
     Switch(config-if)# bridge-port mode trunk
     Switch(config-if)# bridge-port trunk allowed vlan <VLAN ID (1-4093)>
     Switch(config-if)# exit
     Switch(config)#
     ```

3. Add the interface to a LAG without an aggregation protocol:

   ```
   Switch(config)# interface ethernet <chassis number/port number>
   Switch(config-if)# aggregation-group <1-4096> mode on
   Switch(config-if)# exit
   Switch(config)#
   ```

4. Check the current LAG configuration:
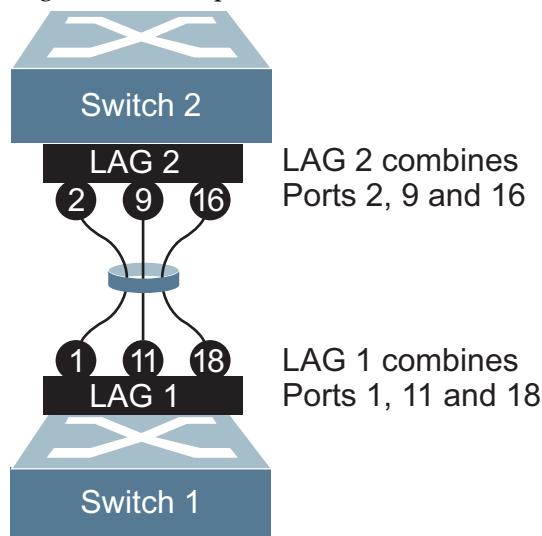
```
Switch(config)# display interface port-aggregation <1-4096>

Interface po1
  Hardware is AGGREGATE  Current HW addr: a897.dcde.2503
  Physical:(not set)  Logical:(not set)
  index 100001 metric 1 MTU 1500 Bandwidth 0 Kbit
  Port Mode is access
  <UP,BROADCAST,MULTICAST>
  VRF Binding: Not bound
  Members in this port-aggregation:
    Ethernet1/1, Ethernet1/2, Ethernet1/3, Ethernet1/4, Ethernet1/5
...
```

To remove an interface from a LAG, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# no aggregation-group
```

In the following example, three ports are aggregated between two switches.

Figure 4. LAG Configuration Example

Prior to configuring each switch in this example, you must connect to the appropriate switches as the administrator.

Note: For details about accessing and using any of the commands described in the following example, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

1. On Switch 1:

   a. Add ethernet interfaces 1, 11, and 18 to LAG 1:

   ```
   Switch(config)# interface ethernet 1/1, ethernet 1/11, ethernet 1/18
   Switch(config-if-range)# aggregation-group 1 mode on
   Switch(config-if-range)# exit
   ```

   Note: LAG 1 is automatically created when you assign ethernet interfaces 1, 11, and 18 to it.

   b. Check the configured LAG settings:

   ```
   Switch(config)# display interface port-aggregation 1
   ```

2. On Switch 2:

   a. Add ethernet interfaces 2, 9, and 16 to LAG 2:

   ```
   Switch(config)# interface ethernet 1/2, ethernet 1/9, ethernet 1/16
   Switch(config-if-range)# aggregation-group 2 mode on
   Switch(config-if-range)# exit
   ```

   Note: LAG 2 is automatically created when you assign ethernet interfaces 2, 9, and 16 to it.

   b. Check the configured LAG settings:

   ```
   Switch(config)# display interface port-aggregation 2
   ```

3. Connect the switch ports that will be members in the LAG.

   LAG 1 on Switch 1 is now connected to LAG 2 on Switch 2.

   Note: In this example, two switches are used. If a third-party device supporting link aggregation is used (such as Cisco routers and switches with EtherChannel technology or Sun's Quad Fast Ethernet Adapter), LAGs on the third-party device must be configured manually. Connection problems could arise when using automatic LAG negotiation on the third-party device.

Link Aggregation Control Protocol

The Link Aggregation Control Protocol (LACP) is an IEEE 802.3ad standard for grouping several physical ports into one logical port (known as a Link Aggregation Group - LAG) with any device that supports the standard. Please refer to IEEE 802.3ad-2002 for a full description of the standard.

The 802.3ad standard allows standard ethernet links to form a single Layer 2 link using Link Aggregation Control Protocol (LACP). Link aggregation is a method of grouping physical link segments of the same media type and speed in full duplex, and treating them as if they were part of a single, logical link segment. If a link in a LACP LAG fails, traffic is reassigned dynamically to the remaining link(s) of the dynamic LAG.

LACP ensures that a local LAG does not attempt to send traffic to a remote single interface. With LAG enabled on the switch, a local LAG can only transmit packets to a remote LAG also configured with LACP.

LACP allows the exchange of aggregation information between the two LAG participants in the form of Link Aggregation Control Protocol Data Units (LACPDUs). If the information is not agreed upon, the LAG will not form.

In comparison with a static LAG, LACP has the following advantages:

● If a physical link fails, LACP will detect the error and remove the link from the aggregate, ensuring packets are not lost due to the failure.

● Both of the participating devices can mutually confirm the LAG configuration. In the case of static link aggregation, configuration errors are often not detected as quickly.

Up to a maximum of 32 physical ports can be configured on a single LACP LAG.

By default, LACP is globally enabled on the switch and it cannot be disabled. However, it can be enabled or disabled for individual interfaces.

Configuring LACP

Before configuring ethernet ports as members of an LACP LAG, you must ensure they meet the same configuration requirements (such as switch port mode or VLAN membership) as specified for static LAGs. For more details, see .

To include a physical port in a LACP LAG, you must add the port to the desired LAG and specify its channel mode. You can configure a port with the following channel modes:

● on - configures the port as part of a static LAG without an aggregation protocol

● active - enables LACP on the interface and configures the port as an active member of the LACP negotiation that starts the negotiations by sending LACP packets to other ports

● passive - configures the port as a passive member of the LACP negotiation that only responds to received LACP packets from ports in the active state

Note: LACP is enabled on ports configured in the passive state only when an active LACP device is detected at the other end of the physical link.

A port in LACP active state will be able to form a LAG with another port that is either in the active or passive state. A port in LACP passive state will form a LAG with another port only if that port is in the active state.

Note: Two ports in LACP passive state will not form a LAG, because neither will start the negotiation process.

An interface configured as a static LAG member does not participate in LACP traffic. Thus, when a LACP active port tries to negotiate with that interface, it will not receive any reply and will become an individual link with the interface.

To configure an interface as a LACP LAG member, use the following command to add that interface to the desired LAG and specify its LACP state:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# aggregation-group <1-4096> mode {active|passive}
```

To remove an interface from a LACP LAG, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# no aggregation-group
```

System Priority

A switch running LACP has an assigned system priority. LACP uses the configured system priority with the switch's MAC address to generate a LACP system ID.

The system priority is also used during negotiations with other devices. The switch with the lowest configured system priority determines which physical ports participate in the LACP aggregation. If both switches have the same system priority, the LACP system ID is used to determine the controlling device.

To configure the switch LACP system priority, use the following command:

```
Switch(config)# lacp system-priority <1-65535>
```

By default, the system priority has a value of 32768. To reset the switch's system priority to the default value, use the following command:

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

Port Priority

Each physical interface that is a member of an LACP LAG has an assigned port priority. LACP uses the configured port priority with the port number to form the LACP port ID.

To configure the port priority of an interface, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# lacp port-priority <1-65535>
```

By default, the port priority of an interface is 32768. To reset the port priority to its default value, use the following command:

```
Switch(config-if)# no lacp port-priority
```

LACP Timeout

An interface that is a member of a LACP LAG will send LACPDUs at a regular interval. After not receiving any response to three consecutive LACPDUs, the link is timed out and the state of the interface transitions to individual. Interfaces in the individual or suspended state will function as normal physical ports.

You can configure a short or a long timeout interval. A short timeout interval is equal to three seconds, meaning that LACPDUs are sent every second. A long timeout interval has a value of 90 seconds, with LACPDUs being sent every 30 seconds.

To configure the timeout interval of an interface, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# lacp timeout {short|long}
```

By default, LACP uses a long timeout interval. To reset an interface LACP timeout interval to this value, use the following command:

```
Switch(config-if)# no lacp timeout
```

LACP Individual

A port with LACP enabled (active or passive) that cannot join a LACP LAG has its state changed to individual and is able to form an individual link. This can be useful when a LACP enabled port is connected to port with LACP disabled or when two LACP enabled ports are both in the passive state. In such scenarios, LACP individual helps avoid the loss of traffic.

By default, LACP individual is globally enabled on the switch and cannot be disabled. However, it can enabled or disabled individually on each interface.

Note: LACP individual can only be enabled or disabled on LACP enabled port aggregations. Static LAGs member are unaffected.

Although, this behavior is helpful in preventing traffic loss, it can cause problems by permitting loops to be created by mismatching port configurations.

Ports with LACP enabled can be configured to transition to the suspended state instead of individual. Ports in the suspended state are not able to process traffic.

To configure a LACP enabled interface to transition to the suspended state after it times out, use the following commands:

```
Switch(config)# interface port-aggregation <1-4096>
Switch(config-if)# shutdown
Switch(config-if)# lacp suspend-individual
Switch(config-if)# no shutdown
```

Note: The LAG must be disabled (shut down) to configure LACP individual. Once the desired changes are made, the LAG can be re-enabled.

To configure a LACP enabled interface to transition to individual state after it times out, use the following commands:

```
Switch(config-if)# shutdown
Switch(config-if)# no lacp suspend-individual
Switch(config-if)# no shutdown
```

Note: LACP individual can only be configured for LAGs. When the LACP individual configuration is changed, it will apply the modification to all member ports.

LACP Minimum Links

The LACP minimum links feature is used to configure the minimum number of member ports that must be in the link-up state and bundled in the LACP portchannel. You can use this feature to prevent low-bandwidth LACP port channels from becoming active. This feature also causes LACP port channels to become inactive if they have too few active member ports to supply your required minimum bandwidth.

When the number of all link-up and aggregated member ports is less than the LACP minimum links number, those member ports will go into standby state. Interfaces in standby state do not forward data traffic but continue to run the LACP protocol.

To set the number of LACP minimum links, enter:

```
Switch(config-if)# lacp min-links <min-links>
```

where *min-links* is an integer from 1-32 with a default value of 1.

To reset this feature, enter:

```
Switch(config-if)# no lacp min-links
```

To view the number of minimum links, enter:

```
Switch(config-if)# display lacp min-links [interface port-aggregation <range>]
```

where *range* is the port aggregation range; one or more integers from 1-4096

Notes:

● The LACP min-links number must be equal to or less than the maximum number of physical links in one LAG.

● The LACP minimum links feature can be configured on a LAG, but it will only be active when the LAG is an LACP portchannel.

LACP Configuration Example

Use the following procedure to configure LACP for ethernet ports 1, 2, and 3 to participate in link aggregation.

1. Ensure that ethernet ports 1, 2, and 3 have the same port mode settings and are members of the same VLAN (in this example, VLAN 10 is used). Configure the ethernet ports as one of the following:

   - switch access ports:

   ```
   Switch(config)# interface ethernet 1/1-3
   Switch(config-if-range)# bridge-port mode access
   Switch(config-if-range)# bridge-port access vlan 10
   Switch(config-if-range)# exit
   Switch(config)#
   ```

   - switch trunk ports:

   ```
   Switch(config)# interface ethernet 1/1-3
   Switch(config-if-range)# bridge-port mode trunk
   Switch(config-if-range)# bridge-port trunk native vlan 10
   Switch(config-if-range)# bridge-port trunk allowed vlan 10
   Switch(config-if-range)# exit
   Switch(config)#
   ```

2. Create a LAG (in this example, LAG 1 is used) and ensure that its port mode settings and VLAN membership are the same as the ones configured on ethernet ports 1, 2, and 3. Configure the LAG with one of the following port modes:

   - access port mode:

   ```
   Switch(config)# interface port-aggregation 1
   Switch(config-if)# bridge-port mode access
   Switch(config-if)# bridge-port access vlan 10
   Switch(config-if)# exit
   Switch(config)#
   ```

   - trunk port mode:

   ```
   Switch(config)# interface port-aggregation 1
   Switch(config-if)# bridge-port mode trunk
   Switch(config-if)# bridge-port trunk native vlan 10
   Switch(config-if)# bridge-port trunk allowed vlan 10
   Switch(config-if)# exit
   Switch(config)#
   ```

3. (Optional) Configure the port priority and timeout of the LAG members:

   ```
   Switch(config)# interface ethernet 1/1-3
   Switch(config-if)# lacp port-priority <1-65535>
   Switch(config-if)# lacp timeout {short|long}
   Switch(config-if)# exit
   Switch(config)#
   ```

4. (Optional) Configure the LACP system priority:

```
Switch(config)# lacp system-priority <1-65535>
```

5. Configure the ports as members of LAG 1 and set them in LACP active mode:

```
Switch(config)# interface ethernet 1/1-3
Switch(config-if-range)# aggregation-group 1 mode active
```

LAG Hashing

Traffic in a LAG is statistically distributed among member ports using a hash process where various address and attribute bits from each transmitted frame are recombined to specify the particular LAG physical port the frame will use.

LAG hashing can be only configured globally on the switch, thus affecting all existing LAGs.

Note: LAG hashing is enabled by default on the switch and cannot be disabled.

The switch uses RTAG7 as its hashing method. It offers different information options from which to choose the parameters used to calculate the hash value. To achieve the most even traffic distribution, configure options that exhibit a wide range of values for your specific network. Avoid hashing on information that is not usually present in the expected traffic or which does not vary.

The switch can use the following packet information options to calculate the hash value for traffic balancing:

● Packet ingress physical interface

● Layer 2 information:
  ○ destination MAC address
  ○ source MAC address
  ○ source and destination MAC addresses

● Layer 3 information:
  ○ destination IP address
  ○ source IP address
  ○ source and destination IP addresses

● Layer 4 information:
  ○ destination port number
  ○ source port number
  ○ source and destination port numbers

Notes:

● All packet information options are mutually exclusive when calculating the hash value. Only one of these options can be selected.

● The packet ingress physical interface can also be used in combination with of the above options to calculate the hash value.

When the hash value is calculated using Layer 3 or Layer 4 packet information, the associated Layer 2 is also applied to non-IP protocol traffic. Table 21 presents the applied hashing criteria associated with every hashing configuration:

Table 21. *Hashing Criteria*

Hash configuration	Layer 2 Criteria	Layer 3 Criteria	Layer 3 Criteria
destination MAC	destination MAC	destination MAC	destination MAC
source MAC	source MAC	source MAC	source MAC
source and destination MAC	source and destination MAC	source and destination MAC	source and destination MAC
destination IP	destination MAC	destination IP	destination IP
source IP	source MAC	source IP	source IP
source and destination IP	source and destination MAC	source and destination IP	source and destination IP
destination port	destination MAC	destination IP	destination IP, destination port
source port	source MAC	source IP	source IP, source port
source and destination port	source and destination MAC	source and destination IP	source and destination IP, source and destination port

By default, the hash value is calculated using the source and destination IP addresses. This means that for Layer 3 or Layer 4 packets the hash value is calculated using the source and destination IP addresses, and for Layer 2 packets the hash value is calculated using the source and destination MAC addresses.

To check the current LAG hash configuration, use the following command:

```
Switch> display port-aggregation load-balance

Port-aggregation Load-Balancing Configuration:
System: source-dest-ip

Port-aggregation Load-Balancing Addresses Used Per-Protocol:
Non-IP: source-dest-mac
IP: source-dest-ip
```

Chapter 11: Ports and Link Aggregation

LAG Hashing Configuration

To configure what packet information options are used to calculate the hash value, use the following command:

```
Switch(config)# port-aggregation load-balance ethernet <hash method>

Switch(config)# port-aggregation load-balance ethernet ?
  destination-ip    Load distribution on the destination IP address
  destination-mac   Load distribution on the destination MAC address
  destination-port  Load distribution on the destination TCP/UDP port
  source-dest-ip    Load distribution on the source and destination IP
                    address
  source-dest-mac   Load distribution on the source and destination MAC
                    address
  source-dest-port  Load distribution on the source and destination
                    TCP/UDP port
  source-interface  Load distribution on the source ethernet interface
  source-ip         Load distribution on the source IP address
  source-mac        Load distribution on the source MAC address
  source-port       Load distribution on the source TCP/UDP port
```

For more details about this command, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

To reset the hashing method to its default settings, use the following command:

```
Switch(config)# no port-aggregation load-balance ethernet
```

To view the LAG hash value, use the following command:

```
Switch> display port-aggregation load-balance forwarding-path interface
port-aggregation <1-4096> <hash parameters>

Switch> display port-aggregation load-balance forwarding-path interface
port-aggregation 1 ?
  src-interface  Optional source interface (physical switch port only)
  dst-mac        Destination MAC Address
  src-mac        Source MAC Address
  dst-ip         Destination IP Address
  src-ip         Source IP Address
  dst-ipv6       Destination IPV6 Address
  src-ipv6       Source IPV6 Address
  l4-dst-port    Destination Port
  l4-src-port    Source Port
```

For more details about this command, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

The *hash parameters* include the following:

- Packet ingress physical interface
- source and destination MAC addresses
- source and destination IPv4 or IPv6 addresses
- source and destination port numbers

The hash parameters used to calculate the hashing value are decided by the LAG hashing configuration and the traffic type. If there is IP or port information input, then the traffic type is Layer 3 or Layer 4. If there is no such information input, then the traffic type is Layer 2. Any unspecified hash parameters are substituted with 0.

For example, if the LAG hashing configuration is source and destination IP addresses, and IP information is contained in the input, then the Layer 2 and Layer 4 information is ignored when calculating the hash value.

```
Switch> display port-aggregation load-balance forwarding-path interface
port-aggregation <1-4096> dst-ip 10.158.70.96 src-ip 10.169.2.36 src-mac
78AA.F560.EF0F l4-src-port 14255 l4-dst-port 16874

Missing params will be substituted by 0's.
Load-balance Algorithm on switch: source-dest-ip
Outgoing port id: Ethernet 1/1

Param(s) used to calculate load balance:
destination-ip: 10.158.70.96
source-ip: 10.169.2.36
```

If the LAG hashing configuration is source and destination IP addresses, and no IP information is contained in the input, then the default IPv4 address (0.0.0.0) is used when calculating the hash value:

```
Switch> display port-aggregation load-balance forwarding-path interface
port-aggregation <1-4096> src-mac 78AA.F560.EF0F l4-src-port 14255
l4-dst-port 16874

Missing params will be substituted by 0's.
Load-balance Algorithm on switch: source-dest-ip
Outgoing port id: Ethernet 1/1

Param(s) used to calculate load balance:
destination-ip: 0.0.0.0
source-ip: 0.0.0.0
```

Chapter 12. Spanning Tree Protocol

When multiple paths exist between two points on a network, Spanning Tree Protocol (STP) or one of its enhanced variants ensure that the switch uses the most efficient network path.

Lenovo Cloud Network Operating System supports Multiple Spanning Tree Protocol (MSTP) and Rapid Per VLAN Spanning Tree Plus (Rapid PVST+).

IEEE 802.1D (2004), Rapid Spanning Tree Protocol (RSTP), allows devices to detect and eliminate logical loops in a bridged or switched network. When multiple paths exist, STP configures the network so only the most efficient path is used. If that path fails, STP automatically configures the best alternative active path on the network to sustain network operations. RSTP is an enhanced version of IEEE 802.1D (1998) STP, providing more rapid convergence of the Spanning Tree network path states on STG 1.

Rapid Per VLAN Spanning Tree Plus (Rapid PVST+) is based on RSTP to provide rapid Spanning Tree convergence, but supports multiple spanning tree instances, allowing one Spanning Tree Group (STG) per VLAN.

Based on IEEE 802.1Q (2005), the Multiple Spanning Tree Protocol (MSTP) provides both rapid convergence and load balancing in a VLAN environment. MSTP allows multiple STGs, with multiple VLANs in each.

This chapter covers the following topics:

STP Overview

The Spanning Tree Protocol (STP) builds a logical loop-free network topology. The primary function of STP is to prevent bridge loops from being created. Bridge loops occur when there are more than a single Layer 2 path between two endpoints. For example, two switch ports are connected to each other, or two switches have multiple connections between them. While having multiple redundant connection improves network availability, these connections can create bridge loops, that generate broadcast storms and reduce network performance.

STP creates a spanning tree in the network of Layer 2 connected bridges. It disables the links that are not part of the spanning tree, leaving a single active path between any two connected endpoints.

In Rapid PVST+, the switch can be configured with up to a maximum of 500 VLANs. The operational behavior of the switch depends on the maximum number of interfaces on the VLANs.

By default, STP is globally enabled on the switch, running Rapid PVST+. To disable it, use the following command:

```
Switch(config)# spanning-tree mode disable
```

To enable STP or change the STP mode between Rapid PVST+ or MSTP, use the following command:

```
Switch(config)# spanning-tree mode {mst|rapid-pvst}
```

To reset the STP mode to the default value, use the following command:

```
Switch(config)# no spanning-tree mode
```

To display STP information, use the following command:

```
Switch> display spanning-tree
```

To enable or disable STP on a switch ethernet port or Link Aggregation Group (LAG), use the following commands:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree {enable|disable}
```

```
Switch(config)# interface port-aggregation <1-4096>
Switch(config-if)# spanning-tree {enable|disable}
```

To display STP information for a specified switch ethernet port or Link Aggregation Group (LAG), use the following command:

```
Switch> display spanning-tree interface {ethernet <chassis number/port number>|
|port-aggregation <1-4096>}
```

Bridge Protocol Data Units

To create a Spanning Tree, the switch generates a configuration Bridge Protocol Data Unit (BPDU), which it then forwards out of its ports. All switches in the Layer 2 network participating in the Spanning Tree gather information about other switches in the network through an exchange of BPDUs.

A bridge sends BPDU packets at a configurable regular interval (2 seconds by default). The BPDU is used to establish a path, much like a hello packet in IP routing. BPDUs contain information about the transmitting bridge and its ports, including bridge MAC addresses, bridge priority, port priority and path cost.

The generic action of a switch on receiving a BPDU is to compare the received BPDU to its own BPDU that it will transmit. If the priority of the received BPDU is better than its own priority, it will replace its BPDU with the received BPDU. Then, the switch adds its own bridge ID number and increments the path cost of the BPDU. The switch uses this information to block any necessary ports.

Note: If STP is globally disabled, BPDUs from external devices will transit the switch transparently. If STP is globally enabled, for ports where STP is turned off, inbound BPDUs will instead be discarded.

Determining the Path for Forwarding BPDUs

When determining which port to use for forwarding and which port to block, the switch uses information in the BPDU, including each bridge ID. A technique based on the "lowest root cost" is then computed to determine the most efficient path for forwarding.

BPDU Guard

The BPDU Guard will put a switch interface into the shut down (error disabled) state when a BPDU is received on that interface. The administrator can manually put the interface in the up state or, if Error Disable Recovery is enabled, it will automatically bring the interface up (for more information, see "Error Disable Recovery" on page 283).

Note: BPDU Guard will function even when STP is disabled. However, it will not work when BPDU Filter is also enabled.

By default, BPDU Guard is disabled. You can enable it on each of the switch interfaces by using the following command:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree bpduguard enable
```

To disable BDPU Guard on an interface, use the following command:

```
Switch(config-if)# spanning-tree bpduguard disable
```

To reset the BPDU Guard to its default configuration, use the following command:

```
Switch(config-if)# no spanning-tree bpduguard
```

BPDU Filter

The BPDU Filter prevents the transmission or reception of BPDUs on a switch interface.

Notes:

● When the BPDU Filter is enabled on an interface, the BPDU Guard will not function, because all BPDUs will be dropped by the switch.

● BPDU Filter will function even when STP is disabled.

By default, BPDU Filter is disabled. You can enable it on each of the switch interfaces by using the following command:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree bpdufilter enable
```

To disable BDPU Filter on an interface, use the following command:

```
Switch(config-if)# spanning-tree bpdufilter disable
```

To reset the BPDU Filter to its default configuration, use the following command:

```
Switch(config-if)# no spanning-tree bpdufilter
```

Root Guard

The root guard feature provides a way to enforce the root bridge placement in the network. It keeps a new device from becoming root and thereby forcing STP re-convergence. If a root-guard enabled port detects a root device, that port will be placed in a blocked state and then automatically recovered.

Note: This differs from BPDU Guard, where the port is blocked and recovered only if Error Disabled Recovery is also enabled, or the port is manually brought back up by an administrator.

To configure the root guard at the port level, use the following commands:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree guard root
```

Note: When you enable root guard on a switch interface, it will disable loop guard, if configured to run on that interface.

To disable the root guard, use the following command:

```
Switch(config-if)# no spanning-tree guard
```

By default, root guard is disabled.

Loop Guard

In general, STP resolves redundant network topologies into loop-free topologies. The loop guard feature performs additional checking to detect loops that might not be found using Spanning Tree. When a port is blocked by STP and it erroneously transitions to the forwarding state, a loop is created in the topology.

Loop guard checks if any BDPUs are received on a non-designated port. If the port doesn't receive any BDPUs, loop guard prevents the port from transitioning into the forwarding state and sends the port into the loop-inconsistent state. In this state, the port is blocked and no traffic is not forwarded. If loop guard is disabled, when the port does not receive BPDUs, it will transition into the forwarding state, thus creating a bridge loop.

To configure the loop guard at the port level, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree guard loop
```

Note: When you enable loop guard on a switch interface, it will disable root guard, if configured to run on that interface.

To disable the loop guard, use the following command:

```
Switch(config-if)# no spanning-tree guard
```

By default, loop guard is disabled.

Port Priority

The port priority helps determine which bridge port becomes the root port or the designated port. The case for the root port is when two switches are connected using a minimum of two links with the same path-cost. The case for the designated port is in a network topology that has multiple bridge ports with the same path-cost connected to a single segment, the port with the lowest port priority becomes the designated port for the segment.

Use the following command to configure the port priority:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree port-priority <0-224>
```

Note: Port priority must be configured in increments of 32 (such as 0, 64, 96, 128 and so on). If the specified port priority value is not evenly divisible by 32, the value will be automatically rejected.

The default port priority is 128. To reset it to its default value, use the following command:

```
Switch(config-if)# no spanning-tree port-priority
```

Port Path Cost

The port path cost assigns lower values to high-bandwidth ports, such as 10 Gigabit Ethernet, to encourage their use. The objective is to use the fastest links so that the route with the lowest cost is chosen. A value of auto (the default) indicates that the default cost will be computed for an auto-negotiated link or LAG speed.

To globally configure the calculation method for the default path cost, use the following command:

```
Switch(config)# spanning-tree pathcost method <method type>
```

where *method type* is one of the following:

● long: 32 bit based values

● short: 16 bit based values

Note: When using MSTP mode, the switch uses only the long method for calculating path cost.

By default, the switch uses 16 bit based values (short) when calculating the default path cost.

To reset the calculation method to the default value, use the following command:

```
Switch(config)# no spanning-tree pathcost method
```

You can also configure the port path cost differently on each interface by using the following command:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree cost {<1-200000000>|auto}
```

By default, the port path cost is automatically calculated based on the ethernet port speed. To reset the port path cost to its default setting, use the following command:

```
Switch(config-if)# no spanning-tree cost
```

Error Disable Recovery

When BPDU Guard is enabled on a switch interface, it will put that interface in the error disabled (shutdown) state when a BPDU is received.

To bring the interface back up, the administrator can manually put it in the up state or he can configure Error Disable Recovery to automatically transition interface from the error disabled state to the up state after a certain time interval.

By default, Error Disable Recovery is disabled on the switch. To enable or disable it, use the following command:

```
Switch(config)# [no] errdisable recovery cause bpduguard
```

By default, Error Disable Recovery will bring the port back up after 300 seconds (five minutes) since it was detected as being in the error disabled state.

To configure the recovery time interval (in seconds), use the following command:

```
Switch(config)# errdisable recovery interval <30-65535>
```

To reset the recovery time interval to its default value, use the following command:

```
Switch(config)# no errdisable recovery interval
```

Port Type and Link Type

The following port and link types are supported by STP.

Edge Port

A port that does not connect to a bridge is called an *edge port*. Since edge ports are assumed to be connected to non-STP devices (such as directly to hosts or servers), they are placed in the forwarding state as soon as the link is up.

Edge ports send BPDUs to connected STP devices like normal STP ports, but do not receive BPDUs. If a port with edge enabled does receive a BPDU, it immediately begins working as a normal (non-edge) port, and participates fully in Spanning Tree.

To configure a port as an edge port or as non-edge port, use the following command:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# [no] spanning-tree port type edge
```

Link Type

The link type determines how the port behaves in regard to Multiple Spanning Tree. Use the following command to define the link type for the port:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree link-type <type>
```

where *type* corresponds to the duplex mode of the port, as follows:

- point-to-point

  A full-duplex direct link to another device (point-to-point).

- shared

  A half-duplex link that is shared by multiple network devices.

- auto

  The switch automatically configures the link type.

Note: Any STP port in full-duplex mode can be manually configured as a shared port when connected to a non-STP-aware shared device (such as a typical Layer 2 switch) used to interconnect multiple STP-aware devices.

By default, the selected duplex mode is auto.

To reset the duplex mode to its default value, use the following command:

```
Switch(config-if)# no spanning-tree link-type
```

Rapid Per VLAN Spanning Tree Plus

Using STP, network devices detect and eliminate logical loops in a bridged or switched network. When multiple paths exist, the Spanning Tree Protocol (STP) configures the network so that a switch uses only the most efficient path. If that path fails, STP automatically sets up another active path on the network to sustain network operations.

Rapid Per VLAN Spanning Tree Plus (Rapid PVST+) creates a spanning tree topology that is implemented per VLAN. A single instance of STP runs on each configured VLAN where STP was not manually disabled. Each Rapid PVST+ instance on a VLAN has a single root switch. Using Rapid PVST+, spanning trees converge much faster than spanning trees using 802.1D STP. Spanning tree re-configurations can happen in less than a second.

Note: In Rapid PVST+, the switch can be configured with up to a maximum of 500 VLANs. The operational behavior of the switch depends on the maximum number of interfaces on the VLANs.

To configure the switch to use Rapid PVST+, use the following command:

```
Switch(config)# spanning-tree mode rapid-pvst
```

Note: By default, STP is enabled on the switch and runs using Rapid PVST+.

To disable STP on the switch, use the following command:

```
Switch(config)# spanning-tree mode disable
```

Rapid PVST+ is configured for each active VLAN, allowing different spanning tree topologies to be created using different parameters.

To enable or disable Rapid PVST+ on a VLAN, use the following command:

```
Switch(config)# [no] spanning-tree vlan <VLAN ID (1-4093)>
```

Notes:

- Rapid PVST+ cannot be disabled on VLAN 1.

- Rapid PVST+ can be configured on a single VLAN or on a range of VLANs, either in a continuous format (1-10), in a discrete format (3, 5, 90), or a as a combination of the two (1-5, 7, 9-12, 45).

- VLAN creation is not allowed if the new VLAN would exceed the limit of STGs.

- If the switch is running in Multiple Spanning Tree Protocol (MSTP) mode and the number of VLANs exceeds the maximum supported number, STP mode cannot be changed to Rapid PVST+.

To display STP information about a VLAN, use the following command:

```
Switch> display spanning-tree vlan <VLAN ID (1-4093)>
```

Rapid PVST+ Parameters

You can configure the following parameters for Rapid PVST+ per VLAN:

- bridge priority
- port priority
- port path cost
- forward delay
- hello timer
- maximum age interval

Bridge Priority

The bridge priority parameter controls which bridge on the network is the STG root bridge. To make one switch become the root bridge, configure the bridge priority lower than all other switches and bridges on your network. The lower the value, the higher the bridge priority. Use the following command to configure the bridge priority per VLAN:

```
Switch(config)# spanning-tree vlan <VLAN ID (1-4093)> priority <0-61440>
```

Note: Priority must be configured in increments of 4096 (such as 0, 4096, 8192, or 12288).

By default, the bridge priority is set to 32768. To reset the bridge priority to its default value, use the following command:

```
Switch(config)# no spanning-tree vlan <VLAN ID (1-4093)> priority
```

Port Priority

The port priority helps determine which bridge port becomes the root port or the designated port. The case for the root port is when two switches are connected using a minimum of two links with the same path-cost. The case for the designated port is in a network topology that has multiple bridge ports with the same path-cost connected to a single segment, the port with the lowest port priority becomes the designated port for the segment.

Use the following command to configure the port priority per VLAN:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree vlan <VLAN ID (1-4093)> port-priority <0-224>
```

Note: Port priority must be configured in increments of 32 (such as 0, 64, 96, or 128). If the specified port priority value is not evenly divisible by 32, the value will be automatically rejected.

The default port priority is 128. To reset it to its default value, use the following command:

```
Switch(config-if)# no spanning-tree vlan <VLAN ID (1-4093)> port-priority
```

Port Path Cost

The port path cost assigns lower values to high-bandwidth ports, such as 10 Gigabit Ethernet, to encourage their use. The objective is to use the fastest links so that the route with the lowest cost is chosen. A value of auto (the default) indicates that the default cost will be computed for an auto-negotiated link or LAG speed.

To configure the port path cost differently on each interface, use the following command per VLAN:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree vlan <VLAN ID (1-4093)> cost {<1-200,000,000>|
auto}
```

By default, the port path cost is automatically calculated based on the ethernet port speed. To reset the port path cost to its default setting, use the following command:

```
Switch(config-if)# no spanning-tree vlan <VLAN ID (1-4093)> cost
```

Forward Delay

Forward delay specifies the amount of time that a bridge port has to wait before it changes from the discarding and learning states to the forwarding state.

Use the following command to configure the forward delay (in seconds) per VLAN:

```
Switch(config)# spanning-tree vlan <VLAN ID (1-4093)> forward-time <4-30>
```

By default, the forward delay is 15 seconds. To reset it to its default value, use the following command:

```
Switch(config)# no spanning-tree vlan <VLAN ID (1-4093)> forward-time
```

Hello Timer

The Hello timer specifies how often the bridge transmits a configuration BPDU. Any bridge that is not the root bridge uses the root bridge hello timer value.

Use the following command to configure the hello timer (in seconds) per VLAN:

```
Switch(config)# spanning-tree vlan <VLAN ID (1-4093)> hello-time <1-10>
```

By default, the hello timer is 2 seconds. To reset it to its default value, use the following command:

```
Switch(config)# no spanning-tree vlan <VLAN ID (1-4093)> hello-time
```

Maximum Age Interval

The maximum age interval specifies the maximum time the bridge waits without receiving a configuration BPDU before it reconfigures the spanning tree topology.

Use the following command to configure the maximum age interval (in seconds) per VLAN:

```
Switch(config)# spanning-tree vlan <VLAN ID (1-4093)> max-age <6-40>
```

By default, the maximum age interval is 20 seconds. To reset it to its default value, use the following command:

```
Switch(config)# no spanning-tree vlan <VLAN ID (1-4093)> max-age
```

Rapid PVST+ Configuration

The following is an example of configuring Rapid PVST+ on the switch. In this example there are two different STP configurations for two different VLAN ranges.

1. Configure port and VLAN membership on the switch.

2. Configure the switch to use Rapid PVST+:

   ```
   Switch(config)# spanning-tree mode rapid-pvst
   ```

3. Configure the Rapid PVST+ parameters based on VLAN:

   ```
   Switch(config)# spanning-tree vlan 1-10 priority 4096
   Switch(config)# spanning-tree vlan 1-10 forward-time 5
   Switch(config)# spanning-tree vlan 1-10 hello-time 3
   Switch(config)# spanning-tree vlan 1-10 max-age 10

   Switch(config)# spanning-tree vlan 11-20 priority 8192
   Switch(config)# spanning-tree vlan 11-20 forward-time 10
   Switch(config)# spanning-tree vlan 11-20 hello-time 5
   Switch(config)# spanning-tree vlan 11-20 max-age 50
   ```

4. Configure the port parameters:

   ```
   Switch(config)# interface ethernet 1/1-10
   Switch(config-if)# spanning-tree vlan 1-10 port-priority 32
   Switch(config-if)# spanning-tree vlan 1-10 cost 2300
   Switch(config-if)# exit
   Switch(config)#

   Switch(config)# interface ethernet 1/11-20
   Switch(config-if)# spanning-tree vlan 11-20 port-priority 96
   Switch(config-if)# spanning-tree vlan 11-20 cost 15000
   Switch(config-if)# exit
   ```

5. Verify the Rapid PVST+ configuration:

   ```
   Switch# display spanning-tree vlan 1-20
   ```

Multiple Spanning Tree Protocol

Multiple Spanning Tree Protocol (MSTP) extends Rapid Spanning Tree Protocol (RSTP), allowing multiple Spanning Tree Groups (STGs) that may each include multiple VLANs. MSTP was originally defined in IEEE 802.1s (2002) and was later included in IEEE 802.1Q (2005).

In MSTP mode, the switch supports up to 64 instances of Spanning Tree, corresponding to STGs 1-64, with each STG acting as an independent, simultaneous instance of RSTP.

MSTP allows frames assigned to different VLANs to follow separate paths, with each path based on an independent Spanning Tree instance. This approach provides multiple forwarding paths for data traffic, thereby enabling load-balancing, and reducing the number of Spanning Tree instances required to support a large number of VLANs.

Due to Spanning Tree's sequence of discarding, learning, and forwarding, lengthy delays may occur while paths are being resolved. Ports defined as *edge* ports ("Port Type and Link Type" on page 284) bypass the Discarding and Learning states, and enter directly into the Forwarding state.

Note: By default, STP is enabled on the switch and runs using Rapid PVST+.

MSTP can be enabled by specifying the STP mode:

```
Switch(config)# spanning-tree mode mst
```

To display MSTP information, use the following command:

```
Switch> display spanning-tree mst <0-64>
```

Common Internal Spanning Tree

The Common Internal Spanning Tree (CIST) or MST0 provides a common form of Spanning Tree Protocol, with one Spanning-Tree instance that can be used throughout the MSTP region. CIST allows the switch to interoperate with legacy equipment, including devices that run IEEE 802.1D (1998) STP.

CIST allows the MSTP region to act as a virtual bridge to other bridges outside of the region, and provides a single Spanning-Tree instance to interact with them.

Port States

The port state controls the forwarding and learning processes of Spanning Tree. In MSTP, the port state has been consolidated to the following: discarding, learning, and forwarding.

Due to the sequence involved in these STP states, considerable delays may occur while paths are being resolved. To mitigate delays, ports defined as *edge* ports ("Port Type and Link Type" on page 284) may bypass the discarding and learning states and enter directly into the forwarding state.

MST Region

A group of interconnected bridges that share the same attributes is called an MST region. Each bridge within the region must share the following attributes:

● Region name (up to a maximum of 32 alphanumeric characters)

● Revision number

● VLAN-to STG mapping scheme

MSTP provides rapid re-configuration, scalability, and control due to the support of regions and multiple Spanning Tree instances support within each region.

To enter the MST Region Configuration mode, use the following command:

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)#
```

To assign a name to the MST region, use the following command:

```
Switch(config-mst)# name <region name>
```

To assign a revision number to the MST region, use the following command:

```
Switch(config-mst)# revision <0-65535>
```

By default, the MST region revision number is 0.

The administrator may manually assign VLANs to specific MSTIs (Multiple Spanning Tree instances).

1. If no VLANs exist (other than default VLAN 1), see "Creating a VLAN" on page 220 for information about creating VLANs and assigning ports to them.

2. Assign the VLAN to a MSTI using the following command:

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# instance <0-64> vlan <VLAN ID (1-4093)>
```

MSTP Parameters

You can configure the following parameters for MSTP per MST instance:

● hop count

● forward delay

● hello timer

● maximum age interval

● bridge priority

● port priority

● port path cost

Hop Count

To compute the STP topology inside the MST region, the protocol uses, along with the path cost to the root, a hop-count system similar to the IP time-to-live (TTL) mechanism.

Use the following command to configure the maximum number of hops inside the region:

```
Switch(config)# spanning-tree mst max-hops <1-255>
```

By default, the maximum number of hops is 20. To reset it to its default value, use the following command:

```
Switch(config)# no spanning-tree mst max-hops
```

Forward Delay

Forward delay specifies the amount of time that a bridge port has to wait before it changes from the discarding and learning states to the forwarding state.

Use the following command to configure the forward delay (in seconds):

```
Switch(config)# spanning-tree mst forward-time <4-30>
```

By default, the forward delay is 15 seconds. To reset it to its default value, use the following command:

```
Switch(config)# no spanning-tree mst forward-time
```

Hello Timer

The Hello timer specifies how often the bridge transmits a configuration BPDU. Any bridge that is not the root bridge uses the root bridge hello timer value.

Use the following command to configure the hello timer (in seconds):

```
Switch(config)# spanning-tree mst hello-time <1-10>
```

By default, the hello timer is 2 seconds. To reset it to its default value, use the following command:

```
Switch(config)# no spanning-tree mst hello-time
```

Maximum Age Interval

The maximum age interval specifies the maximum time the bridge waits without receiving a configuration BPDU before it reconfigures the spanning tree topology.

Use the following command to configure the maximum age interval (in seconds):

```
Switch(config)# spanning-tree mst max-age <6-40>
```

By default, the maximum age interval is 20 seconds. To reset it to its default value, use the following command:

```
Switch(config)# no spanning-tree mst max-age
```

Bridge Priority

The bridge priority parameter controls which bridge on the network is the STG root bridge. To make one switch become the root bridge, configure the bridge priority lower than all other switches and bridges on your network. The lower the value, the higher the bridge priority. Use the following command to configure the bridge priority:

```
Switch(config)# spanning-tree mst <0-64> priority <0-61440>
```

Note: Priority must be configured in increments of 4096 (such as 0, 4096, 8192, or 12288).

By default, the bridge priority is set to 32768. To reset the bridge priority to its default value, use the following command:

```
Switch(config)# no spanning-tree mst <0-64> priority
```

Port Priority

The port priority helps determine which bridge port becomes the root port or the designated port. The case for the root port is when two switches are connected using a minimum of two links with the same path-cost. The case for the designated port is in a network topology that has multiple bridge ports with the same path-cost connected to a single segment, the port with the lowest port priority becomes the designated port for the segment.

Use the following command to configure the port priority per MST instance:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree mst <0-64> port-priority <0-224>
```

Note: Port priority must be configured in increments of 32 (such as 0, 64, 96, or 128). If the specified port priority value is not evenly divisible by 32, the value will be automatically rejected.

The default port priority is 128. To reset it to its default value, use the following command:

```
Switch(config-if)# no spanning-tree mst <0-64> port-priority
```

Port Path Cost

The port path cost assigns lower values to high-bandwidth ports, such as 10 Gigabit Ethernet, to encourage their use. The objective is to use the fastest links so that the route with the lowest cost is chosen. A value of auto (the default) indicates that the default cost will be computed for an auto-negotiated link or LAG speed.

To configure the port path cost differently on each interface, use the following command per MST instance:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# spanning-tree mst <0-64> cost {<1-200,000,000>|auto}
```

Note: The maximum configurable port path cost depends on the path cost calculation method used. If the switch uses the short method type, then port path cost can range from *1* to *65,535*. If the switch uses the long method type, then port path cost can range from *1* to *200,000,000*.

By default, the port path cost is automatically calculated based on the ethernet port speed. To reset the port path cost to its default setting, use the following command:

```
Switch(config-if)# no spanning-tree mst <0-64> cost
```

MSTP Configuration

Note the following when configuring Multiple Spanning Tree Groups:

- You may configure up to 64 MST instances and 4094 VLANs.

- Switches must have the same MST configuration identification elements (name, revision number and VLAN to MSTI mapping) to be in the same MST region.

- MST regions appear as virtual bridges connecting to single spanning trees.

- When you enable MSTP, a default revision number of 0 and a blank region name are automatically configured.

- The switch supports a single instance of the MSTP Algorithm consisting of the CIST and up to 64 MSTIs.

- A VLAN can only be mapped to one MST instance or to the CIST. One VLAN mapped to multiple spanning trees is not allowed. All the VLANs are mapped to the CIST by default. Once a VLAN is mapped to a specified MST instance, it is removed from the CIST.

MSTP Configuration Example

This section provides steps to configure MSTP on the switch.

1. Configure port and VLAN membership on the switch.

2. Configure the switch to use MSTP:

```
Switch(config)# spanning-tree mode mst
```

3. Configure the Multiple Spanning Tree region parameters:

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# name Lenovo
Switch(config-mst)# revision 5
Switch(config-mst)# instance 1 vlan 10
Switch(config-mst)# exit
```

4. Configure bridge priority for instance number 1:

```
Switch(config)# spanning-tree mst 1 priority 4096
```

5. Configure MSTP timers:

```
Switch(config)# spanning-tree mst hello-time 5
Switch(config)# spanning-tree mst max-age 10
Switch(config)# spanning-tree mst forward-time 25
```

6. Configure edge ports:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# spanning-tree port type edge
```

7. Configure BPDU guard:

```
Switch(config-if)# spanning-tree bpduguard enable
Switch(config-if)# exit
```

8. Enable the timeout mechanism for a port to be recovered automatically from being operationally shut down by BPDU:

```
Switch(config)# errdisable recovery cause bpduguard
Switch(config)# errdisable recovery interval 30
```

9. Configure Root Guard:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# spanning-tree guard root
```

10. Enable BPDU filtering:

```
Switch(config-if)# spanning-tree bpdufilter enable
```

11. Configure link type:

```
Switch(config-if)# spanning-tree link-type point-to-point
```

12. Configure path cost value and port priority:

```
Switch(config-if)# spanning-tree cost 10000
Switch(config-if)# spanning-tree port-priority 64
```

Note: To configure these parameters per instance, use the following commands:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# spanning-tree mst 1-10 port-priority 64
Switch(config-if)# spanning-tree mst 1-10 cost 10000
```

13. Configure max hops value:

```
Switch(config)# spanning-tree mst max-hops 20
```

14. Configure path cost method:

```
Switch(config)# spanning-tree pathcost method long
```

15. Verify MSTP configuration:

```
Switch# display spanning-tree mst
```

Chapter 13. Virtual Link Aggregation Groups

Virtual Link Aggregation Groups (vLAGs) are a mechanism which allows the redundant uplinks to remain active by utilizing all available bandwidth.
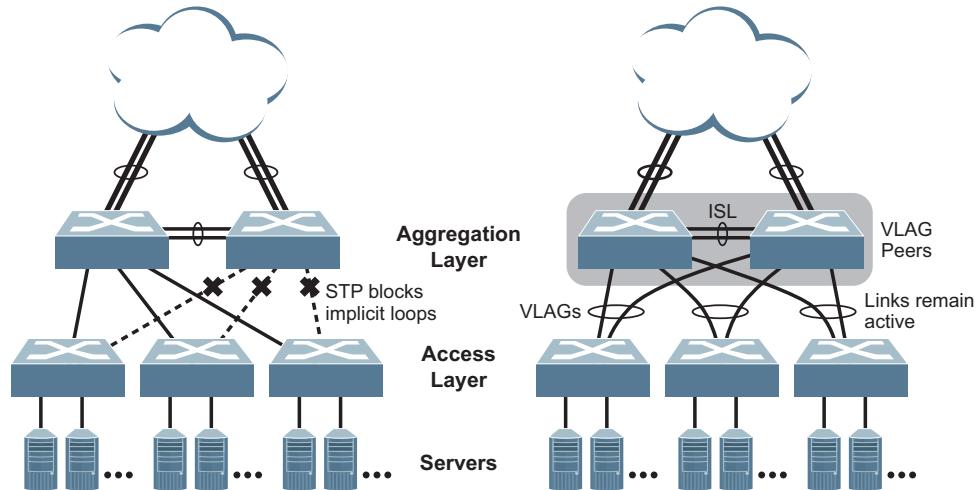
This section discusses the following topics:

vLAG Overview

In many data center environments, downstream servers or switches connect to upstream devices which consolidate traffic. For example, see Figure 5.

Figure 5. Typical Data Center Switching Layers with STP vs. vLAG

As shown in Figure 5, a switch in the access layer may be connected to more than one switch in the aggregation layer to provide for network redundancy. Typically, Spanning Tree Protocol (STP - see Chapter 12, "Spanning Tree Protocol") is used to prevent broadcast loops, blocking redundant uplink paths. This has the unwanted consequence of reducing the available bandwidth between the layers by as much as 50%. In addition, STP may be slow to resolve topology changes that occur during a link failure and can result in considerable MAC address flooding.
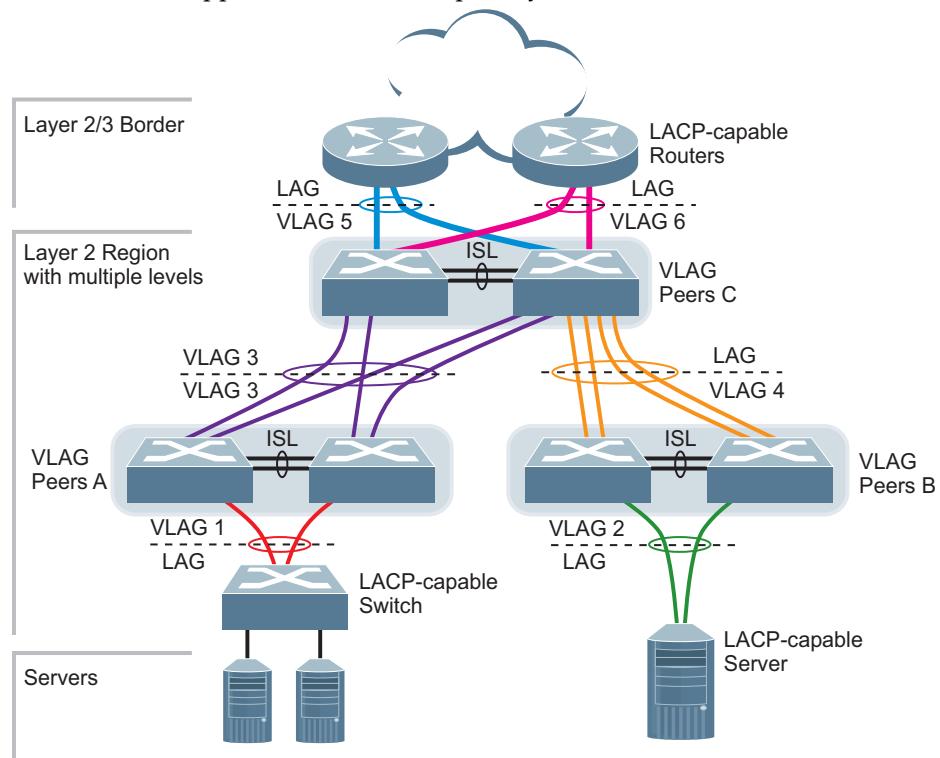
Two switches are paired into vLAG peers and act as a single virtual entity for the purpose of establishing a multi-port aggregation. Ports from both peers can be grouped into a vLAG and connected to the same LAG-capable target device. From the perspective of the target device, the ports connected to the vLAG peers appear to be a single LAG connecting to a single logical device. The target device uses the configured Tier ID to identify the vLAG peers as this single logical device. It is important that you use a unique Tier ID for each vLAG pair you configure. The vLAG-capable switches synchronize their logical view of the access layer port structure and internally prevent implicit loops. The vLAG topology also responds more quickly to link failure and does not result in unnecessary MAC flooding.

vLAGs also support Layer 2 MultiPathing, which offers redundancy and traffic load balancing across multiple paths between network nodes.

A vLAG can be set up as either a static LAG or an LACP LAG.

vLAGs are useful in multi-layer environments for both uplink and downlink redundancy to any regular LAG-capable device, such as in Figure 6.

Figure 6. vLAG Application with Multiple Layers

Wherever ports from both peered switches are aggregated to another device, the aggregated ports must be configured as a vLAG. For example, vLAGs 1 and 3 must be configured for both vLAG Peer A switches. vLAGs 2 and 4 must be configured for both vLAG Peer B switches. vLAGs 3, 5, and 6 must be configured on both vLAG Peer C switches. Other devices connecting to the vLAG peers are configured using regular static or LACP LAGs.

Note: Do not configure a vLAG for connecting only one switch in the peer set to another device or peer set. For instance, in vLAG Peer C, a regular LAG is employed for the downlink connection to vLAG Peer B because only one of the vLAG Peer C switches is involved.

vLAG Capacities

Servers or switches that connect to the vLAG peers using a multi-port vLAG are considered vLAG clients. vLAG clients are not required to be vLAG-capable. The ports participating in the vLAG are configured as regular LAGs on the vLAG client end.

On the vLAG peers, the vLAGs are configured similarly to regular LAGs, using many of the same features and rules. See Chapter 11, "Ports and Link Aggregation" for general information concerning all port LAGs.

Each vLAG begins as a regular port LAG on each vLAG-peer switch. The vLAG may be either a static LAG or a LACP LAG, and consumes one slot from the overall LAG capacity pool. The LAG type must match that used on vLAG client devices. Additional configuration is then required to implement the vLAG on both vLAG peer switches.

Depending on the switch type, you may configure up to a maximum number of LAGs, with all types (regular or vLAG, static or LACP) sharing the same pool:

- G8272: maximum of 72 LAGs
- G8296: maximum of 96 LAGs
- G8332: maximum of 104 LAGs
- NE1032T: maximum of 32 LAGs
- NE1032: maximum of 32 LAGs
- NE1072T: maximum of 72 LAGs
- NE10032: maximum of 126 LAGs
- NE2572: maximum of 72 LAGs

For the G8272, G8296, G8332, NE1072T, NE10032 and NE2572 the maximum number of supported vLAG instances is 64.

For the NE1032 and NE1032T, the maximum number of supported vLAG instances is 24.

vLAG Benefits

Using vLAGs offers the following benefits:

- Enables a single device to use a LAG across two upstream switches
- STP blocked interfaces are eliminated
- Topology loops are also eliminated
- Enables the use of all available uplink bandwidth
- Allows fast convergence times in case of link or device failure
- Allows link-level resilience
- Enables high availability

vLAG Synchronization Mechanism

All vLAG status and protocol synchronization information is transmitted between vLAG peers using the inter-switch link (ISL) LAG. vLAG uses a Layer 2 transport protocol (Edge Control Protocol or ECP) for communicating across the ISL LAG.

If the transport channel used by ECP for communication stops working, the ISL is marked as down. If vLAG health check detects that the primary switch is still functioning, the ports of the secondary switch that are members of the vLAG instance are shut down (error disabled) to avoid creating loops.

To view information about the ECP transport channel, use the following command:

```
Switch# display vlag ecp channels
```

The state of the transport channel used by ECP is detected periodically by vLAG hello packets. Both vLAG peers send hello packets every 5 seconds. If no hello packets are received for 30 seconds, the channel is marked as down.

ECP is a transport protocol that operates between two peers over an IEEE 802 LAN. ECP provides reliable, in-order delivery of ULPDUs (Upper Layer Protocol Data Units).

To view information about the upper layer protocols associated with ECP, use the following command:

```
Switch# display vlag ecp upper-layer-protocols
```

ECP has the following service characteristics:

- Reliable delivery of ULPDUs, that is resilient against frame loss

- Delivery of ULPDUs to the recipient Upper Layer Protocol (ULP) in the order that they were transmitted by the sending ULP

- Delivery of a single copy of each ULPDU to the recipient

- Flow control that provides protection against buffer overrun on the receiving side

vLAG System MAC

vLAG supports both static LAGs and LACP LAGs. For connections between an access switch and an end host, vLAGs that are configured using LACP LAGs will use a special reserved MAC address called vLAG system MAC.

The vLAG system MAC address is created using the vLAG tier ID. Together with the LACP system priority, the MAC address is used to generate a LACP system ID, which is used during LACP negotiation. For more details about LACP, see "Link Aggregation Control Protocol" on page 266.

vLAG and LACP Individual

By default, physical interfaces that are members of an LACP LAG that do not receive any Link Aggregation Control Protocol Data Units (LACPDUs) after a set amount of time will be expelled from the LAG and enter individual state. Interfaces in the individual state will function as normal physical ports. For more details, see "LACP Individual" on page 268.

If the interface is part of a vLAG, it will be marked as a non-vLAG port when it enters individual state. The port will forward and block traffic according to the upper layer protocol, for example, STP.

This behavior can lead to the creation of loops when the vLAG does not connect the switch to an end host. To avoid such scenarios, only LACP LAGs with LACP individual enabled interfaces that are connected to an end device should be added to the vLAG. When you assign an LACP LAG to a vLAG, the switch will display a warning if it detects that the LAG member ports have LACP individual enabled.

```
warning: port-aggregation 1 with lacp individual enabled is added for
VLAG instance 1
```

Notes:

- The LAG used for the vLAG ISL cannot have LACP individual enabled.
- When the upper layer protocol (for example, STP) is disabled, you must manually ensure that there are no loops.

vLAG and LACP System Priority

To reduce configuration dependency, LACP system priority information is synchronized between the vLAG peers. The secondary switch will use the system priority of the primary switch to reaggregate with the access switch, regardless of its own priority. This ensures that LACP LAG member ports that participate in a vLAG are aggregated in the same LAG on the access switch, even if the LACP system priorities of the primary and secondary switches do not match.

If the secondary switch becomes the primary, it will replace its current LACP system priority with its original value and then reaggregate with the access switch.

vLAG LACP Misconfigurations or Cabling Errors

On a single switch, LACP can ensure that only one LAG will form within an aggregation group, despite incorrect configurations or cabling. In a vLAG topology, LACP misconfiguration or erroneous cabling can cause the formation of two LAGs within an aggregation group, thus creating a network loop.

The switch detects such LACP misconfigurations or cabling errors. After detection, vLAG will suspend the ports that are part of the vLAG on the secondary switch and a syslog message will be generated. Once the misconfiguration or cabling error is corrected, vLAG will include the suspended ports to be part of the LAG.

Any misconfiguration will be checked only on the secondary vLAG switch. The vLAG primary switch will synchronize partner information with the secondary switch, followed by a misconfiguration check on the secondary vLAG switch. When a misconfiguration occurs, the following syslog message will be generated:

```
LACP port <port name>'s received partner info is mismatch with vLAG
instance <1-64>'s expected one, it will be suspended.
```

Note: On the primary vLAG switch, no misconfiguration syslog messages will be displayed.

FDB Synchronization

To prevent the flooding of unknown unicast packets and offer local preference, the Forwarding Databases (FDB) are synchronized between the vLAG peers.

MAC addresses learned by a switch on its vLAG ports are synchronized across the corresponding vLAG ports of its peer. MAC addresses not learned on vLAG ports are synchronized over the ISL of the vLAG peers.

Note: Up to 32k MACs can be synchronized.

A peer MAC address is added as a static unicast FDB entry across the ISL to avoid unnecessary traffic flooding of the vLAG peer. A peer MAC address also permits Address Resolution Protocol (ARP) entries to be shared across the ISL when FDB learning is disabled after the FDB synchronization process starts, thus allowing Layer 3 traffic to the peer to work properly.

FDB synchronization will start as soon as the first vLAG instance is formed and will stop when the last vLAG instance will be disbanded.

The member ports of the vLAG ISL will disable MAC learning once FDB synchronization starts and enable it once the FDB synchronization stops. This happens regardless of the configured MAC learning settings of the ports prior to the starting of FDB synchronization.

Synchronized MAC entries will not be aged or cleaned by the local switch. When this happens, the MAC entries will be re-installed in the FDB. To avoid certain scenarios where an aged MAC entry is not re-installed fast enough, thus triggering a traffic flood, vLAG uses a timer to periodically check the status of all FDB entries and re-install any MAC addresses that are scheduled for cleaning. For more details, see "FDB Refresh" on page 314.

To get FDB learning and aging callbacks, vLAG uses the FDB interrupt mechanism.

vLAG and STP

The switch supports the Spanning Tree Protocol (STP) running over individual vLAG instances. It supports both STP modes: Rapid PVST+ and MSTP. For more details about STP, see Chapter 12, "Spanning Tree Protocol".

Bridge protocol data units (BPDU) are received by both vLAG peers and are sent only through the link that received the BPDUs from the upstream switches. A BPDU is a data message transmitted across a local area network to detect loops in network topologies. A BPDU contains information regarding ports, switches, port priority, and addresses, which is necessary to configure and maintain the spanning tree topology.

Note: If a vLAG switch with STP enabled is connected to a switch with STP disabled, BPDUs will be flooding back to vLAG switch causing the vLAG port be put into the backup state and discard traffic if there are multiple links.

Both vLAG peers use the vLAG system MAC address as the designated bridge ID on their vLAG member ports. The root bridge ID remains unchanged.

The ISL LAG forwards traffic even if the root bridge is configured on a switch not participating in the vLAG. In this scenario, it is possible that the switch will display two root ports. MSTP will consider the ports that are members of the ISL LAG as such only if the ISL has formed. If the ISL fails, the ports are considered regular ports and may be blocked.

STP will calculate the path cost of non-vLAG ports as normal. For vLAG ports, the protocol assigns a fixed value of 1, regardless of the path cost method used.

For regular LAGs, STP uses a dynamic calculation to determine the path cost of the LAG. This method is based on the path cost of the LAG member ports. When a LAG member's state changes to down, the path cost of the LAG is recalculated.

For LAGs participating in a vLAG, STP uses a fixed path cost value of 1, instead of the dynamic calculation described above. This means that a vLAG has better path preference than a regular single port. However, when a vLAG member's state changes to down, the path cost of the LAG is not recalculated.

The path cost of the LAGs involved in the vLAG can be changed to another value than 1, but this must be manually configured on both vLAG peers. The new path cost must be identical on both primary and secondary switches. When this condition is not met, the results of the STP path cost calculation are incorrect.

To ensure that vLAG works properly with MSTP, the vLAG peers must be part of the same MST region. When this condition is not met, the vLAG ports of the secondary switch will be considered as being disabled (not error disabled) in the MSTP configuration. This will negatively affect traffic flow through the vLAG. Once the vLAG peers are correctly configured as being part of the same MST region, the vLAG ports of the secondary switch will transition to the up state.

BPDU guard and BPDU filter will consider the vLAG as a single link. For example, when the BPDU guard will receive a BPDU packet, the vLAG ports from both vLAG peers will be put in the error disabled state.

As long as one of the vLAG instances running on the vLAG peers has not failed, a vLAG port is visible in the STP configuration, even if the interface is in the administratively down or error disabled state. This means that the status of the vLAG port is not determined by its physical state, but instead by the vLAG. STP considers the vLAG port as being a single logical link and is oblivious if one of the vLAG members fails. STP will mark the vLAG port as being in the down state only if the vLAG link fails on both vLAG peers.

vLAG and VRRP

The switch supports both vLAG and VRRP running simultaneously. To enable vLAG peer switches to function as a gateway, VRRP must be in active-active mode. Both the primary and secondary vLAG switches can forward Layer 3 traffic across the virtual router instance independent of their VRRP state (master or backup virtual router). The backup virtual router will act as a master when the vLAG is formed. For more details about VRRP, see "Virtual Router Redundancy Protocol" on page 543.

Notes:

- Only a maximum of 64 VRRP instances can be configured on a switch that is part of a vLAG.

- The VRRP Advertisement Timer cannot be configured with a time interval below 100 centiseconds (1 second).

vLAG VRRP Passive Mode (Half Active-Active)

In passive mode or half active-active mode, the vLAG backup virtual router checks if its vLAG peer is the master virtual router. If none of the vLAG peer switches is the master virtual router, a Layer 3 routing entry for the VRRP domain will not be installed on the vLAG backup virtual router. Only the master virtual router and its vLAG peer, which is also the backup virtual router, are in the IP active state. The vLAG backup virtual router will install a Layer 3 routing entry. If the master virtual router fails, the newly elected master virtual router and its vLAG peer, that has become the backup virtual router, will transition to the IP active state.

Note: vLAG VRRP passive mode is applicable only in a 4xVRRP configuration, a two vLAGs configuration, or a vLAG multiple tier configuration.

To enable vLAG VRRP passive mode on the switch, use the following command:

```
Switch(config)# no vlag vrrp active
```

vLAG VRRP Active Mode (Full Active-Active)

When the vLAG peer switches are in VRRP active mode or full active-active mode, they are in the IP active state. This means that the vLAG peer switches will install the Layer 3 routing entry regardless of their virtual router role or that their vLAG peer are the master or backup virtual router.

Note: vLAG VRRP active mode is applicable only in a 4xVRRP configuration, a two vLAGs configuration, or a vLAG multiple tier configuration.

By default, vLAG peers function in VRRP active mode. To enable it, use the following command:

```
Switch(config)# vlag vrrp active
```

vLAG Configuration Consistency Check

vLAG operates simultaneously on two different switches that perform as one logical device. If there any incompatible configuration differences between the two vLAG peers, this may lead to undesirable effects, such as traffic loss.

The vLAG Configuration Consistency Check ensures that the network behaves correctly. Using the vLAG synchronization mechanism, vLAG Consistency Check will verify the configurations of the vLAG peers for incompatibilities. Each configuration parameter is classified accordingly with its associated priority and different actions will be applied when an incompatibility is detected based upon the parameter's priority.

Whenever a high priority parameter is detected as being inconsistent across vLAG peers, a syslog message with a critical severity level is logged and the following actions are taken:

- if the inconsistent parameter is global or related to the ISL, the Link Aggregation Groups (LAGs) of all the vLAG instances on the vLAG secondary switch are put into the down state

- if the inconsistent parameter is related to the vLAG instance, the LAG associated with that instance is put into the down state

If the detected inconsistency refers to a low priority parameter, by default the switch will only record a syslog message with a warning severity level for that inconsistency. To allow the switch to perform the same actions as when dealing with high priority parameters, strict consistency checking must be manually enabled.

The following table lists the parameters verified during a configuration consistency check:

Table 22. *Configuration Consistency Check Parameters*

Switch Feature	Parameter	Type of parameter	Priority
FDB	Global MAC address learning	Global	high
FDB	MAC address learning per interface	Instance	high
VLAN	Switch port mode	ISL/instance	high
VLAN	Access VLAN	ISL/instance	high
VLAN	Native VLAN	ISL/instance	high
VLAN	Allowed VLAN list	ISL/instance	high
VLAN	Global 802.1Q native VLAN tagging	Global	high
VLAN	802.1Q native VLAN tagging per interface	ISL/instance	high
VLAN	Tag ingress	ISL/instance	high
LACP	Aggregation group type	Instance	high
LACP	Suspend individual	Instance	low

Table 22. *Configuration Consistency Check Parameters (continued)*

Switch Feature	Parameter	Type of parameter	Priority
STP	STP mode	Global	high
STP	STP path cost	Global	high
STP	MST region name	Global	high
STP	MST region revision number	Global	high
STP	MST instance to VLAN mapping	Global	high
STP	MST maximum age timer	Global	low
STP	MST maximum number of hops	Global	low
STP	MST hello timer	Global	low
STP	MST forwarding timer	Global	low
STP	STP port mode	Instance	high
STP	STP port path cost	Instance	high
STP	STP port type	Instance	high
STP	MST port path cost	Instance	high
STP	STP port BPDU filter	Instance	low
STP	STP port BPDU guard	Instance	low
STP	STP port root guard	Instance	low
STP	STP port loop guard	Instance	low
STP	STP link type	Instance	low
STP	STP port priority	Instance	low
STP	MST port priority	Instance	low

By default, configuration consistency checking is enabled on the switch. To function properly, consistency checking must be enabled on both vLAG peers.

To disable configuration consistency checking, use the following command:

```
Switch(config)# vlag config-consistency disable
```

To enable configuration consistency checking, use the following command:

```
Switch(config)# no vlag config-consistency disable
```

By default, strict configuration consistency checking is disabled. To enable or disable strict configuration consistency checking, use the following command:

```
Switch(config)# [no] vlag config-consistency strict
```

vLAG and IGMP Snooping

For more details about IGMP Snooping, see "Internet Group Management Protocol" on page 431.

Multicast Router Synchronization

When the switch receives an IGMP Query message or a Protocol Independent Multicast (PIM) Hello message, it installs an IGMP multicast router. The switch sends a vLAG synchronization message to its vLAG peer, which also installs the new IGMP multicast router. If the switch receives the IGMP Query message or PIM Hello message on a non-vLAG port, the vLAG peer will install the multicast router on the ISL LAG.

IGMP Groups Synchronization

When the switch receives an IGMP Report message, it installs an IGMP group in its IGMP group table. The switch sends a vLAG synchronization message to its vLAG peer, which also installs the new IGMP group. If the switch receives the IGMP Report messages on a non-vLAG port, the vLAG peer will install the IGMP group on the ISL LAG.

When the switch receives an IGMP Leave message, it removes the specified IGMP group from the IGMP group table. The switch sends a vLAG synchronization message to its vLAG peer, which also removes the IGMP group. The IGMP Leave message is forwarded to multicast routers, except those installed on the ISL LAG.

IGMP groups can be statically configured on switch ports that are part of the vLAG or the ISL LAG. The IGMP group configuration must be consistent on both vLAG peer switches.

Note: The same IGMP group cannot simultaneously be dynamically installed or statically configured on the same switch interface if vLAG is enabled on the switch.

IGMP Querier Synchronization

The IGMP Querier can be enable on both vLAG peer switches. The vLAG peers must share the same IP address and have the same IGMP Querier configuration.

After the IGMP Querier election, both vLAG peer switches can act as the Querier at the same time, but only the vLAG primary switch will send IGMP Query messages when the vLAG is in formed state. The *other querier present timer* will be synchronized between the vLAG peers.

When an STP topology change occurs on a switch interface, the vLAG primary switch will initiate General Query messages across all interface/VLAN pairs belonging to the Spanning Tree Group (STG) specified in the Topology Change Notification (TCN) message. If the switch does not receives any replies to the General Query message, the IGMP group entries are deleted from the IGMP group table.

Note: When vLAG and IGMP run together and STP is also enabled on the switch, it is recommended that TCN flood is disabled to avoid IGMP table deletion after the vLAG switches recover from a failure. To disable TCN flood, use the following command:

```
Switch(config)# no ip igmp snooping tcn flood
```

vLAG Peer Gateway

The two vLAG peer switches cannot be used as a gateway for a network route. This occurs because the two switches have different MAC addresses and when traffic reaches the vLAG, it can be sent to either of the two vLAG peers. For example, if a packet that has the Layer 2 destination the MAC address of one vLAG switch is sent to the other switch, the packet will be discarded.

The vLAG Peer Gateway allows a vLAG switch to function as the active gateway for traffic that is addressed to its vLAG peer. It enables the local forwarding of such packets between vLAG peers without using the ISL. This optimizes the use of the ISL and avoids potential traffic loss.

By default, the vLAG Peer Gateway is disabled. For it to function properly, it must be enabled on both vLAG peers.

To enable or disable the vLAG Peer Gateway, use the following command:

```
Switch(config)# [no] vlag peer-gateway
```

To check the status of the vLAG Peer Gateway, use the following command:

```
Switch> display vlag information
```

vLAGs versus regular LAGs

Though similar to regular LAGs in many regards, vLAGs differ from regular LAGs in the following ways:

- A vLAG can consist of multiple ports on two vLAG peers, which are connected to one logical client device such as a server, switch, or another vLAG device.

- The participating ports on the client device are configured as regular LAG member ports.

- The vLAG peers must be the same model and run the same software version.

- vLAG peers require a dedicated inter-switch link (ISL) for synchronization. The ports used to create the ISL must have the following properties:
  - ISL ports must be configured for all vLAG VLANs
  - ISL ports must be placed into a regular LAG (either static or LACP)
  - A minimum of two ports on each switch are recommended for ISL use

- Dynamic routing protocols, such as OSPF, cannot terminate on vLAGs.

- vLAGs are configured using additional commands.

- It is recommended that end-devices connected to vLAG switches use NICs with dual-homing. This increases traffic efficiency, reduces ISL load, and provides faster link failover.

Configuring vLAGs

It is recommended that any vLAG configuration is done without traffic flowing through the vLAG peers. This can cause FDB flush events, resulting in a delayed stabilization of the vLAG configuration.

Note: The vLAG peers can be reloaded while traffic is flowing, but vLAG configuration changes are not recommended during the switch startup process if vLAG Startup Delay has been enabled on the switch.

When configuring vLAG or making changes to your vLAG configuration, consider the following vLAG behavior:

● When vLAG is enabled, you may see two root ports on the secondary vLAG switch. One of these will be the actual root port for the secondary vLAG switch and the other will be a root port synchronized with the primary vLAG switch.

● The STG to VLAN mapping on both vLAG peers must be identical.

The following parameters must be identically configured on the vLAG ports of both the vLAG peers:

● VLANs

● Native VLAN tagging

● Native VLAN/PVID

● STP mode

● BPDU Guard setting

● STP port setting

● MAC aging timers

● Static MAC entries

● ACL configuration parameters

● QoS configuration parameters

When configuring a vLAG, the participating switches are checked for matching on the following parameters:

● Tier ID - it is used to divide vLAG domains. Different domains cannot be aggregated together.

● System type - both vLAG peers must be the same switch model to reduce the impact of different system performance and capability.

● CNOS version - it is used to avoid vLAG feature compatibility issues.

If the Tier ID or system type do not match, the vLAG will not form. If the peers use different CNOS versions, the vLAG will form, but notification messages will be displayed every 10 seconds, informing you to upgrade to the same CNOS version.

By default, the vLAG feature is disabled.

To enable or disable vLAG, use the following command:

```
Switch(config)# [no] vlag enable
```

vLAG ISL

vLAG uses an Inter-Switch Link (ISL) connection to synchronize information between the two vLAG peers. The ISL is a dedicated regular LAG (static or LACP) that must be formed from at least two physical interfaces.

It is recommended that the total bandwidth of the ISL LAG be larger than the total bandwidth of the vLAG ports. When a vLAG port fails, its traffic will be forwarded through the ISL LAG.

To configure the LAG used by the ISL, use the following command:

```
Switch(config)# vlag isl port-aggregation <1-4096>
```

Note: This command also triggers the forming of the ISL between the vLAG peers.

To delete the vLAG ISL, use the following command:

```
Switch(config)# no vlag isl port-aggregation
```

vLAG Role Election

For a centralized vLAG operation, vLAG assigns a role to each of the two switches participating in the vLAG: primary and secondary. The primary switch controls the centralized operation.

Each vLAG role is determined by comparing the local vLAG priorities and MAC addresses of the participating switches. The switch with the lower priority will become the primary. If the switches are configured with identical priorities, the switch with the smaller MAC address will be elected as the primary.

The vLAG role election is not preemptive. If a switch is already elected as the primary, another switch that joins the vLAG will become the secondary and remain in that role even if its priority or MAC address are smaller than those of the primary switch.

To configure the vLAG priority of a switch, use the following command:

```
Switch(config)# vlag priority <0-65535>
```

By default, the switch has a vLAG priority of 0. To reset the priority to the default value, use the following command:

```
Switch(config)# no vlag priority
```

vLAG Instance

vLAGs are configured using an instance ID. The instance ID is used to identify which LAGs are connected to downstream devices as a vLAG. A single vLAG instance can group LAGs with different IDs on the vLAG peers. For example, vLAG instance 1 can bind together LAG 1 from the primary vLAG peer and LAG 6 from the secondary vLAG peer.

The instance ID must be the same for both vLAG peers.

The maximum number of configurable vLAG instances is 64.

When creating a vLAG, you must also assign a LAG to it. To create a vLAG instance, use the following command:

```
Switch(config)# vlag instance <1-64> port-aggregation <1-4096>
```

Note: Only one LAG can be assigned to a vLAG on a single switch.

By default, vLAG instances are disabled. To enable or disable a vLAG instance, use the following command:

```
Switch(config)# [no] vlag instance <1-64> enable
```

To remove the associated LAG from the vLAG instance, use the following command:

```
Switch(config)# no vlag instance <1-64> port-aggregation
```

Note: When removing the assigned LAG from an enabled vLAG instance, the instance will also be disabled.

To remove a vLAG instance, use the following command:

```
Switch(config)# no vlag instance <1-64>
```

vLAG formation depends on the state of the two participating LAGs (local and remote). The possible states of the vLAG are listed below:

- DOWN - both the local and remote LAGs are in the down state
- LOCAL_UP - only the local LAG is in the up state
- REMOTE_UP - only the remote LAG is in the up state
- FORMED - both the local and remote LAGs are in the up state

Note: If the ISL is in the down state, vLAG synchronization is not possible and the remote vLAG will be treated as being in the down state regardless of its real state.

FDB Refresh

vLAG uses a timer (FDB refresh interval) to periodically check the status of synchronized FDB entries. If a MAC address is detected as waiting to be removed from the FDB, the address will instead be re-installed in the FDB.

The FDB refresh interval is shorter than the MAC aging time, which can be configured by using the following command:

```
Switch(config)# mac address-table aging-time <0-1000000 seconds>
```

By default, FDB refresh is enabled. To enable or disable FDB refresh, use the following command:

```
Switch(config)# [no] vlag mac-address-table refresh
```

Note: If MAC aging time is configured below 40 seconds, FDB refresh will not work regardless if it is enabled or not on the switch.

vLAG Tier ID

vLAG uses a Tier Identifier to separate different vLAG domains. This is helpful when configuring multiple vLAGs in a multi-tier network environment, because vLAG domains with different configured tier IDs cannot be aggregated together.

To configure the tier ID of a vLAG, use the following command:

```
Switch(config)# vlag tier-id <1-512>
```

To remove a vLAG's tier ID, use the following command:

```
Switch(config)# no vlag tier-id
```

vLAG Startup Delay

A startup delay is used to prevent traffic loss while a vLAG peer reloads. During the reload process, a switch's vLAG ports are in the error disabled state. Once the ISL is established, the vLAG ports transition to the up state one at a time only after the configured startup delay has passed.

The startup delay is initiated only after the ISL is established. Until this timer has expired, all vLAG ports will be maintained in the error disabled state. This delay will allow the switch to properly initiate BGP or OSPF for routing traffic through the upstream links. This ensures that traffic will flow smoothly once the vLAG ports are up.

Any vLAG ports configured in the administrative up or down state will retain their state after the startup delay has passed. During the delay, their status is error disabled.

During the startup delay interval, administratively up ports that become members of the vLAG instance will be error disabled and administratively up ports that leave the vLAG instance will transition back to the up state.

Any vLAG ports that are configured in the administrative up state during the startup delay are immediately put in the error disabled state until the delay interval expires.

Note: vLAG configuration changes are not recommended during the reload process of a vLAG peer on which the startup delay timer has not expired.

To configure the startup delay, in seconds, use the following command:

```
Switch(config)# vlag startup-delay <0-3600>
```

By default, the startup delay is 120 seconds (2 minutes). To reset the interval to its default value, use the following command:

```
Switch(config)# no vlag startup-delay
```

Note: The startup delay is cancelled if the two vLAG peers reload simultaneously or the ISL fails.

vLAG Auto-recovery

During network operations, it is common for the vLAG peers to be reloaded simultaneously. In this scenario, if the vLAG ports are in the error disabled state, traffic will be lost. To avoid traffic loss, bring the ports up manually. When this is not done, an auto-recovery mechanism is required to automatically transition ports from the error disabled state to the up state.

vLAG auto-recovery lets the switch automatically re-enable any error disabled vLAG ports after a specified time interval if it detects that its vLAG peer is not functional.

During the reloading process, the auto-recovery timer is initiated. If the ISL is established between the vLAG peers before the auto-recovery timer expires, the timer is cancelled.

If the auto-recovery timer expires or the ISL fails before that, auto-recovery stops the vLAG startup-delay process and then determines the status of the vLAG peer through Health Check. If the peer is functional, all of the switch's vLAG ports are maintained in the error disabled state. If the vLAG peer is not detected as functional, the switch assumes the role of the primary, and then transitions all of its vLAG ports to the up state.

To configure the auto-recovery interval, in seconds, use the following command:

```
Switch(config)# vlag auto-recovery <240-3600>
```

Notes:

● It is recommended that the auto-recovery interval be longer than the reload duration of the switch. Some vLAG ports might remain in the error disabled state if this condition is not met.

● Any changes to the vLAG auto-recovery mechanism will not take effect until the switch is reloaded.

vLAG auto-recovery is always enabled and cannot be disabled. By default, the auto-recovery interval is set to 300 seconds (5 minutes). To reset the interval to its default value, use the following command:

```
Switch(config)# no vlag auto-recovery
```

Health Check

The status of a vLAG peer is usually monitored through the ISL. You can configure vLAG Health Check to provide an alternate way of checking the status of the peer when the ISL fails.

Health Check detects and mitigates the failure of the vLAG ISL. When Health Check determines that the vLAG peer is still functioning, but the ISL has failed, the path from the secondary switch to the access switch will be shut down.

The Health Check mechanism can be configured either over the management interface or over an ethernet port.

Note: Health Check supports both IPv4 and IPv6 addressing. When using IPv6 addressing, only global addresses are allowed. Link local IPv6 addresses are not permitted.

To configure vLAG peer IP address used for Health Check, use the following command:

```
Switch(config)# vlag hlthchk peer-ip <peer IP address> [vrf {default|
|management}]
```

To remove the vLAG peer, use the following command:

```
Switch(config)# no vlag hlthchk peer-ip
```

If a switch is reload and its ISL is in the down state, the switch is not elected. When both the primary and secondary switches are reloaded with ISL in the down state, all the ports that are members of the vLAG from both switches will transition to the error disabled state.

For the Health Check mechanism to work, the vLAG peers should be in the same subnetwork (preferably directly connected).

The Health Check mechanism uses hello messages to determine the status of the vLAG peer. The hello messages are exchanged periodically between the vLAG peers over a TCP session.

To configure the interval, in seconds, between consecutive hello messages, use the following command:

```
Switch(config)# vlag hlthchk keepalive-interval <2-300>
```

By default, hello messages are exchanged every 5 seconds. To reset the hello interval to its default value, use the following command:

```
Switch(config)# no vlag hlthchk keepalive-interval
```

The hello messages are used to maintain the TCP session between the vLAG peers. After sending a certain number of consecutive hello messages and not receiving any reply, the local switch will declare its vLAG peer as being in the down state.

To configure the number of unanswered hello messages, use the following command:

```
Switch(config)# vlag hlthchk keepalive-attempts <1-24>
```

By default, the vLAG peer is marked as down after the local switch does not receive a reply to 3 consecutive hello messages. To reset the number of attempts to the default value, use the following command:

```
Switch(config)# no vlag hlthchk keepalive-attempts
```

When a TCP session between the two vLAG peers cannot be established, the Health Check mechanism will retry to reconnect to the peer at a certain interval. To configure the retry interval, in seconds, use the following command:

```
Switch(config)# vlag hlthchk retry-interval <1-300>
```

By default, the retry interval is set at 30 seconds. To reset it to this value, use the following command:

```
Switch(config)# no vlag hlthchk retry-interval
```

Basic Health Check Configuration Example

To configure the vLAG Health Check mechanism, follow these steps:

1. Configure the management interface or an ethernet port of the switch:

```
Switch(config)# interface mgmt 0
Switch(config-if)# ip address 10.10.10.1 255.255.255.0
Switch(config-if)# exit
```

Note: Configure a similar interface on vLAG Peer 2. For example, use IP address 10.10.10.2.

2. Specify the IPv4 address of the vLAG peer:

```
Switch(config)# vlag hlthchk peer-ip 10.10.10.2 vrf management
```

Note: For vLAG Peer 2, the interface should be configured as 10.10.10.2 and the health check peer IP should be configured as 10.10.10.1, pointing back to vLAG Peer 1.

The local health check peer IP address cannot be configured manually. It is automatically configured with the IP address of the interface Health Check is using, within the same subnetwork as the peer's vLAG health check IP address.
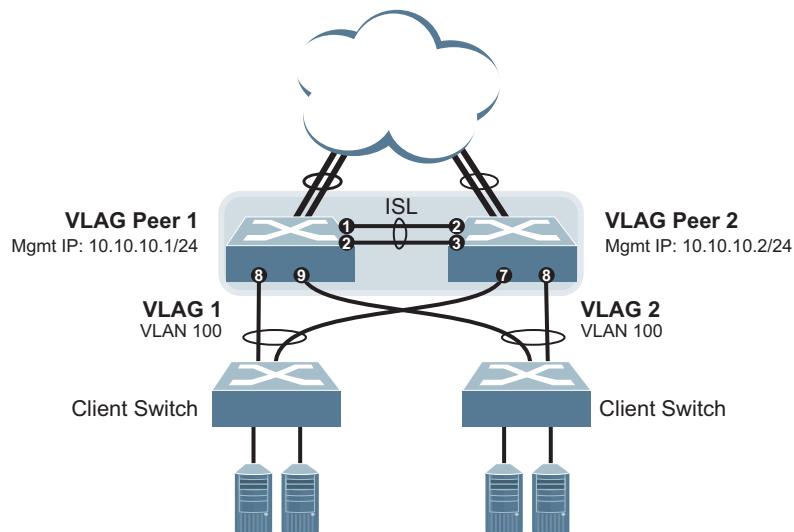
After configuring the peer IP address, the Health Check mechanism will attempt to establish a TCP session with the specified switch. To close the connection, use the following command:

```
Switch(config)# no vlag hlthchk peer-ip
```

Basic vLAG Configuration Example

Figure 7 shows an example configuration where two vLAG peers are used for aggregating traffic from downstream devices.

Figure 7. Basic vLAGs

In this example, each client switch is connected to both vLAG peers. On each client switch, the ports connecting to the vLAG peers are configured as members of a LACP LAG. The vLAG peer switches share a dedicated ISL for synchronizing vLAG information. On the individual vLAG peers, each port leading to a specific client switch (and part of the client switch's port LAG) is configured as a vLAG.

In the following example configuration, only the configuration for vLAG 1 on vLAG Peer 1 is shown. vLAG Peer 2 and all other vLAGs are configured in a similar fashion.

Note: When making changes to a running vLAG configuration, impact can be minimized by temporarily disabling vLAG configuration consistency check.

Configuring the ISL

The ISL connecting the vLAG peers is shared by all their vLAGs. The ISL needs to be configured only once on each vLAG peer.

Note: Lenovo recommends setting one unused VLAN as the native VLAN for the ISL path.

1. Configure the ISL ports and place them into a port LAG:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if-range)# bridge-port mode trunk
Switch(config-if-range)# bridge-port trunk allowed vlan 100
Switch(config-if-range)# aggregation-group 10 mode active
Switch(config-if-range)# exit
```

Notes:

- In this case, a LACP LAG is shown. A static LAG can be configured instead.

- The ISL ports and vLAG ports must be members of the same VLANs.

- LAG 10 is automatically created when you assign the ethernet ports to it.

- All VLANs can be added to the allowed VLAN list by using the following command:

```
Switch(config)# interface port-aggregation 10
Switch(config-if)# bridge-port trunk allowed vlan all
```

When adding all VLANs to the allowed VLAN list of a switch trunk port, VLAN configuration needs to be the same on both vLAG peers to avoid configuration consistency check failure.

2. Associate the LAG to the ISL:

```
Switch(config)# vlag isl port-aggregation 10
```

3. Configure Health Check:

```
Switch(config)# vlag htlhchk peer-ip 10.10.10.2 vrf management
```

4. Configure VLAG Tier ID:

```
Switch(config)# vlag tier-id 10
```

5. Configure the ISL for the vLAG peer.

Make sure you configure the vLAG peer (vLAG Peer 2) with the same ISL aggregation type (LACP or static), the same VLAN for vLAG and ISL ports, STP mode, and tier ID used on vLAG Peer 1.

Configuring the vLAG

To configure the vLAG, follow these steps:

1. Create VLAN 100:

```
Switch(config)# vlan 100
Switch(config-vlan)# exit
```

2. Configure the vLAG ports:

```
Switch(config)# interface ethernet 1/8
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 100
Switch(config-if)# aggregation-group 1 mode active
Switch(config-if)# exit

Switch(config)# interface ethernet 1/9
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 100
Switch(config-if)# aggregation-group 2 mode active
Switch(config-if)# exit
```

Notes:

- In MSTP mode, VLANs are automatically mapped to CIST.

- LAGs 1 and 2 are automatically created when you assign ethernet interfaces 8 and 9 to them.

3. Associate the ports with their respective vLAGs:

```
Switch(config)# vlag instance 1 port-aggregation 1
Switch(config)# vlag instance 1 enable
Switch(config)# vlag instance 2 port-aggregation 2
Switch(config)# vlag instance 2 enable
```

4. Enable vLAG globally:

```
Switch(config)# vlag enable
```

5. Verify the completed configuration:

```
Switch# display vlag information
```

6. Repeat the configuration for vLAG Peer 2.

   For each corresponding vLAG on the peer, the port LAG type (LACP or static), VLAN, STP mode, and Tier ID must be the same as on vLAG Peer 1.

vLAG Configuration - VLANs Mapped to a MST Instance

Follow the steps in this section to configure vLAG in environments where the STP mode is MSTP and no previous vLAG was configured.

Configuring the ISL

The ISL connecting the vLAG peers is shared by all their vLAGs. The ISL needs to be configured only once on each vLAG peer. Ensure you have the same region name, revision, and VLAN-to-STG mapping on both vLAG switches.

Note:

1. Configure STP:

```
Switch(config)# spanning-tree mode mst
```

2. Configure the ISL ports and place them into a LAG (LACP or static):

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if-range)# bridge-port mode trunk
Switch(config-if-range)# bridge-port trunk allowed vlan 100
Switch(config-if-range)# aggregation-group 1 mode active
Switch(config-if-range)# exit
```

Notes:

- In this case, a LACP LAG is shown. A static LAG can be configured instead.
- ISL ports and vLAG ports must be members of the same VLAN.

3. Associate the LAG to the ISL:

```
Switch(config)# vlag isl port-aggregation 1
```

4. Configure the vLAG Tier ID:

```
Switch(config)# vlag tier-id 10
```

5. Configure the ISL for the vLAG peer.

Make sure you configure the vLAG peer (vLAG Peer 2) with the same ISL aggregation type (LACP or static), the same VLAN for vLAG and ISL ports, STP mode, and tier ID used on vLAG Peer 1.

Configuring the vLAG

To configure the vLAG, follow these steps:

1. Create the VLAN:

```
Switch(config)# vlan 100
Switch(config-vlan)# exit
```

2. Configure the vLAG ports:

```
Switch(config)# interface ethernet 1/3
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 100
Switch(config-if)# aggregation-group 3 mode active
Switch(config-if)# exit

Switch(config)# interface ethernet 1/4
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 100
Switch(config-if)# aggregation-group 4 mode active
Switch(config-if)# exit
```

Notes:

- In MSTP mode, VLANs are automatically mapped to CIST.
- LAGs 3 and 4 are automatically created when you assign ethernet interface 3 and 4 to them.

3. Map the VLAN to an MST instance:

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# instance 1 vlan 100
```

4. Associate the ports with their respective vLAGs:

```
Switch(config)# vlag instance 1 port-aggregation 3
Switch(config)# vlag instance 1 enable
Switch(config)# vlag instance 2 port-aggregation 4
Switch(config)# vlag instance 2 enable
```

5. Enable vLAG:

```
Switch(config)# vlag enable
```

6. Verify the completed configuration:

```
Switch# display vlag information
```

7. Configure the ISL for the vLAG peer.

   For each corresponding vLAG on the peer, the port LAG type (LACP or static), VLAN, STP mode, and Tier ID must be the same as on vLAG Peer 1.

Configuring vLAGs in Multiple Layers

Figure 8. vLAG in Multiple Layers
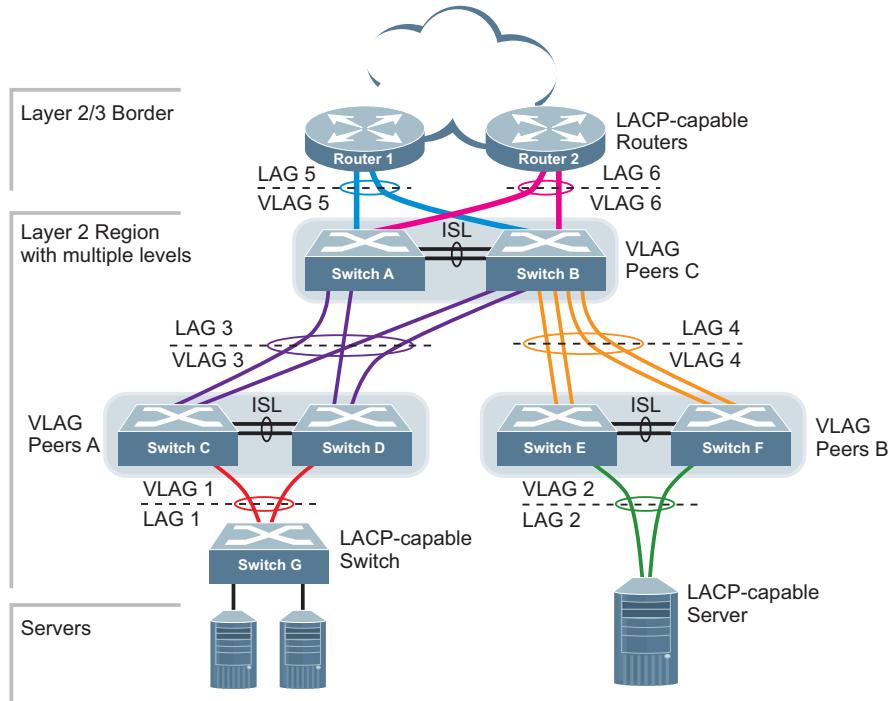
Figure 8 shows an example of a vLAG being used in a multi-layer environment. Following are the configuration steps for the topology.

Task 1: Configure Layer 2/3 Border Region

Consider the following:

- Border routers 1 and 2 are connected to Switch A through ethernet port 1
- Border routers 1 and 2 are connected to Switch B through ethernet port 2

Configure Border Router 1

1. Create a VLAN:

```
Switch(config)# vlan 50
Switch(config-vlan)# exit
```

2. Configure the ports on border router 1 as follows:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 50
Switch(config-if)# aggregation-group 5 mode active
Switch(config-if)# exit
```

Configure Border Router 2

1. Create a VLAN:

```
Switch(config)# vlan 60
Switch(config-vlan)# exit
```

2. Configure the ports on border router 2 as follows:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 60
Switch(config-if)# aggregation-group 6 mode active
Switch(config-if)# exit
```

Task 2: Configure switches in the Layer 2 region

Given the following:

- ISL ports on switches A - F: ports 1, 2
- Ports connecting switches A and B to the Layer 2/3 border routers: ports 5, 6
- Ports on switches A and B connecting to switches C and D: ports 10, 11
- Ports on switches C and D connecting to switches A and B: ports 10, 11
- Ports on switch B connecting to switch E: ports 15, 16
- Ports on switch B connecting to switch F: ports 17, 18
- Ports on switch E connecting to switch B: ports 15, 16
- Ports on switch F connecting to switch B: ports 17, 18
- Port on switches C and D connecting to switch G: port 3
- Port on switches E and F connecting to server: port 3

Configuring Switch A

1. Configure the vLAG tier ID and enable vLAG globally:

```
Switch(config)# vlag tier-id 10
Switch(config)# vlag enable
```

2. Configure the ISL ports on Switch A:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if-range)# bridge-port mode trunk
Switch(config-if-range)# bridge-port trunk allowed vlan 30,50,60,100
Switch(config-if-range)# aggregation-group 10 mode active
Switch(config-if-range)# exit
```

3. Associate the LAG to the ISL:

```
Switch(config)# vlag isl port-aggregation 10
```

4. Create VLANs 30, 50, 60 and 100:

```
Switch(config)# vlan 30,50,60,100
Switch(config-vlan)# exit
```

5. Configure the ports on Switch A connecting to the Layer 2/3 border routers:

```
Switch(config)# interface ethernet 1/5
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 50
Switch(config-if)# aggregation-group 5 mode active
Switch(config-if)# exit

Switch(config)# interface ethernet 1/6
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 60
Switch(config-if)# aggregation-group 6 mode active
Switch(config-if)# exit
```

6. Configure the ports on switch A connecting to switches C and D:

```
Switch(config)# interface ethernet 1/10-11
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 30
Switch(config-if)# aggregation-group 3 mode active
Switch(config-if)# exit
```

7. Associate the previously configured ports with their respective vLAGs:

```
Switch(config)# vlag instance 3 port-aggregation 3
Switch(config)# vlag instance 3 enable

Switch(config)# vlag instance 5 port-aggregation 5
Switch(config)# vlag instance 5 enable

Switch(config)# vlag instance 6 port-aggregation 6
Switch(config)# vlag instance 6 enable
```

Configuring Switch B

1. Configure the vLAG tier ID and enable vLAG globally:

```
Switch(config)# vlag tier-id 10
Switch(config)# vlag enable
```

2. Configure the ISL ports on Switch B:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if-range)# bridge-port mode trunk
Switch(config-if-range)# bridge-port trunk allowed vlan 30,40,50,60,100
Switch(config-if-range)# aggregation-group 10 mode active
Switch(config-if-range)# exit
```

3. Associate the LAG to the ISL:

```
Switch(config)# vlag isl port-aggregation 10
```

4. Create VLANs 30, 40, 50, 60 and 100:

```
Switch(config)# vlan 30,40,50,60,100
Switch(config-vlan)# exit
```

5. Configure the ports on Switch B connecting to the Layer 2/3 border routers:

```
Switch(config)# interface ethernet 1/5
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 50
Switch(config-if)# aggregation-group 5 mode active
Switch(config-if)# exit

Switch(config)# interface ethernet 1/6
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 60
Switch(config-if)# aggregation-group 6 mode active
Switch(config-if)# exit
```

6. Configure the ports on switch B connecting to switches C and D:

```
Switch(config)# interface ethernet 1/10-11
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 30
Switch(config-if)# aggregation-group 3 mode active
Switch(config-if)# exit
```

7. Configure the ports on switch B connecting to switches E and F:

```
Switch(config)# interface ethernet 1/15-18
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 40
Switch(config-if)# aggregation-group 4 mode active
Switch(config-if)# exit
```

8. Associate the previously configured ports with their respective vLAGs:

```
Switch(config)# vlag instance 3 port-aggregation 3
Switch(config)# vlag instance 3 enable

Switch(config)# vlag instance 4 port-aggregation 4
Switch(config)# vlag instance 4 enable

Switch(config)# vlag instance 5 port-aggregation 5
Switch(config)# vlag instance 5 enable

Switch(config)# vlag instance 6 port-aggregation 6
Switch(config)# vlag instance 6 enable
```

Configuring Switches C and D

1. Configure the vLAG tier ID and enable vLAG globally:

```
Switch(config)# vlag tier-id 20
Switch(config)# vlag enable
```

2. Configure the ISL ports on Switch C:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if-range)# bridge-port mode trunk
Switch(config-if-range)# bridge-port trunk allowed vlan 10,30,100
Switch(config-if-range)# aggregation-group 10 mode active
Switch(config-if-range)# exit
```

3. Associate the LAG to the ISL:

```
Switch(config)# vlag isl port-aggregation 10
```

4. Create VLANs 10, 30 and 100:

```
Switch(config)# vlan 10,30,100
Switch(config-vlan)# exit
```

5. Configure the ports on Switch C connecting to switches A and B:

```
Switch(config)# interface ethernet 1/10-11
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 30
Switch(config-if)# aggregation-group 3 mode active
Switch(config-if)# exit
```

6. Configure the ports on switch C connecting to switch G:

```
Switch(config)# interface ethernet 1/3
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 10
Switch(config-if)# aggregation-group 1 mode active
Switch(config-if)# exit
```

7. Associate the previously configured ports with their respective vLAGs:

```
Switch(config)# vlag instance 1 port-aggregation 1
Switch(config)# vlag instance 1 enable

Switch(config)# vlag instance 3 port-aggregation 3
Switch(config)# vlag instance 3 enable
```

8. Repeat these steps for Switch D.

Configuring Switch E

1. Configure the vLAG tier ID and enable vLAG globally:

```
Switch(config)# vlag tier-id 30
Switch(config)# vlag enable
```

2. Configure the ISL ports on Switch E:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if-range)# bridge-port mode trunk
Switch(config-if-range)# bridge-port trunk allowed vlan 20,40,100
Switch(config-if-range)# aggregation-group 10 mode active
Switch(config-if-range)# exit
```

3. Associate the LAG to the ISL:

```
Switch(config)# vlag isl port-aggregation 10
```

4. Create VLANs 20, 40 and 100:

```
Switch(config)# vlan 20,40,100
Switch(config-vlan)# exit
```

5. Configure the ports on Switch E connecting to switch B:

```
Switch(config)# interface ethernet 1/15-16
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 40
Switch(config-if)# aggregation-group 4 mode active
Switch(config-if)# exit
```

6. Configure the ports on switch E connecting to the server:

```
Switch(config)# interface ethernet 1/3
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 20
Switch(config-if)# aggregation-group 2 mode active
Switch(config-if)# exit
```

7. Associate the previously configured ports with their respective vLAGs:

```
Switch(config)# vlag instance 2 port-aggregation 2
Switch(config)# vlag instance 2 enable

Switch(config)# vlag instance 4 port-aggregation 4
Switch(config)# vlag instance 4 enable
```

Configuring Switch F

1. Configure the vLAG tier ID and enable vLAG globally:

```
Switch(config)# vlag tier-id 30
Switch(config)# vlag enable
```

2. Configure the ISL ports on Switch F:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if-range)# bridge-port mode trunk
Switch(config-if-range)# bridge-port trunk allowed vlan 20,40,100
Switch(config-if-range)# aggregation-group 10 mode active
Switch(config-if-range)# exit
```

3. Associate the LAG to the ISL:

```
Switch(config)# vlag isl port-aggregation 10
```

4. Create VLANs 20, 40 and 100:

```
Switch(config)# vlan 20,40,100
Switch(config-vlan)# exit
```

5. Configure the ports on Switch E connecting to switch B:

```
Switch(config)# interface ethernet 1/17-18
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 40
Switch(config-if)# aggregation-group 4 mode active
Switch(config-if)# exit
```

6. Configure the ports on switch E connecting to the server:

```
Switch(config)# interface ethernet 1/3
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 20
Switch(config-if)# aggregation-group 2 mode active
Switch(config-if)# exit
```

7. Associate the previously configured ports with their respective vLAGs:

```
Switch(config)# vlag instance 2 port-aggregation 2
Switch(config)# vlag instance 2 enable

Switch(config)# vlag instance 4 port-aggregation 4
Switch(config)# vlag instance 4 enable
```

Chapter 14. Quality of Service

Quality of Service features allow you to allocate network resources to mission-critical applications at the expense of applications that are less sensitive to such factors as time delays or network congestion. You can configure your network to prioritize specific types of traffic, ensuring that each type receives the appropriate Quality of Service (QoS) level.

The following topics are discussed in this section:

QoS Overview

QoS helps you allocate guaranteed bandwidth to critical applications and limit bandwidth for less critical applications. Applications such as video and voice must have a specific amount of bandwidth to work correctly; using QoS, you can provide that bandwidth when necessary. Also, you can put a high priority on applications that are sensitive to timing out or that cannot tolerate delay, by assigning their traffic to a high-priority queue.

By assigning QoS levels to traffic flows on your network, you can ensure that network resources are allocated where they are needed most. QoS features allow you to prioritize network traffic, thereby providing better service for selected applications.

The basic QoS model works as follows:

- Classify traffic:
  - Match the DSCP or IP Precedence value
  - Match the IP RTP priority
  - Match the protocol (ARP, DHCP, IS-IS etc) value
  - Match the 802.1p priority value
  - Match the ACL filter parameters
- Perform actions:
  - Define the bandwidth and burst parameters
  - Select the actions to perform on in-profile and out-of-profile traffic
  - Mark the 802.1p Priority and DSCP or IP Precedence values
  - Set the QoS group (with or without re-marking)
- Queue and schedule traffic:
  - Place the packets in one of the COS queues
  - Schedule transmission based on the COS queue
  - Configure Traffic Shaping
  - Define a guaranteed bandwidth
  - Configure Strict Priority queuing

Class Maps

A *class map* is a named object that represents a class of traffic. In the class map, you specify a set of match criteria for classifying the packets. You can then reference class maps in policy maps.

You define the following class and policy maps types when you create them:

- *control plane (CoPP)*: receives packets that are necessary to configure the switch hardware and for remote switch management
- *qos*: defines a QoS class map that is used to match packets to a specified class
- *queuing*: defines a queuing class map that is used to match packets to a specified class

To configure a class map, use one of the following commands:

```
Switch(config)# class-map type control-plane match-any <CoPP class map name>
Switch(config-cmap-control-plane)#
```

```
Switch(config)# class-map type qos {match-all|match-any} <class map name>
Switch(config-cmap-qos)#
```

```
Switch(config)# class-map type queuing match-any <queue name>
Switch(config-cmap-que)#
```

QoS Classification Types

You can classify traffic by matching packets based on predefined classification criteria, as follows:

- Access Control List (ACL)
- Class of Service (CoS)
- DiffServ Code Point (DSCP)
- UDP or TCP port number
- Precedence
- Protocol

Using ACL Filters

Access Control Lists (ACLs) are filters that allow you to classify and segment traffic, so you can provide different levels of service to different traffic types. Each filter defines the conditions that must match for inclusion in the filter, and also the actions that are performed when a match is made.

To set an QoS ACL filter, use the following command:

```
Switch(config)# class-map type qos {match-all|match-any} <class map name>
Switch(config-cmap-qos)# match [not] access-group name <ACL name>
```

The switch allows you to classify packets based on various parameters. For example:

- Ethernet: source MAC, destination MAC, VLAN number/mask, ethernet type, priority
- IPv4: source IP address/mask, destination address/mask, type of service, IP protocol number
- TCP/UPD: source port, destination port, TCP flag
- Packet format

For ACL details, see Chapter 7, "Access Control Lists".

Summary of QoS Actions

Actions determine how the traffic is treated. The switch QoS actions include the following:

- Re-mark a new DSCP or IP Precedence
- Re-mark the 802.1p field
- Set the QOS group

Using Class of Service Filters

You can classify traffic based on the class of service (CoS). To do that, use the following command:
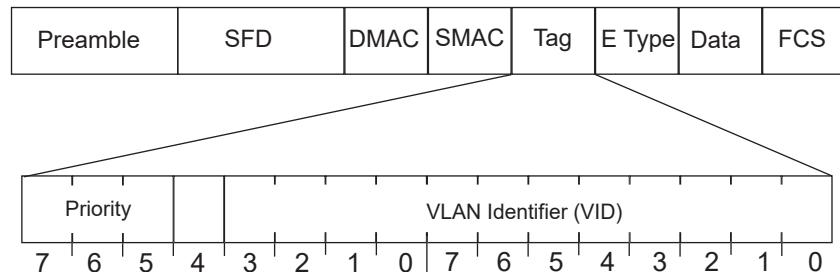
```
Switch(config-cmap-qos)# match [not] cos <CoS value (0-7)>
```

Using 802.1p Priority to Provide QoS

The switch provides Quality of Service functions based on the priority bits in a packet's VLAN header (the priority bits are defined by the 802.1p standard within the IEEE 802.1Q VLAN header). The 802.1p bits, if present in the packet, specify the priority to be given to the packet during forwarding. Packets with a numerically higher (non-zero) priority are given forwarding preference over packets with lower priority values.

The IEEE 802.1p standard uses eight levels of priority (0-7). Priority 7 is assigned to highest priority network traffic, such as OSPF or RIP routing table updates. Priorities 5-6 are assigned to delay-sensitive applications such as voice and video, while lower priorities are assigned to standard applications. A value of 0 (zero) indicates a "best effort" traffic prioritization and this is the default when traffic priority has not been configured on your network. The switch can filter packets based on the 802.1p values.

Figure 9. Layer 2 802.1q/802.1p VLAN tagged packet

Preamble	SFD	DMAC	SMAC	Tag	E Type	Data	FCS

Priority		VLAN Identifier (VID)
7 6 5	4	3 2 1 0 7 6 5 4 3 2 1 0

Ingress packets receive a priority value, as follows:

- **Tagged packets**—switch reads the 802.1p priority in the VLAN tag

- **Untagged packets**—switch tags the packet and assigns an 802.1p priority value, based on the port's default 802.1p priority

Egress packets are placed in a COS queue based on the priority value and scheduled for transmission based on the scheduling weight of the COS queue. Higher COS queue numbers provide forwarding precedence.

Using DiffServ Code Point (DSCP) Filters

The switch uses the Differentiated Services (DiffServ) architecture to provide QoS functions. DiffServ is described in IETF RFCs 2474 and 2475.
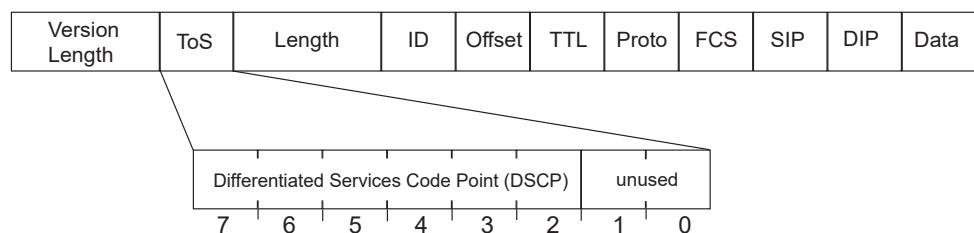
The six most significant bits in the Type of Service (ToS) byte of the IP header are defined as DiffServ Code Points (DSCP). Packets are marked with a certain value depending on the type of treatment the packet must receive from the network device. DSCP is a measure of the Quality of Service (QoS) level of the packet.

The switch can classify traffic by reading the DiffServ Code Point (DSCP) or IEEE 802.1p priority value, or by using filters to match specific criteria. When network traffic attributes match those specified in a traffic pattern, the policy instructs the switch to perform specified actions on each packet that passes through it. The packets are assigned to different Class of Service (CoS) queues and scheduled for transmission.

Differentiated Services Concepts

To differentiate between traffic flows, packets can be classified by their DSCP value. The Differentiated Services (DS) field in the IP header is an octet and the first six bits, called the DS Code Point (DSCP), can provide QoS functions. Each packet carries its own QoS state in the DSCP. There are 64 possible DSCP values (0-63).

Figure 10. Layer 3 IPv4 packet

Version Length	ToS	Length	ID	Offset	TTL	Proto	FCS	SIP	DIP	Data

Differentiated Services Code Point (DSCP)	unused
7 6 5 4 3 2	1 0

The switch can perform the following actions to the DSCP:

- Read the DSCP value of ingress packets

- Re-mark the DSCP value to a new value

- Map the DSCP value to a Class of Service queue (COSq)

The switch can use the DSCP value to direct traffic prioritization.

With DiffServ you can establish policies to direct traffic. A policy is a traffic-controlling mechanism that monitors the characteristics of the traffic (for example, its source, destination, and protocol) and performs a controlling action on the traffic when certain characteristics are matched.

Per Hop Behavior

The DSCP value determines the Per Hop Behavior (PHB) of each packet. The PHB is the forwarding treatment given to packets at each hop. QoS policies are built by applying a set of rules to packets, based on the DSCP value, as they hop through the network.

The default settings are based on the following standard PHBs, as defined in the IEEE standards:

- Expedited Forwarding (EF)—This PHB has the highest egress priority and lowest drop precedence level. EF traffic is forwarded ahead of all other traffic. EF PHB is described in RFC 2598.

- Assured Forwarding (AF)—This PHB contains four service levels, each with a different drop precedence, as shown in the following table. Routers use drop precedence to determine which packets to discard last when the network becomes congested. AF PHB is described in RFC 2597.

Table 23. *Assured Forwarding service levels*

Drop Precedence	Class 1	Class 2	Class 3	Class 4
Low	AF11 (DSCP 10)	AF21 (DSCP 18)	AF31 (DSCP 26)	AF41 (DSCP 34)
Medium	AF12 (DSCP 12)	AF22 (DSCP 20)	AF32 (DSCP 28)	AF42 (DSCP 36)
High	AF13 (DSCP 14)	AF23 (DSCP 22)	AF33 (DSCP 30)	AF43 (DSCP 38)

- Class Selector (CS)—This PHB has eight priority classes, with CS7 representing the highest priority and CS0 representing the lowest priority, as shown in the following table. CS PHB is described in RFC 2474.

Table 24. *Class Selector priority classes*

Priority	Class Selector	DSCP
Highest	CS7	56
	CS6	48
	CS5	40
	CS4	32
	CS3	24
	CS2	16
	CS1	8
Lowest	CS0	0

QoS Levels

Table 25 shows the default service levels provided by the switch, listed from highest to lowest importance:

Table 25. *Default QoS Service Levels*

Service Level	Default PHB	802.1p Priority
Critical	CS7	7
Network Control	CS6	6
Premium	EF, CS5	5
Platinum	AF41, AF42, AF43, CS4	4
Gold	AF31, AF32, AF33, CS3	3
Silver	AF21, AF22, AF23, CS2	2
Bronze	AF11, AF12, AF13, CS1	1
Standard	DF, CS0	0

Using TCP/UDP Port Filters

You can define the TCP/UDP port used by Real-time Transport Protocol (RTP) processes as the classification criterion.

To apply this filter, use the following command:

```
Switch(config-cmap-qos)# match [not] ip rtp <TCP port or UDP port>
```

Using Precedence Filters

You can define the precedence as the classification criterion. The precedence value is from 0 to 7 and it can be one of the following:

- `routine`: Routine precedence (0)

- `priority`: Priority precedence (1)

- `immediate`: Immediate precedence (2)

- `flash`: Flash precedence (3)

- `flash-override`: Flash override precedence (4)

- `critical`: Critical precedence (5)

- `internet`: Internetwork control precedence (6)

- `network`: Network control precedence (7)

To apply this filter, use the following command:

```
Switch(config-cmap-qos)# match [not] precedence <precedence value (0-7)>
```

Using Protocol Filters

To define a protocol as a QoS filter, use the following command:

```
Switch(config-cmap-qos)# match [not] protocol <protocol name>
```

where *protocol name* is one of the following:

- `arp` - Address Resolution Protocol

- `bridging` - Bridging

- `cdp` - CISCO Discovery Protocol

- `clns` - Connectionless Network Service

- `clns-es` - CLNS End Systems

- `clns-is` - CLNS Intermediate Systems

- `dhcp` - Dynamic Host Configuration

- `isis` - Intermediate System to Intermediate System

- `ldp` - Label Distribution Protocol

- `netbios` - NetBIOS extended user interface

Queuing Classification Types

The switch has the following eight predefined queues:

- `1p7q1t-out-q-default` (queue 0 or default queue)
- `1p7q1t-out-pq1` (queue 1 or priority queue)
- `1p7q1t-out-q2` (queue 2)
- `1p7q1t-out-q3` (queue 3)
- `1p7q1t-out-q4` (queue 4)
- `1p7q1t-out-q5` (queue 5)
- `1p7q1t-out-q6` (queue 6)
- `1p7q1t-out-q7` (queue 7)

`1p7q1t` refers to how many priority queues, standard queues, and thresholds are supported. In this case, there are:

- 1 priority queue
- 7 standard queues
- 1 threshold

To apply this filter, use the following command:

```
Switch(config)# class-map type queuing match-any <queue name>
```

You can filter packets to different queues based on a CoS or QoS group.

Class Map Configuration Examples

Following are two basic configuration examples for QoS and queueing class maps.

QoS Class Map Configuration Example

To configure a QoS class map, follow these steps:

1. Create a new QoS class map:

```
Switch(config)# class-map type qos cmap-qos-01
Switch(config-cmap-qos)#
```

2. Define the classification criteria (multiple ones can be configured):

```
Switch(config-cmap-qos)# match protocol arp
Switch(config-cmap-qos)# match precedence 3
```

3. Verify the QoS class map configuration:

```
Switch# display class-map type qos

  Type qos class-maps
  ==================
      class-map type qos match-any class-default

      class-map type qos cmap-qos-01
         match precedence 3
         match protocol arp
```

Queueing Class Map Configuration Example

To configure a queueing class map, follow these steps:

1. Create a new queueing class map:

```
Switch(config)# class-map type queuing match-any 1p7q1t-out-q2
Switch(config-cmap-que)#
```

2. Define the classification criteria (multiple ones can be configured):

```
Switch(config-cmap-que)# match cos 5
Switch(config-cmap-que)# match qos-group 2
```

3. Verify the QoS class map configuration:

```
Switch# display class-map type queueing

  Type queuing class-maps
  =====================
      class-map type queuing match-any 1p7q1t-out-q4

      class-map type queuing match-any 1p7q1t-out-q2
         match qos-group 2
         match cos 5
...
```

Policy Maps

A *policy map* is a named object that represents a set of policies that are to be applied to a set of traffic classes, such as limiting the bandwidth or dropping a packet.

The following predefined policy maps are used as default service policies:

- `control-plane`: Control Plane Protection (CoPP)
- `qos`: Quality of Service (QoS)
- `queuing`

To set a policy map, use the following command:

```
Switch(config)# policy-map type {control-plane|qos|queing} <policy-map name>
```

Ingress Policing

The switch can use ingress policing to monitor the data rates for a particular class of traffic. When the data rate exceeds user-configured values, the switch drops packets immediately.

Defining Single-Rate and Dual-Rate Policers

You can define single-rate and dual-rate policers. While single-rate policers monitor the committed information rate (CIR) of traffic, dual-rate policers monitor both CIR and peak information rates (PIR) of traffic. The system also monitors associated burst sizes.

Only one action can be configured for each condition.

To configure polices, use the following command:

```
Switch(config-pmap-c-qos)# police [cir] {<committed-rate> [<data-rate>]|percent
<CIR-link-percent>} [bc <committed-burst-rate> [<link-speed>]] [pir] {<peak-rate>
[<data-rate>]|percent <CIR-link-percent>} [be <peak-burst-rate> [<link-speed>]] [conform
<conform actions>] [exceed <exceed actions>] [violate <violate actions>]
```

where:

- `cir` specifies the use of the committed information rate, or desired bandwidth, as a bit rate or a percentage of the link rate.
- *committed-rate* can range from 8,000 bps to the maximum rate of the link, in steps that are multiples of 8,000 bps.
- *data-rate* can have one of the following units:
  - `bps:` bits per second
  - `kbps:` 1,000 bits per second
  - `mbps:` 1,000,000 bits per second
  - `Gbps:` 1,000,000,000 bits per second
- `percent` specifies the rate as a percentage of the interface bandwidth.
- `bc` indicates how much the CIR can be exceeded as a bit rate or for how long can the CIR be exceeded.

- *committed-burst-rate* has the following options:
  - o `bytes`: bytes
  - o `kbytes`: 1,000 bytes
  - o `mbytes`: 1,000,000 bytes
  - o `ms`: milliseconds
  - o `us`: microseconds
- `pir` sets a peak information rate value.
- `be` indicates how much the PIR can be exceeded as a bit rate or for how long can the PIR be exceeded.
- `conform` specifies to take action if the traffic data rate is within bounds. The following actions are allowed:

Table 26. *Conform Actions*

Conform Actions	Description
transmit	Send the packet. This action is applicable only when the packet conforms to the parameters.
set-prec-transmit	Set the IP precedence field to a specified value and send the packet.
set-dscp-transmit	Set the Differentiated Service Code Point (DSCP) field to a specified value and send the packet.
set-cos-transmit	Set the class of service (CoS) field to a specified value and send the packet.

The default action is `transmit`.

- `exceed` specifies to take action if traffic data rate is exceeded.
- `violate` specifies to take action if traffic data rate violates the set rate values.

The following actions are available for both `exceed` and `violate` scenarios:

Table 27. *Exceed and Violate Actions*

Exceed and Violate Actions	Description
drop	Drop the packet. This is only available when the packet exceeds or violates the parameters.
set-dscp-transmit	Set the Differentiated Service Code Point (DSCP) field to a specified value and send the packet.
set-cos-transmit	Set the class of service (CoS) field to a specified value and send the packet.

Marking

Marking is a method that you use to modify the QoS fields of the incoming and outgoing packets.

To do so, use the following command:

```
Switch(config-pmap-c-qos)# apply {cos <0-7>|dscp <0-63>|precedence <0-7>|
|qos-group <0-7>}
```

Queuing Policing

Traffic queuing is a mechanism allowing the switch to order the packets and it applies to output data. Device modules can support multiple queues, which you can use to control the sequencing of packets in different traffic classes.

Bandwidth

Bandwidth on egress queues can be configured to allocate a minimum percentage of the interface bandwidth to a queue. To set the bandwidth, use the following command:

```
Switch(config-pmap-c-que)# bandwidth {<rate> {bps|kbps|mbps|gbps}|percent
<percent value>}
```

You can configure the bandwidth by specifying its bit rate, or as a percentage of the underlying link rate.

You can also configure the data rate as a percentage of the bandwidth that is not allocated to other classes by using the following command:

```
Switch(config-pmap-c-que)# bandwidth remaining percent <percent value>
```

Shaping

You can configure shaping on an egress queue to impose a maximum rate on it. A data rate may be configured by either using a bit rate, or as a percentage of the underlying interface link rate.

To configure shaping, use the following command:

```
Switch(config-pmap-c-que)# shape [average] {<rate> {bps|kbps|mbps|gbps}|
percent <percent value>}
```

Priority

You can configure only one level of priority on an egress priority queue. To do so, use the following command:

```
Switch(config-pmap-c-que)# priority [level 1]
```

Policy Map Configuration Examples

Following are two basic examples of configuring QoS and queueing policy maps.

QoS Policy Map Configuration Example

To configure a QoS policy map, follow these steps:

1. Create a new QoS policy map:

```
Switch(config)# policy-map type qos pmap-qos-01
Switch(config-pmap-qos)#
```

2. Select an existing QoS class map:

```
Switch(config-pmap-qos)# class type qos cmap-qos-01
Switch(config-pmap-c-qos)#
```

3. Configure the ingress policing or marking:

```
Switch(config-pmap-c-qos)# police cir 2 mbps conform transmit
Switch(config-pmap-c-qos)# apply cos 3
```

4. Verify the QoS policy map configuration:

```
Switch# display policy-map type qos

Type qos policy-maps
====================

policy-map type qos pmap-qos-01
 class type qos cmap-qos-01
  apply cos 3
  police cir 2 mbps conform transmit
```

Queuing Policy Map Configuration Example

To configure a queueing policy map, follow these steps:

1. Create a new queueing policy map:

```
Switch(config)# policy-map type queuing pmap-que-01
Switch(config-pmap-que)#
```

2. Select an existing queueing class map:

```
Switch(config-pmap-que)# class type queuing 1p7q1t-out-q3
Switch(config-pmap-c-que)#
```

3. Configure the queueing policing:

```
Switch(config-pmap-c-que)# bandwidth percent 25
```

4. Verify the queueing policy map configuration:

```
Switch# display policy-map type queueing

Type queuing policy-maps
========================

policy-map type queuing default-out-policy
 class type queuing 1p7q1t-out-pq1
  priority level 1
 class type queuing 1p7q1t-out-q2
 class type queuing 1p7q1t-out-q3
 class type queuing 1p7q1t-out-q-default
  bandwidth remaining percent 25

policy-map type queuing pmap-que-01
 class type queuing 1p7q1t-out-q2
 class type queuing 1p7q1t-out-q3
  random-detect  minimum-threshold percent 5 maximum-threshold percent 35
  bandwidth percent 25
```

Control Plane Protection

Control Plane Protection (CoPP) receives packets that are required for the internal protocol state machines. This type of traffic is usually received at low rate. However, in some situations such as DOS attacks, the switch may receive this traffic at a high rate. If the control plane protocols are unable to process the high rate of traffic, the switch may become unstable.

The control plane receives packets that are channeled through protocol-specific packet queues. Multiple protocols can be channeled through a common packet queue. However, one protocol cannot be channeled through multiple packet queues. These packet queues are applicable only to the packets received by the software and does not impact the regular switching or routing traffic. Packet queues with a higher number have higher priority.

You can configure the bandwidth for each packet queue. Protocols that share a packet queue will also share the bandwidth.

Given below are the commands to configure CoPP:

1. Enter Control Plane Configuration mode by using the following command:

```
Switch(config)# control-plane
```

2. To apply the control plane service policy, use the command below:

```
Switch(config-cp)# service-policy input copp-system-policy
```

3. To modify a control-plane policy-map, use the command below:

```
Switch(config)# [no] policy-map type control-plane <policy-map name>
```

4. To modify a control-plane class map that is used to match packets to a specified class, use the command below:

```
Switch(config)# [no] class-map type control-plane match-any <class map name>
```

5. To view CoPP switch configuration, use the following commands:

```
Switch(config)# display class-map type control-plane
Switch(config)# display policy-map type control-plane
Switch(config)# display policy-map interface control-plane
```

Control Plane Configuration Examples

To configure a CoPP class map, use the following steps:

1. Since new CoPPs cannot be created, modify one of the predefined CoPP class maps. To view the current CoPP class maps, use the following command:

```
Switch(config)# display class-map type control-plane

  Type control plane class-maps
  =============================
      class-map match-any copp-s-lacp

      class-map match-any copp-s-default

      class-map match-any copp-s-bfd

      class-map match-any copp-s-arpresponse

      class-map match-any copp-s-arprequest

      class-map match-any copp-s-authentication
            QOS-ACCESS-LIST-NAME: copp-system-acl-authentication
...
```

2. Enter configuration mode for one of the above CoPP class maps:

```
Switch(config)# class-map type control-plane match-any copp-s-bfd
Switch(config-cmap-control-plane)#
```

3. Configure the classification criteria for the CoPP class map:

```
Switch(config-cmap-control-plane)# match access-group name myACL
```

Note: You cannot define multiple classification criteria for a single CoPP class map.

4. Verify the configuration:

```
Switch# display class-map type control-plane

  Type control plane class-maps
  =============================
      class-map match-any copp-s-lacp

      class-map match-any copp-s-default

      class-map match-any copp-s-bfd
            QOS-ACCESS-LIST-NAME: myACL

      class-map match-any copp-s-arpresponse

      class-map match-any copp-s-arprequest

      class-map match-any copp-s-authentication
            QOS-ACCESS-LIST-NAME: copp-system-acl-authentication
...
```

To configure a CoPP policy map, use the following steps:

1. Since new CoPPs cannot be created, modify the predefined CoPP policy map
(copp-system-policy). To view the current CoPP policy maps, use the following
command:

```
Switch(config)# display policy-map type control-plane

Type control-plane policy-maps
==============================

policy-map type control-plane copp-system-policy
 class type control-plane copp-s-default
  police pps 20000
 class type control-plane copp-s-ntp
  police pps 200
 class type control-plane copp-s-arprequest
  police pps 500
 class type control-plane copp-s-nd
  police pps 500
...
```

2. Enter configuration mode for one of the above CoPP policy maps:

```
Switch(config)# policy-map type control-plane match-any copp-system-policy
Switch(config-pmap-control-plane)#
```

3. Enter configuration mode for one of the above CoPP class maps:

```
Switch(config-pmap-control-plane)# class type control-plane copp-s-ntp
Switch(config-pmap-c-control-plane)#
```

4. Configure the number of packet per second processed by the switch for the current
class map:

```
Switch(config-pmap-c-control-plane)# police pps 100
```

Note: The switch will not process more than the configured number of packets per
second for each type of traffic. The excess packets will be dropped.

5. Verify the configuration:

```
Switch# display policy-map type control-plane

Type control-plane policy-maps
==============================

policy-map type control-plane copp-system-policy
 class type control-plane copp-s-default
  police pps 20000
 class type control-plane copp-s-ntp
  police pps 100
 class type control-plane copp-s-arprequest
  police pps 500
 class type control-plane copp-s-nd
  police pps 500
...
```

WRED

Weighted Random Early Detection (WRED) is a congestion avoidance algorithm that helps prevent a TCP collapse, where a congested port indiscriminately drops packets from all sessions. The transmitting hosts wait to retransmit, resulting in a dramatic drop in throughput. Often times, this TCP collapse repeats in a cycle, which results in a saw-tooth pattern of throughput. WRED selectively drops packets before the queue gets full, allowing majority of the traffic to flow smoothly.

WRED discards packets based on the CoS queues. Packets are discarded in order of their priority, from lowest to highest.

WRED calculates the average size of the queue. If the average queue size is below a configured minimum threshold, then an arriving packet is immediately queued. If the average queue size is between the minimum and maximum threshold, then an arriving packet is either queued or discarded depending on the configured drop probability. If the average queue size is over the maximum threshold, then an arriving packets is immediately discarded.

For implementing WRED, you must define a profile with a minimum threshold, a maximum threshold, and a maximum drop probability. The profiles can be defined on a port or a CoS.

Configuring WRED

WRED can be configured only on physical ports and not on LAGs. WRED is applicable only to unicast traffic.

To configure WRED, use the following command:

```
Switch(config)# policy-map type queuing <policy map name>
Switch(config-pmap-que)# class type queuing <class map name>
Switch(config-pmap-c-que)# random-detect {[minimum-threshold] {<min-threshold>
[packets|bytes|kbytes|mbytes|ms|us]|percent <min-percent>}
[maximum-threshold] {<max-threshold> [packets|bytes|kbytes|mbytes|ms|us]|
|percent <max-percent>} {drop-probability <0-100>}
```

WRED Configuration Example

Follow these steps to enable WRED and configure a global and/or port-level profile. If you configure global and port-level profiles, WRED uses the port-level profile to make transmit/discard decisions when experiencing traffic congestion.

1. Configure a policy-map of type queuing:

```
Switch(config)# policy-map type queuing policy-1
```

2. Assign a system defined class-map to the policy-map configured at step 1:

```
Switch(config-pmap-que)# class type queuing 1p7q4t-out-pq2
```

3. Enable WRED for the class-map by specifying (in bytes) the minimum and the maximum thresholds, and the drop probability when the maximum threshold is reached:

```
Switch(config-pmap-c-que)# random-detect [minimum-threshold] <min-threshold>
bytes [maximum-threshold] <max-threshold> bytes drop-probability <0-100>
```

4. Use the following command to view global WRED information:

```
Switch(config)# display policy-map type queuing

Type queuing policy-maps
========================

policy-map type queuing default-out-policy
 class type queuing 1p7q1t-out-pq1
  priority level 1
 class type queuing 1p7q1t-out-q2
 class type queuing 1p7q1t-out-q3
 class type queuing 1p7q1t-out-q-default
  bandwidth remaining percent 25

policy-map type queuing policy_queue1
 class type queuing 1p7q1t-out-q5
  shape average 4 bps
  random-detect  minimum-threshold 12000 packets maximum-threshold 30000
    packets drop-probability 50
 class type queuing 1p7q1t-out-q3
...
```

Use the following command to view port-level WRED information:

```
Switch(config)# display policy-map interface ethernet 1/4 output

Global statistics status : disabled

Ethernet1/4

 Service-policy (queuing) output: policy-1

 Class-map (queuing):  1p7q1t-out-q2 (match any)
   random-detect  minimum-threshold 35000 bytes maximum-threshold 250000 bytes
     drop-probability 50
```

Interface Service Policy

An input QoS policy is a service policy applied to incoming traffic on an Ethernet interface for classification. For type queuing, the output policy is applied to all outgoing traffic that matches the specified class.

Apply an Interface Service Policy

To apply a policy to an interface, use one of the following commands:

```
Switch(config-if)# service-policy [type qos] input <policy-map-name>
```

```
Switch(config-if)# service-policy type queuing output <policy-map-name>
```

Interface Service Policy Limitations

- Service policy is applied only to ingress traffic for type QoS and to egress traffic for type queuing.
- The device restricts QoS policies to one per interface per direction (ingress or egress) for each of the policy types QoS and queuing.
- Queuing policy maps are not support for Aggregator Port.

Microburst Detection

Microbursts are short peaks in data traffic that manifest as a sudden increase in the number of data packets transmitted over a specific millisecond-level time frame, potentially overwhelming network buffers. Microburst detection allows users to analyze and mitigate microburst-related incidents, thus preventing network congestion.

Microburst Detection can be enabled or disabled on each switch interface. To enabled it on an interface, use the following command (for this example, ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# microburst-detection enable threshold <threshold value>
```

To disable microburst detection on an interface, use the following command:

```
Switch(config-if)# no microburst-detection enable
```

To configure the polling interval (in milliseconds) used by microburst detection to evaluate traffic burst:

```
Switch(config)# microburst-detection interval <5-5000>
```

Note: By default, microburst detection is disabled.

To see the current microburst statistics, use the following command:

```
Switch# display statistics microburst
```

Below is a basic configuration example for Microburst Detection.

1. Enter Interface Configuration mode and enabled Microburst Detection, choosing a threshold value:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# microburst-detection enable threshold 20
Switch(config-if)# exit
```

2. Configure the polling interval:

```
Switch(config)# microburst-detection interval 100
```

3. See the current microburst statistics:

```
Switch# display statistics microburst
--------------------------------------------------------------------------
Interface     # of uburst  avg size  max size  avg duration  max duration
--------------------------------------------------------------------------
Ethernet1/12            0         0         0             0             0
```

Chapter 15. CEE

This chapter provides conceptual background and configuration examples for using Converged Enhanced Ethernet (CEE) features of the switch, with an emphasis on RoCEv1, RoCEv2, iSCSI solutions. The following topics are addressed in this chapter:

-

    Converged Enhanced Ethernet (CEE) refers to a set of IEEE standards developed primarily to enable FCoE, requiring enhancing the existing Ethernet standards to make them lossless on a per-priority traffic basis, and providing a mechanism to carry converged (LAN/SAN/IPC) traffic on a single physical link. CEE features can also be used in traditional LAN (non-FCoE) networks to provide lossless guarantees on a per-priority basis, and to provide efficient bandwidth allocation.

-

    RDMA over Converged Ethernet (RoCE) allows remote direct memory access (RDMA) over an Ethernet network. This provides accelerated communications between applications hosted on clusters of servers and storage arrays, low latency and overall a better performance compared to software based protocols. Internet Small Computer System Interface (iSCSI) protocol is used over RoCe to allow a block-level storage capability similar to Fibre Channel (FC) SAN technology, which is the same system of encapsulating the SCSI protocol within an external "carrier." The difference is that the iSCSI SAN uses Ethernet instead of FC transport technology.

-

    Priority-Based Flow Control (PFC) extends 802.3x standard flow control to enable the switch to pause traffic based on the 802.1p priority value in each packet's VLAN tag. PFC is vital for FCoE/RoCE/iSCSI environments, where RoCE/iSCSI/FCoE traffic must remain lossless and must be paused during congestion, while LAN traffic on the same links is delivered with "best effort" characteristics.

-

    Enhanced Transmission Selection (ETS) provides a method for allocating link bandwidth based on the 802.1p priority value in each packet's VLAN tag. Using ETS, different types of traffic (such as lossless LAN, SAN, and management) that are sensitive to different handling criteria can be configured either for specific bandwidth characteristics, low-latency, or best-effort transmission, despite sharing converged links as in an FCoE environment.

-

    Data Center Bridging Capability Exchange Protocol (DCBX) allows neighboring network devices to exchange information about their capabilities. This is used between CEE-capable devices for the purpose of discovering their peers, negotiating peer configurations, and detecting misconfigurations.

-

    This section provides the following CEE configuration examples:

    o Default configuration

    o Configuration with RoCEv2 and another Biz-critical LAN application or iSCSI

RoCE and iSCSI

RDMA over Converged Ethernet (RoCE) allows remote direct memory access (RDMA) over an Ethernet network. RoCE provides direct memory to memory transfers at the application level without involving the host CPU. Both the transport processing and the memory translation and placement are performed by hardware resulting in dramatically lower latency and higher performance. There are two RoCE versions, RoCEv1 and RoCEv2: RoCEv1 is an Ethernet link layer protocol and hence allows communication between any two hosts in the same Ethernet broadcast domain, while RoCEv2 is designed to allow lossless traffic in the layer 3 network environment.

Internet Small Computer System Interface (iSCSI) is an Internet Protocol (IP)-based storage networking standard for linking data storage facilities. It provides block-level access to storage devices by carrying SCSI commands over a TCP/IP network. iSCSI takes a popular high-performance local storage bus and emulates it over a wide range of networks, creating a storage area network (SAN). Unlike some SAN protocols, iSCSI requires no dedicated cabling; it can be run over existing IP infrastructure. As a result, iSCSI is often seen as a low-cost alternative to Fibre Channel.

RoCE Requirements

The following are required for implementing RoCE using the switches with CNOS 10.6 software:

● An underlying lossless Ethernet network is required for RoCE traffic only to avoid systematic packet drops resulting from resource contention within network switches and adapters.

● CEE must be turned on (see "Turning CEE On or Off" on page 355). When CEE is on, the DCBX, PFC, and ETS features are enabled and configured with default settings. These features may be reconfigured, but must remain enabled for RoCE to function.

Converged Enhanced Ethernet

Converged Enhanced Ethernet (CEE) refers to a set of IEEE standards designed to allow different physical networks with different data handling requirements to be converged together, simplifying management, increasing efficiency and use, and leveraging legacy investments without sacrificing evolutionary growth.

CEE standards were developed primarily to enable Fibre Channel/RoCE/iSCSI traffic to be carried over Ethernet networks. This required enhancing the existing Ethernet standards to make them lossless on a per-priority traffic basis, and to provide a mechanism to carry converged (LAN/SAN/IPC) traffic on a single physical link. Although CEE standards were designed with FCoE in mind, they are not limited to FCoE installations. CEE features can be utilized in traditional LAN (non-FCoE) networks to provide lossless guarantees on a per-priority basis, and to provide efficient bandwidth allocation based on application needs.

Turning CEE On or Off

By default, CEE is turned off. To turn CEE on or off, use the following command:

```
Switch(config)# [no] cee enable
```

CAUTION:
Turning CEE on will automatically change some 802.1p QoS and 802.3x standard flow control settings on the switch. Read the following material carefully to determine whether you will need to take action to reconfigure expected settings.

We recommended that you backup your configuration before turning CEE on. Having the file will allow you to manually re-create the equivalent configuration once CEE is turned on, and will also allow you to recover your prior configuration if you need to turn CEE off.

When CEE is turned on, the following default CEE configuration is automatically triggered if no specific configurations are made to ETS/PFC/DCBX:

- DCBX is enabled on all ports, and ETS/PFC/App-proto are advertised.
- PFC is configured on all ports and `priority-3` is set.
- The default ETS configuration is the following:

```
PGID    BW%   COSq  Priorities
==================
0       10    0      0 1 2
1       0     NA
2       40    2      4 5 6 7
3       50    3      3
4       0     4
5       0     5
6       0     6
7       0     7
15      NA    1
```

- No default application protocol is configured.

Note: You can change the default ETS/PFC/DCBX configuration without enabling CEE, but the configuration will not be effective. The new configuration will be effective only when enabling CEE.

Effects on Link Layer Discovery Protocol

When CEE is turned on, DCBX is turned on and starts sending and receiving DCBX Type-Length-Values(TLVs) on LLDP Protocol Data Units(PDU).

Effects on 802.1p Quality of Service

While CEE is off (the default), the switch allows 802.1p priority values to be used for Quality of Service (QoS) configuration (see page 331). 802.1p QoS default settings are shown in Table 28, but can be changed by the administrator.

When CEE is turned on, 802.1p QoS is replaced by ETS (see "Enhanced Transmission Selection" on page 361). As a result, while CEE is turned on, the 802.1p QoS configuration commands are no longer available on the switch (the menu is restored when CEE is turned off).

In addition, when CEE is turned on, prior 802.1p QoS settings are replaced with new defaults designed for use with ETS priority groups (PGIDs) as shown in Table 28.

Table 28. *802.1p QoS Configuration with CEE On*

PGID	BW%	COSq	Priorities
0	10	0	0, 1, 2
1	0	NA	
2	40	2	4, 5, 6, 7
3	50	3	3
4	0	4	
5	0	5	
6	0	6	
7	0	7	
15	NA	1	

When CEE is on, the default ETS configuration also allocates a portion of link bandwidth to each PGID as shown in Table 29:

Table 29. *Default ETS Bandwidth Allocation*

PGID	Typical Use	Bandwidth
	LAN	10%
	RoCE/iSCSI	50%
	Latency-sensitive LAN	40%

If the prior, non-CEE configuration used 802.1p priority values for different purposes, or does not expect bandwidth allocation as shown in Table 29 on page 356, when CEE is turned on, have the administrator reconfigure ETS settings as appropriate.

It is recommended that a configuration backup be made prior to turning CEE on or off. Viewing the configuration file will allow the administrator to manually re-create the equivalent configuration under the new CEE mode, and will also allow for the recovery of the prior configuration if necessary.

Effects on Flow Control

When CEE is off (the default), 802.3x standard flow control is enabled on all switch ports by default.

When CEE is turned on, standard flow control is disabled on all ports, and in its place, PFC (see "Priority-Based Flow Control" on page 358) is enabled on all ports for 802.1p priority value 3. This default is chosen because priority value 3 is commonly used to identify lossless traffic in a CEE environment and must be guaranteed lossless behavior. PFC is disabled for all other priority values.

Each time CEE is turned off, the prior 802.3x standard flow control settings will be restored (including any previous changes from the defaults).

It is recommended that you make a backup of the configuration before turning CEE on or off. Viewing the configuration file will allow you to manually re-create the equivalent configuration under the new CEE mode, and will also allow for the recovery of the prior configuration if necessary.

When CEE is on, PFC can be enabled only on any two priorities based on the requirement, with the exception of the NE10032, which supports only one PFC priority. If flow control is required on additional priorities on any given port, consider using standard flow control on that port, so that regardless of which priority traffic becomes congested, a flow control frame is generated.

Priority-Based Flow Control

Priority-based Flow Control (PFC) is defined in IEEE 802.1Qbb. PFC extends the IEEE 802.3x standard flow control mechanism. Under standard flow control, when a port becomes busy, the switch manages congestion by pausing all the traffic on the port, regardless of the traffic type. PFC provides more granular flow control, allowing the switch to pause specified types of traffic on the port, while other traffic on the port continues.

PFC pauses traffic based on 802.1p priority values in the VLAN tag. The administrator can assign different priority values to different types of traffic and then enable PFC for up to two specific priority values: priority value 3, and one other. The configuration can be applied globally for all ports on the switch. Then, when traffic congestion occurs on a port (caused when ingress traffic exceeds internal buffer thresholds), only traffic with priority values where PFC is enabled is paused. Traffic with priority values where PFC is disabled proceeds without interruption but may be subject to loss if port ingress buffers become full.

PFC requires CEE to be turned on ("Turning CEE On or Off" on page 355). When CEE is turned on, PFC is enabled on priority value 3 by default. Optionally, the administrator can also enable PFC on one other priority value, providing lossless handling for another traffic type, such as for a business-critical LAN application.

Note: For any given port, only one flow control method can be implemented at any given time: either PFC or standard IEEE 802.3x flow control.

PFC Configuration

PFC requires CEE to be turned on ("Turning CEE On or Off" on page 355). When CEE is turned on, standard flow control is disabled on all ports, and PFC is enabled on all ports for 802.1p priority value 3. This default is chosen because priority value 3 is commonly used to identify RoCE/iSCSI traffic in a CEE environment and must be guaranteed lossless behavior. PFC is disabled for all other priority values by default, but can be enabled for one additional priority value.

PFC configuration can also be used in some mixed environments where traffic with PFC-enabled priority values occurs only on ports connected to CEE devices, and not on any ports connected to non-CEE devices. In such cases, PFC can be configured globally on specific priority values even though not all ports make use them.

PFC has the following characteristics:

- PFC is not restricted to CEE and FCoE networks. In any LAN where traffic is separated into different priorities, PFC can be enabled on priority values for loss-sensitive traffic.

- If you want to enable PFC on a priority, add the priority to a priority group with the same number. For example, if you want to enable PFC on priority 0, you must map priority 0 to PG 0.

- If you want to enable PFC on a priority, do one of the following:
  - Create a separate PG (separate COS Q) (or)
  - Move the priority to the existing PG in which PFC is turned on.

  Option 1 will be more preferred as you have separate Q and separate ETS configuration.
- When configuring ETS and PFC on the switch, perform ETS configuration before performing PFC configuration.
- If two priorities are enabled on a port, the switch sends PFC frames for both priorities, even if only traffic tagged with one of the priorities is being received on that port.
- The administrator can enable or disable PFC on a port-by-port basis.

Notes:

- When using PFC configuration in conjunction with the ETS feature (see "Enhanced Transmission Selection" on page 361), ensure that only pause-tolerant traffic (such as lossless traffic) is assigned priority values where PFC is enabled. Pausing other types of traffic can have adverse effects on LAN applications that expect uninterrupted traffic flow and tolerate dropping packets during congestion.
- The NE10032 only supports one PFC enabled priority. When you enable CEE,priority 3 will be enabled by default, but you can disable PFC on priority 3 and enable PFC on another priority.

PFC Configuration Example

Note: DCBX may be configured to permit sharing PFC configuration with or from external devices. This example assumes that PFC configuration is being performed manually. See "Data Center Bridging Capability Exchange" on page 367 for more information on DCBX.

For example, the following topology is used.

Table 30. *PFC Configuration*

802.1p Priority	Usage	PFC Setting
0-2	LAN	Disabled
3	RoCE	**Enabled**
4	Business-critical LAN	Disabled
others	(not used)	Disabled

In this example, PFC is to facilitate lossless traffic handling for RoCE (priority value 3).

Assuming that CEE is off, the example topology shown in Table 30 can be configured using the following commands:

1. Turn CEE on.

```
Switch(config)# cee enable
```

Note: Turning CEE on will automatically change some 802.1p QoS and 802.3x standard flow control settings and menus (see "Turning CEE On or Off" on page 355).

2. Optionally, set a PFC description:

   Note: PFC is enabled on priority 3 by default.

```
Switch(config)# cee pfc priority 3 description "RoCEv2"
                                                    (Optional description)
```

3. Save the configuration.

Enhanced Transmission Selection

Enhanced Transmission Selection (ETS) is defined in IEEE 802.1Qaz. ETS provides a method for allocating port bandwidth based on 802.1p priority values in the VLAN tag. Using ETS, different amounts of link bandwidth can specified for different traffic types (such as for LAN, SAN, and management).

ETS is an essential component in a CEE environment that carries different types of traffic, each of which is sensitive to different handling criteria, such as Storage Area Networks (SANs) that are sensitive to packet loss, and LAN applications that may be latency-sensitive. In a single converged link, such as when implementing FCoE, ETS allows SAN and LAN traffic to coexist without imposing contrary handling requirements upon each other.

The ETS feature requires CEE to be turned on (see "Turning CEE On or Off" on page 355).

802.1p Priority Values

Under the 802.1p standard, there are eight available priority values, with values numbered 0 through 7, which can be placed in the priority field of the 802.1Q VLAN tag:

16 bits	3 bits	1	12 bits
Tag Protocol ID (0x8100)	Priority	CFI	VLAN ID
0 15	16		32

Servers and other network devices may be configured to assign different priority values to packets belonging to different traffic types (such as SAN and LAN).

ETS uses the assigned 802.1p priority values to identify different traffic types. The various priority values are assigned to priority groups (PGID), and each priority group is assigned a portion of available link bandwidth.

Priorities values within in any specific ETS priority group are expected to have similar traffic handling requirements with respect to latency and loss.

An administrator may assign 802.1p priority values for a variety of purposes. However, when CEE is tugrned on, the switch sets the initial default values for ETS configuration as shown in Table 31.

Table 31. *Default ETS Priority Group*

Typical Traffic Type	802.1p Priority	PGID	Bandwidth Allocation
LAN	0	0	10%
LAN	1	0	10%
LAN	2	0	10%
SAN	3	3	50%

Table 31. *Default ETS Priority Group*

Typical Traffic Type	802.1p Priority	PGID	Bandwidth Allocation
Latency-Sensitive LAN	4	2	40%
Latency-Sensitive LAN	5	2	40%
Latency-Sensitive LAN	6	2	40%
Latency-Sensitive LAN	7	2	40%

In the assignment model shown in Table 31, priorities values 0 through 2 are assigned for regular Ethernet traffic, which has "best effort" transport characteristics.

Because CEE and ETS features are generally associated with FCoE, Priority 3 is typically used to identify FCoE (SAN) traffic.

Priorities 4-7 are typically used for latency sensitive traffic and other important business applications. For example, priority 4 and 5 are often used for video and voice applications such as IPTV, Video on Demand (VoD), and Voice over IP (VoIP). Priority 6 and 7 are often used for traffic characterized with a "must get there" requirement, with priority 7 used for network control which is requires guaranteed delivery to support configuration and maintenance of the network infrastructure.

Priority Groups

For ETS use, each 801.2p priority value is assigned to a priority group which can then be allocated a specific portion of available link bandwidth. To configure a priority group, a priority group must be assigned a priority group ID (PGID), one or more 802.1p priority values, and allocated link bandwidth.

PGID

Each priority group is identified with number (0 through 7, and 15) known as the PGID.

PGID 0 through 7 may each be assigned a portion of the switch's available bandwidth.

PGID 8 through 14 are reserved as per the 802.1Qaz ETS standard.

PGID 15 is a strict priority group. It is generally used for critical traffic, such as network management. Any traffic with priority values assigned to PGID 15 is permitted as much bandwidth as required, up to the maximum available on the switch. After serving PGID 15, any remaining link bandwidth is shared among the other groups, divided according to the configured bandwidth allocation settings.

Make sure all 802.1p priority values assigned to a particular PGID have similar traffic handling requirements. For example, PFC-enabled traffic must not be grouped with non-PFC traffic. Also, traffic of the same general type must be assigned to the same PGID. Splitting one type of traffic into multiple 802.1p priorities, and then assigning those priorities to different PGIDs may result in unexpected network behavior.

Each 802.1p priority value may be assigned to only one PGID. However, each PGID may include multiple priority values. Up to eight PGIDs may be configured at any given time.

Assigning Priority Values to a Priority Group

Each priority group may be configured from its corresponding ETS Priority Group, available using the following command:

```
Switch(config)# cee ets priority-group pgid <group number (0-7, or 15)> priority
    <priority list>
Switch(config)# cee ets priority-group pgid <group number (0-7, or 15)>
description <description>
                                        (Optional description)
```

where *priority list* is one or more 802.1p priority values (with each separated by a comma). For example, to assign priority values 0 through 2:

```
Switch(config)# cee ets priority-group pgid <group number (0-7, or 15)> priority
    0,1,2
Switch(config)# cee ets priority-group pgid <group number (0-7, or 15)>
description "ETS"
                                        (Optional description)
```

Note: Within any specific PGID, the PFC settings (see "Priority-Based Flow Control" on page 358) must be the same (enabled or disabled) for all priority values within the group. If the PFC setting is inconsistent within a PGID, an error is reported when attempting to apply the configuration.

When assigning priority values to a PGID, the specified priority value will be automatically removed from its old group and assigned to the new group when the configuration is applied.

For PGIDs 0 through 7, bandwidth allocation can also be configured through the ETS Priority Group menu. See for "Allocating Bandwidth" on page 363 for details.

Allocating Bandwidth

Follow these guidelines when allocating bandwidth.

Allocated Bandwidth for PGID 0 Through 7

You may allocate a portion of the switch's available bandwidth to PGIDs 0 through 7. Available bandwidth is defined as the amount of link bandwidth that remains after priorities within PGID 15 are serviced (see "Unlimited Bandwidth for PGID 15" on page 364), and assuming that all PGIDs are fully subscribed. If any PGID does not fully consume its allocated bandwidth, the unused portion is made available to the other priority groups.

Priority group bandwidth allocation can be configured using the following command:

```
Switch(config)# cee ets bandwidth-percentage <bandwidth allocation>
```

where *bandwidth allocation* represents the percentage of link bandwidth, specified as a number between 0 and 100, in 1% increments.

The following bandwidth allocation rules apply:

- Bandwidth allocation must be 0% for any PGID that has no assigned 802.1p priority values.
- Any PGID assigned one or more priority values must have a bandwidth allocation greater than 9%.
- Total bandwidth allocation for groups 0 through 7 must equal exactly 100%. Increasing or reducing the bandwidth allocation of any PGID also requires adjusting the allocation of other PGIDs to compensate.

If these conditions are not met, the switch will report an error when applying the configuration.

To achieve a balanced bandwidth allocation among the various priority groups, packets are scheduled according to a weighted deficit round-robin (WDRR) algorithm. WDRR is aware of packet sizes, which can vary significantly in a CEE environment, making WDRR more suitable than a regular weighted round-robin (WRR) method, which selects groups based only on packet counts.

Note: Actual bandwidth used by any specific PGID may vary from configured values by up to 10% of the available bandwidth in accordance with 802.1Qaz ETS standard. For example, a setting of 10% may be served anywhere from 0% to 20% of the available bandwidth at any given time.

Unlimited Bandwidth for PGID 15

PGID 15 is permitted unlimited bandwidth and is generally intended for critical traffic (such as switch management). Traffic in this group is given highest priority and is served before the traffic in any other priority group.

If PGID 15 has low traffic levels, most of the switch's bandwidth will be available to serve priority groups 0 through 7. However, if PGID 15 consumes a larger part of the switch's total bandwidth, the amount available to the other groups is reduced.

Note: Consider traffic load when assigning priority values to PGID 15. Heavy traffic in this group may restrict the bandwidth available to other groups.

Configuring ETS

Consider an example with two business critical applications:

Table 32. *ETS Configuration*

Priority	Usage	PGID	Bandwidth
0	LAN (best effort delivery)	0	10%
1	LAN (best effort delivery)	0	10%
2	LAN (best effort delivery)	0	10%
3	FCoE or RoCE (PFC enabled)	3	50%
4	Business Critical LAN (lossless Ethernet, with PFC)	4	20%
5	Latency-sensitive LAN	2	20%
6	Latency-sensitive LAN	2	20%
7	Network Management (strict)	15	NA

In this example, Business critical LAN traffic (priority 4) is moved from PG 2 to PG 4 and Network management traffic (priority 7) is moved to PG 15 (strict priority). This leaves latency-sensitive LAN traffic (priorities 5 and 6) in PG 2 itself. Also, a new group for network management traffic has been assigned. Finally, the bandwidth allocation for priority groups 3, 4, and 5 are revised.

Note: DCBX may be configured to permit sharing or learning PFC configuration with or from external devices. This example assumes that PFC configuration is being performed manually. See "Data Center Bridging Capability Exchange" on page 367 for more information on DCBX.

This example can be configured using the following commands:

1. Turn CEE on.

```
Switch(config)# cee enable
```

Notes:

- Turning CEE on will automatically change some 802.1p QoS and 802.3x standard flow control settings and menus (see "Turning CEE On or Off" on page 355).
- The NE10032 only supports one PFC enabled priority. When you enable CEE, priority 3 will be enabled by default, but you can disable PFC on priority 3 and enable PFC on another priority.

2. Create a new PGID-4 for Business critical LAN traffic:

```
Switch(config)#cee ets priority-group pgid 4 priority 4
```

3. Assign bandwidth to the configured Priority Group:

```
Switch(config)#cee ets bandwidth-percentage 0 10 2 20 3 50 4 20
```

4. Enable PFC on priority 4 for lossless:

```
Switch(config)#cee pfc priority 4 enable
```

5. Optionally, set configuration descriptions:

```
Switch(config)#cee ets priority-group pgid 0 description "Regular LAN"
Switch(config)#cee ets priority-group pgid 2 description
   "Latency-sensitive LAN"
Switch(config)#cee ets priority-group pgid 3 description "RoCEv2 Traffic"
Switch(config)#cee ets priority-group pgid 4 description "Biz-critical
   LAN"
Switch(config)#cee ets priority-group pgid 15 description "Network
   Management"
Switch(config)#cee pfc priority 0 description "Regular LAN"
Switch(config)#cee pfc priority 3 description "RoCEv2"
Switch(config)#cee pfc priority 4 description "Biz-Critical LAN"
```

Note: Priority group 15 is permitted unlimited bandwidth. As such, the commands for priority group 15 do not include bandwidth allocation.

6. Save the configuration.

To view the configuration, use the following commands:

```
Switch(config)# display cee ets
ETS Global Admin Configuration:
PGID    BW%   COSq  Priorities          Description
========================================================================
=====
0       10    0      0 1 2               "Regular LAN"
1       0     NA
2       20    2      5 6                 "Latency-sensitive LAN"
3       50    3      3                   "RoCEv2 Traffic"
4       20    4      4                   "Biz-critical LAN"
5       0     5
6       0     6
7       0     7
15      NA    1      7                   "Network Management"
```

```
Switch(config)# display cee pfc
Global Admin PFC State: On
Priority  State  Description
===========================
   0      Dis    "Regular LAN"
   1      Dis
   2      Dis
   3      Ena    "RoCEv2"
   4      Ena    "Biz-Critical LAN"
   5      Dis
   6      Dis
   7      Dis
```

Data Center Bridging Capability Exchange

Data Center Bridging Capability Exchange (DCBX) protocol is a vital element of CEE. DCBX allows peer CEE devices to exchange information about their advanced capabilities. Using DCBX, neighboring network devices discover their peers, negotiate peer configurations, and detect misconfigurations.

DCBX provides two main functions on the switch:

- Peer information exchange

  The switch uses DCBX to exchange information with connected CEE devices. For normal operation of any FCoE implementation on the switch, DCBX must remain enabled on all ports participating in FCoE.

- Peer configuration negotiation

  DCBX also allows CEE devices to negotiate with each other for the purpose of automatically configuring advanced CEE features such as PFC, ETS and Application protocol. The administrator can determine which CEE feature settings on the switch are communicated to and matched by CEE neighbors, and also which CEE feature settings on the switch may be configured by neighbor requirements.

The DCBX feature requires CEE to be turned on (see).

DCBX Modes

Interfaces can use one of the following DCBX modes:

- CEE DCBX (1.01)
- IEEE DCBX (802.1Qaz)

Default mode is IEEE DCBX. The DCBX mode is determined by auto-negotiation with the peer. If the peer/remote port is capable of sending DCBX TLVs in IEEE mode, the interface is set to IEEE DCBX mode. If the peer/remote port doesn't support IEEE mode, the interface uses CEE DCBX mode.

DCBX Settings

When CEE is turned on, DCBX is enabled for peer information exchange on all ports. For configuration negotiation, the following default settings are configured:

- PFC: Enabled on 802.1p priority 3
- ETS
  - Priority group 0 includes priority values 0 through 2, with bandwidth allocation of 10%
  - Priority group 2 includes priority values 4 through 7, with bandwidth allocation of 40%
  - Priority group 3 includes priority value 3, with bandwidth allocation of 50%
- No application protocol is configured by default.

Note: The user may choose to configure RoCEv1/RoCEv2/iSCSI instead of FCoE.

Enabling and Disabling DCBX

When CEE is turned on, DCBX is enabled by default on all ports. Enabling or disabling DCBX TLVs on any port is possible by using the following command:

```
Switch(config-if)# [no] cee dcbx enable
```

When CEE is turned on and DCBX is enabled on a port, Link Layer Detection Protocol (LLDP) is used to exchange DCBX parameters between CEE peers. Also, the interval for LLDP transmission time is set to one second for the first five initial LLDP transmissions, after which it is returned to the administratively configured value. The minimum delay between consecutive LLDP frames is also set to one second as a DCBX default.

Peer Configuration Negotiation

CEE peer configuration negotiation can be enabled or disabled on a per-port basis for a particular CEE feature.For each supported feature, the administrator can configure the following flag:

- The advertise flag

  When this flag is set for a particular feature, the switch settings will be transmit to the remote CEE peer. If the peer is capable of the feature, and willing to accept the switch settings, it will be automatically reconfigured to match the switch.

Note: By default, the advertise flag is set for ETS, PFC and for application protocol.

These flags are available for the following CEE features:

- Application Protocol

  DCBX exchanges information regarding FCoE and FIP snooping, including the 802.1p priority value used for FCoE traffic. The advertise flag is set or reset using the following command:

```
Switch(config-if)# [no] cee dcbx app-proto advertise
```

- PFC

  DCBX exchanges information regarding whether PFC is enabled or disabled on the port. The advertise flag is set or reset using the following command:

```
Switch(config-if)# [no] cee dcbx pfc advertise
```

- ETS

  DCBX exchanges information regarding ETS priority groups, including their 802.1p priority members and bandwidth allocation percentages. The advertise flag is set or reset using the following command:

```
Switch(config-if)# [no] cee dcbx ets advertise
```

Configuring DCBX

Consider the following example:

- RoCEv2 is used on ports 1/1-1/10.
- All other ports are disabled or are connected to regular (non-CEE) LAN devices.

In this example, the switch acts as the central point for CEE configuration. FCoE-related ports will be configured for advertising CEE capabilities, but not to accept external configuration. Other LAN ports that use CEE features will also be configured to advertise feature settings to remote peers, but not to accept external configuration. DCBX will be disabled on all non-CEE ports.

This example can be configured using the following commands:

1. Turn CEE on.

```
Switch(config)# cee enable
```

Note: Turning CEE on will automatically change some 802.1p QoS and 802.3x standard flow control settings and menus (see "Turning CEE On or Off" on page 355).

2. Disable DCBX for each non-CEE port as appropriate:

```
Switch(config-if)# no cee dcbx enable
```

3. Save the configuration.

CEE Configuration Examples

This section provides examples on how to configure CEE on the switch.

CEE Example 1

In this example, the default CEE configuration is explained:

1. By default, PFC is enabled on priority 3. Optionally, set descriptions for priority and priority group.

```
Switch(config)# cee pfc priority 3 description RoCEv2_priority
Switch(config)# cee ets priority-group pgid 3 description RoCEv2_Traffic
```

2. Optionally, configure rocev2 as the application protocol.

```
Switch(config)# cee app-proto RoCEv2_Traffic rocev2 priority 3
```

3. Turn CEE on.

```
Switch(config)# cee enable
```

4. Verify the configuration:

```
Switch(config)# display cee ets
ETS Global Admin Configuration:
PGID    BW%   COSq  Priorities            Description
====================================================
0       10    0      0 1 2
1       0     NA
2       40    2      4 5 6 7
3       50    3      3                     RoCEv2_Traffic
4       0     4
5       0     5
6       0     6
7       0     7
15      NA    1

Switch(config)# display cee pfc
Global Admin PFC State: On
Priority  State  Description
============================
   0      Dis
   1      Dis
   2      Dis
   3      Ena    RoCEv2_priority
   4      Dis
   5      Dis
   6      Dis
   7      Dis


Switch(config)# display cee app-proto
Advertise   Protocol   ProtoId   Priorities         ConfigName
======================================================================
On       UDP      RoCEv2   3             RoCEv2_Traffic
```

CEE Example 2

In this example, CEE is configured with RoCEv2 and another Biz-critical LAN application or iSCSI:

1. Create a new PGID 4 for iSCSI traffic.

```
Switch(config)# cee ets priority-group pgid 4 priority 4
```

2. Allocate bandwidth to the Priority Groups.

```
Switch(config)# cee ets bandwidth-percentage 0 10 2 20 3 50 4 20
```

3. Enable PFC on priority 4 for lossless.

```
Switch(config)# cee pfc priority 4 enable
```

4. By default, PFC is enabled on priority 3. Optionally, set descriptions for priority and priority group 3.

```
Switch(config)# cee pfc priority 3 description RoCEv2_priority
Switch(config)# cee ets priority-group pgid 3 description RoCEv2_Traffic
```

5. Optionally, configure rocev2 as the application protocol for priority 3.

```
Switch(config)# cee app-proto RoCEv2_Traffic rocev2 priority 3
```

6. Optionally, set descriptions for priority and priority group 4.

```
Switch(config)# cee ets priority-group pgid 4 description iSCSI_Traffic
Switch(config)# cee pfc priority 4 description iSCSI
```

7. Optionally, configure iSCSI as the application protocol for priority 4.

```
Switch(config)# cee app-proto iSCSI_Traffic iscsi priority 4
```

8. Turn CEE on.

```
Switch(config)# cee enable
```

9. Verify the configuration:

```
Switch(config)# display cee ets
ETS Global Admin Configuration:
PGID    BW%   COSq  Priorities            Description
========================================================
0       10    0      0 1 2
1       0     NA
2       20    2      5 6 7
3       50    3      3                     RoCEv2_Traffic
4       20    4      4                     iSCSI
5       0     5
6       0     6
7       0     7
15      NA    1


Switch(config)# display cee pfc
Global Admin PFC State: On
Priority  State  Description
============================
   0      Dis
   1      Dis
   2      Dis
   3      Ena    RoCEv2_priority
   4      Ena    iSCSI
   5      Dis
   6      Dis
   7      Dis



Switch(config)# display cee app-proto
Admin Configuration:
Application Protocol Willing mode is not supported on the Switch.
Advertise   Protocol    ProtoId   Priorities          ConfigName
=========================================================================
On          UDP         RoCEv2    3                   RoCEv2_Traffic
On          TCP         iSCSI     4                   iSCSI_Traffic
```

Chapter 16. Secure Mode

Secure mode enables you to determine which protocols can be enabled. In secure mode, only secured traffic and secured authentication management are allowed.

The following topics are discussed in this chapter:

- "Secure Mode Overview" on page 374
- "Using Protocols With Secure Mode" on page 375
- "Enabling and Disabling Secure Mode" on page 378

Secure Mode Overview

The concept of "secure mode" introduces the

- **Legacy Mode**

  Legacy Mode maintains the existing security behavior of the switch. All communication protocols currently supported by switch software continue to be allowed and supported in this mode. All behaviors of the switch remain the same; the only difference is you can set the mode which will take effect after the next reboot of the switch.

- **Secure Mode**

  In *Secure Mode*, only secure communication protocols are allowed to be enabled. Communication protocols that are deemed to be not secure are disabled and not allowed to run on the switch.

Note: Once a switch has entered Secure Mode, it cannot return to Legacy Mode without a reboot.

Secure mode places the switch in a state where the following rules are in effect:

- Older and less secure ciphers will be inaccessible. Currently ssh and REST use these ciphers, but any changes will apply to any future cipher suite uses.

- Communication protocols deemed insecure, such as telnet, will be disabled, requiring the use of secure protocols; in this case, ssh.

- When protocols have newer, more secure versions, these protocols will be restricted to the secure version. For example, SNMP will be restricted to version 3 and ssh will be limited to version 2.0.

Using Protocols With Secure Mode

Some protocols can be used with secure mode. This section explains which protocols can and cannot operate with secure mode on the switch.

Insecure Protocols

When you are in secure mode, the following protocols are deemed "insecure" and are disabled:

- HTTP
- SNMPv1
- SNMPv2
- Telnet
- Telnet6
- FTP
- TFTP server

Except for the TFTP server, these protocols cannot be enabled when the switch is operating in Secure Mode because the commands to enable or disable them disappear with secure mode enabled.

The following protocols, although deemed "insecure" by secure mode, are enabled by default and can be disabled.

- Radius
- TACACS+
- LDAP (with no cleartext support in secure mode)
- TFTP client (signed images only)
- DHCP client
- SysLog

Secure Protocols

The following protocols are deemed "secure" and are enabled by default in Secure Mode:

- SCP Server
- SNMPv3 Client
- SFTP Client
- SSHv2 Server
- SSHv2 Client
- HTTPS Server

You can disable these protocols.

Only the following NIST 800-131a compliant cryptographic algorithms are allowed:

- SSH (Server and Client)

- KexAlgorithms:
  - ecdh-sha2-nistp256
  - ecdh-sha2-nistp384
  - ecdh-sha2-nistp521
  - diffie-hellman-group14-sha1

- Ciphers:
  - aes128-ctr
  - aes192-ctr
  - aes256-ctr
  - aes128-gcm@openssh.com
  - aes256-gcm@openssh.com

- MACs:
  - hmac-sha2-256
  - hmac-sha2-512
  - hmac-sha1-etm@openssh.com
  - hmac-sha2-256-etm@openssh.com
  - hmac-sha2-512-etm@openssh.com

Table 33 contains a list of secure mode compliant ciphers.

Table 33. *Secure mode compliant ciphers*

Cipher Name	Key Exchange	Authenti-cation	Encryption	MAC
ECDHE-RSA-AES256-GCM-SHA384	ECDH	RSA	AESGCM(256)	AEAD
ECDHE-RSA-AES256-SHA384	ECDH	RSA	AES(256)	SHA384
ECDHE-RSA-AES128-GCM-SHA256	ECDH	RSA	AESGCM(128)	AEAD
ECDHE-RSA-AES128-SHA256	ECDH	RSA	AES(128)	SHA256

Insecure Protocols Unaffected by Secure Mode

The following protocols are deemed "insecure" but can be enabled in all Security Policy Modes:

- Ping
- Ping IPv6
- Traceroute
- Traceroute IPv6
- TFTP IPv6
- SNMPv3 IPv6
- bootp

Notes:

- Telnet IPv6 and TFTP IPv6 are disabled in Secure Mode.
- TFTP IPv6 is allowed in Secure Mode for signed image transfers only.

In secure mode, only protocols that are deemed to be secure are allowed to run.:

When in secure mode, the following insecure protocols and the commands to configure them disappear:

- SNMPv1
- SNMPv2
- Telnet (server and client)
- Telnet IPv6 (server and client)
- FTP (server and client)
- TFTP server
- TFTP client (except for signed image transfers)

Enabling and Disabling Secure Mode

To enable Secure Mode on the switch, enter:

```
Switch(config)# secure mode enable
```

Note: The switch will remain in Legacy Mode until you reboot.

To disable Secure Mode on the switch, enter:

```
Switch(config)# no secure mode enable
```

Note: The switch will remain in Secure Mode until you reboot.

To display the running and configured security policies, enter:

```
Switch(config)# display security mode
```

Part 4: IP Routing

This section discusses Layer 3 switching functions. In addition to switching traffic at near line rates, the application switch can perform multi-protocol routing. This section discusses basic routing and advanced routing protocols:

Chapter 17. Basic IP Routing

This chapter provides configuration background and examples for using the switch to perform IP routing functions. The following topics are addressed in this chapter:

IP Routing

Internet Protocol (IP) Routing is the mechanism by which traffic travels across multiple networks, from its source to its destination. IP routing sends packets outside the local network by using Layer 3 protocols, like Address Resolution Protocol (ARP), Border Gateway Protocol (BGP), or Open Shortest Path First (OSPF).

The switch builds a forwarding table that correlates destination IP addresses with their next-hop addresses. A next-hop address is the IP address of the network device that is next in line between the switch and the destination device. If a packet is intended for a certain device, the switch will check its forwarding table for the IP address of the destination device. Once a match is found, the switch will send the packet through the interface on which the next-hop address is present. The next-hop device will perform the same process until the packet reaches its destination.

The switch uses a combination of configurable switch interfaces and IP routing options. The switch IP routing capabilities provide the following benefits:

- Connects the server IP subnets to the rest of the backbone network.
- Provides the ability to route IP traffic between multiple Virtual Local Area Networks (VLANs) configured on the switch.

The switch supports both IPv4 and IPv6 routing. For more detailed information about IPv6 routing, see Chapter 20, "Internet Protocol Version 6".

IP routing runs only on Layer 3 interfaces, such as:

- switch Virtual Interfaces (SVIs)
- the management interface
- ethernet interfaces configured as routed ports

For more information about Layer 3 interfaces, see Chapter 8, "Interface Management" and Chapter 18, "Routed Ports".

By default, IP routing is enabled on the switch.

To enable or disable the forwarding of IPv4 traffic, use the following command:

```
Switch(config)# [no] ip forwarding
```

To enable or disable the forwarding of IPv6 traffic, use the following command:

```
Switch(config)# [no] ipv6 forwarding
```

Direct and Indirect Routing

Direct routing is used when both source and destination devices are present in the same network, the switch sends the packet directly from one device to the other. Forwarding table entries for direct routes are referred to as directly connected routes and are marked in the IP forwarding table with the letter 'C'.

When the source and destination devices are not in the same network, the packet must be forwarded by a network node (router) that knows how to reach the destination device. This network node is called a next-hop device (router).

Static Routing

Static routing means that the packet is forwarded using a manually configured route. Configuring a static route teaches the switch how to reach the specified destination IP address. Forwarding table entries for static routes are marked with the letter 'S'.

Static routing is recommended for small networks, because every static route needs to be manually configured on each switch or router in the network.

To add or remove a static IPv4 route, use the following command:

```
Switch(config)# [no] ip route <IPv4 address>/<prefix length> <next-hop address>
```

For example:

```
Switch(config)# ip route 10.10.10.0/24 10.28.12.10
```

To add or remove a static IPv6 route, use the following command:

```
Switch(config)# [no] ipv6 route <IPv6 address>/<prefix length> <next-hop address>
```

For example:

```
Switch(config)# ipv6 route 2001:1::1/128 fe80::a8bb:ccff:fe00:300
```

For more details about the above commands, consult the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

Note: The switch supports up to a maximum of 256 static routes, either IPv4 or IPv6.

To view configured static routes, use the following command:

```
Switch# display {ip|ipv6} static-route
```

Dynamic Routing

In dynamic routing, the packet is forwarded using routes that the switch learns automatically via routing protocols, such as BGP or OSPF. Routing protocols are used by switches or routers to exchange routing information. Forwarding table entries for dynamic routes are marked with different letters depicting the protocol through which the route was learned. For example, 'B' is used for routes learned through BGP and 'O' for routes learned through OSPF.

Dynamic routing is fault tolerant. If a learned link goes down, the switch can learn another route to the destination IP address. Another advantage is the low administrative overhead compared to configuring static routing.

One of the disadvantages of dynamic routing is the increased memory and CPU usage due to the processing of routing information received from other routers.

For more information about routing protocols, see Chapter 22, "Border Gateway Protocol" and Chapter 23, "Open Shortest Path First".

Note: The switch supports up to a maximum of 15870 IPv4 dynamic routes or 6143 IPv6 dynamic routes.

To remove a route from the forwarding table, use the following command:

```
Switch# remove {ip|ipv6} route <IP address>/<prefix length>
```

To delete all routes from the forwarding table, use the following command:

```
Switch# remove {ip|ipv6} route *
```

Default Gateway

The default gateway is the next-hop address that the switch uses when forwarding packets for which it does not have a dynamic or static route. The default gateway is a static route configured with 0.0.0.0/0 as its destination IP address.

To configure an IPv4 address as the default gateway for IPv4 traffic, use the following command:

```
Switch(config)# ip route 0.0.0.0/0 <next-hop address>
```

To configure an IPv6 address as the default gateway for IPv6 traffic, use the following command:

```
Switch(config)# ipv6 route 0::0/0 <next-hop address>
```

Note: Routes that have a specific destination take precedence over the default gateway.

Virtual Routing and Forwarding

Virtual Routing and Forwarding (VRF) allows multiple instances of a routing table to work simultaneously inside a router.

These are the advantages of using VRF technology:

- Network paths can be segmented without using multiple devices.

- The routing instances are independent; the same or overlapping IP addresses can be used without conflicting with each other.

- VRF acts like a logical router (LSR), but while a LSR may include many routing tables, a VRF instance uses only a single routing table.

- VRF uses routing tables known as forwarding information bases (FIBs). Each VRF context created is associated with a FIB ID.

By default, the switch has 2 pre-defined VRFs:

- *default* vrf (`FIBID 0`)

- *management* vrf (`FIBID 1`). The management VRF contains only the management interface, being a reserved vrf for management purposes only.

Routing Information Base

The Routing Information Base (RIB) is a table in which the switch keeps information about known routes, like the destination IP address, the next-hop address, or the associated metric (path cost). It consists of static entries (routes manually configured) and dynamic entries (routes learned through routing protocols like BGP or OSPF).

The main objective of routing protocols is the construction of the RIB. This implies the exchange of routing information between neighboring network devices.

A route redistribution can be requested by a routing protocol. When the switch receives the request, it will share its routing information based upon the protocol that made the request. For example, if BGP requested a route redistribution, the switch will share all the routes it has learned through BGP.

Any route changes on the switch will trigger route updates to be sent to the neighboring devices. A route update is shared only if the Hardware Specific Layer (HSL) gives a positive acknowledgement for that route.

Routing protocols, such as BGP, can use the Next-Hop Lookup service to check the availability of a route.

RIB supports the maintaining of routes in a stale state for the duration of a graceful restart of the BGP service. This means that traffic forwarding is not disturbed for the duration of the restart.

There are scenarios in which the switch is not directly connect to the next-hop device through an interface. Such occurrences are called indirect next-hops.

RIB supports both normal and ECMP routes with indirect next-hops. If there is a direct route through which the next-hop is reachable (the route leads directly to the next-hop), RIB will install the direct route, instead of an indirect one.

Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) is a Layer 3 detection protocol used to detect failures in the forwarding path of adjacent network routers.

For BFD to work properly, a session must be established between two devices (or BFD peers). This means BFD must be configured on both BFD peers. Once BFD has been enabled on the connecting interfaces and on the peers, a BFD session is established, followed by the negotiation of BFD timers.

Note: The switch supports up to a maximum of 100 BFD sessions.

BFD uses few system resources and provides a fast failure detection, that is independent of all media types, encapsulations, topologies, and routing protocols, like BGP or OSPF. Once a failure is detected, BFD notifies the routing protocols of the local switch to trigger a routing table update, thus reducing network convergence time.

BFD control packets are encapsulated in User Datagram Protocol (UDP) packets, that have an assigned destination port of 3784 or 3785 and source port between 49152 and 65535. BFD control packets are always sent as unicast packets to the configured BFD peer.

BFD sessions are supported for both IPv4 and IPv6 addressing.

By default, BFD is enabled on the switch. It can be enabled or disabled for each individual interface. BFD can be enabled on ethernet interfaces or loopback interfaces.

To enable or disable BFD on an interface, use the following command:

```
Switch(config-if)# [no] bfd {ipv4|ipv6}
```

Note: BFD cannot be enabled or disabled globally on the switch. It can be enabled or disabled on individual interfaces.

To configure the exchange of BFD control packets during BFD sessions, use the following command:

```
Switch(config)# [no] bfd {ipv4|ipv6} interval <interval> minrx <minrx>
multiplier <multiplier>
```

where:

Parameter	Description
interval	The time interval, in milliseconds, between the transmission of two consecutive BFD control packets; an integer from 50-999. Default value:100 milliseconds.

Parameter	Description
minrx	The time interval, in milliseconds, at which BFD control packets are expected to be received; an integer from 50-999. Default value: 100 milliseconds.
multiplier	The number of consecutive BFD control packets that must not be received before the switch declares its BFD peer unavailable and sends a notification to the peer; an integer from 3-50. Default value: 3.

You can also configure these parameters on an interface. They can be different than those configured globally on the switch.

```
Switch(config-if)# [no] bfd interval <50-999> minrx <50-999> multiplier <3-50>
```

Notes:

● The configured intervals at which the switch sends and expects BFD control packets are only the desired values. BFD will negotiate with its BFD peer the actual values used during the BFD session when exchanging packets.

● These parameters apply to BFD control packets while BFD echo mode is disabled. If BFD echo mode is enabled, the configured parameters are applied to BFD echo packets instead.

BFD Asynchronous Mode

BFD operates in asynchronous mode, meaning that control packets are periodically exchanged between BFD peers. If a BFD peer sends consecutive control packets to its peer and does not receive any reply from the other device, it will declare the BFD session to be down.

BFD Echo Mode

If a switch is configured with BFD Echo mode, it sends its peer a series of echo packets and requests the peer to send back (echo) those hello packets the switch previously sent. If hello packets from the echoed traffic are not received by the switch, it will declare the BFD session as being down.

BFD Echo mode can run independently on each BFD peer.

By default, BFD echo mode is disabled and it can be enabled on each switch interface. To enable or disable BFD echo mode on an interface, use the following command:

```
Switch(config-if)# [no] bfd echo
```

BFD Echo mode uses a timer called slow timer that determines how fast a new BFD session is established. The slow timer also specifies the interval that asynchronous BFD sessions use to exchange control packets, replacing the BFD interval. While control packets are transmitted using the slow timer, echo packets still use the configured BFD interval.

To globally configure the BFD slow timer, in milliseconds, use the command:

```
Switch(config)# bfd slow-timer <1000-30000>
```

By default, the slow timer is set to 2000 milliseconds (2 seconds). To reset the slow timer to its default value, use the command:

```
Switch(config)# no bfd slow-timer
```

BFD Peer Support

BFD provides failure detection in the forwarding path for destinations that are directly connected, one hop away, or more than one hop away from the switch. The maximum number of hops connecting a switch and its BFD peer is 255.

To add or remove a static single hop BFD peer on an interface, use the command:

```
Switch(config-if)# [no] bfd neighbor src-ip <source IP address> dest-ip
<destination IP address>
```

Note: The *IP address* can be either an IPv4 address or an IPv6 address.

To add or remove a static multi-hop BFD peer on an interface, use the command:

```
Switch(config-if)# [no] bfd neighbor src-ip <source IP address> dest-ip
<destination IP address> multihop
```

Different parameters (interval, minrx, and multiplier) can be globally configured for multi hop peers:

```
Switch(config)# [no] bfd multihop-peer <peer IP address> interval <50-999> minrx
<50-999> multiplier <3-50>
```

BFD Static Routes

BFD can also be enabled on static IP routes. Only one BFD session is created for multiple static routes using the same next-hop address that run through a particular switch interface.

Note: BFD is supported only on single hop static routes, and not on multi-hop static routes.

To enable or disable BFD on a static IP route, use the command:

```
Switch(config)# [no] {ip|ipv6} route static bfd <interface> <BFD peer address>/
<prefix length>
```

Note: The *BFD peer address* can be either an IPv4 address or an IPv6 address, depending on the type of static route.

For example:

```
Switch(config)# ip route static bfd ethernet 1/12 10.10.10.0/24
```

BFD Authentication

BFD supports the enabling or disabling of authentication for BFD sessions between peers. If authentication does not match for both peers, received packets are discarded.

BFD authentication is enabled by specifying an authentication algorithm and key-chain:

- authentication algorithms:
  - simple - plain text password
  - keyed-md5 - Keyed Message Digest 5 hash algorithm
  - keyed-sha1 - Keyed Secure Hash Algorithm I
  - meticulous-keyed-md5 - Meticulous Keyed Message Digest 5 hash algorithm
  - meticulous-keyed-sha1 - Meticulous Keyed Secure Hash Algorithm I
- authentication key-chains:
  - key-id - a single key
  - key-chain - multiple keys

A key contains the secret data and the time when it becomes valid. The authentication algorithm and key-chain must be configured on both BFD peers, and must be identical. If any mismatch occurs, the BFD session cannot be established.

By default, BFD sessions are established without authentication.

To configure BFD authentication on an interface, use one of the following commands:

- configure authentication to use a single key:

```
Switch(config-if)# [no] bfd [ipv4|ipv6] authentication <authentication
algorithm> key-id <0-255 (key ID)> key <key string>
```

- configure authentication to use a key-chain, containing multiple keys:

```
Switch(config-if)# [no] bfd [ipv4|ipv6] authentication <authentication
algorithm> key-chain <key-chain name>
```

For example:

```
Switch(config-if)# bfd authentication meticulous-keyed-sha1 key-chain kch7
```

Authentication can also be configured for multi-hop BFD peers:

```
Switch(config)# [no] bfd multihop-peer <peer IP address> authentication
<authentication algorithm> key-id <0-255 (key ID)> key <key string>
```

```
Switch(config-if)# [no] bfd multihop-peer <peer IP address> authentication
<authentication algorithm> key-chain <key-chain name>
```

Generalized TTL Security Mechanism

The Generalized TTL Security Mechanism (GTSM) provides the switch's IP-based control plane with protection from CPU utilization based attacks. It relies on a packet's TTL or hop limit to protect the switch from packets sent in a rapid succession.

By default, GTSM is disabled. To enable or disable GTSM for BFD, use the following command:

```
Switch(config)# bfd gtsm {enable|disable}
```

By default, GTSM uses a TTL or hop limit of 255. To change the TTL or hop limit, run the following command:

```
Switch(config)# bfd gtsm ttl <1-255>
```

To reset the TTL or hop limit to its default value of 255, use the following command:

```
Switch(config)# no bfd gtsm ttl <1-255>
```

BFD and BGP

When BFD detects a forwarding failure, it immediately notifies its client protocols.

BFD can work together with BGP to prevent the use of aggressive keep-alive timers. To keep this timer below 10 or 30 seconds, it is recommended to use BFD as well. BFD is compatible with both eBGP and iBGP single or multi-hop peers.

BFD and OSPF

When BFD detects a forwarding failure, it immediately notifies its client protocols.

BFD can work together with OSPF to increase route convergence as an alternative to adjusting the OSPF Hello Interval and Dead Interval. BFD improves the speed of failure detection by having shorter timer limits than the OSPF failure detection mechanisms.

Routing Between IP Subnets

The physical layout of most corporate networks has evolved over time. Classic hub/router topologies have given way to faster switched topologies, particularly now that switches are increasingly intelligent. The switch is intelligent and fast enough to perform routing functions at wire speed.

The combination of faster routing and switching in a single device allows you to build versatile topologies that account for legacy configurations.

For example, consider a corporate campus that has migrated from a router-centric topology to a faster, more powerful, switch-based topology. As is often the case, the legacy of network growth and redesign has left the system with a mix of illogically distributed subnets.

This is a situation that switching alone cannot cure. Instead, the router is flooded with cross-subnet communication. This compromises efficiency in two ways:

- Routers can be slower than switches. The cross-subnet side trip from the switch to the router and back again adds two hops for the data, slowing throughput considerably.

- Traffic to the router increases, increasing congestion.
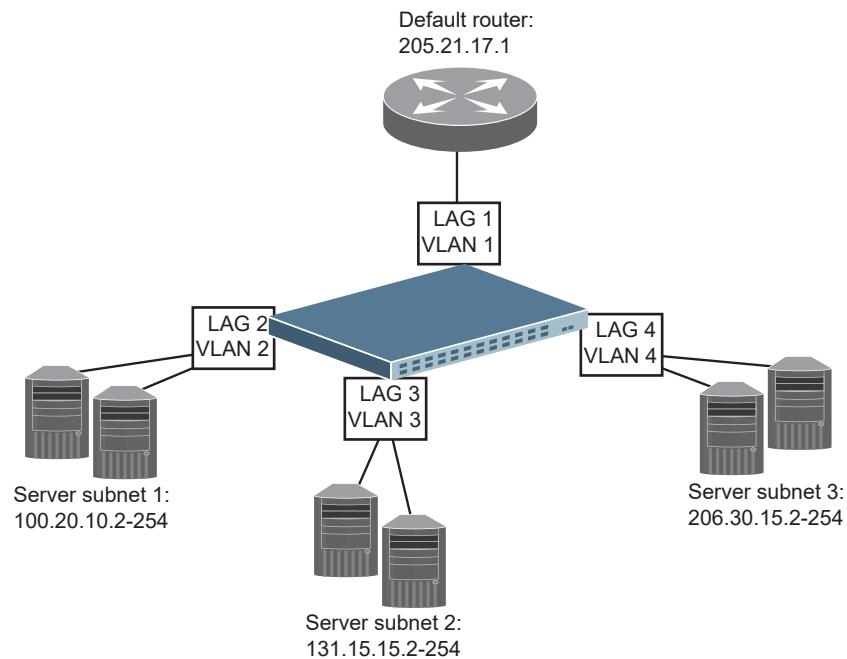
Even if every end-station could be moved to better logical subnets (a daunting task), competition for access to common server pools on different subnets still burdens the routers.

This problem is solved by using switches with built-in IP routing capabilities. Cross-subnet LAN traffic can now be routed within the switches with wire speed switching performance. This eases the load on the router and saves the network administrators from reconfiguring every end-station with new IP addresses.

Example of Subnet Routing

Consider the role of the switch in the following configuration example:

Figure 11. Switch-Based Routing Topology

The switch connects the Gigabit Ethernet and Fast Ethernet LAGs from various switched subnets throughout one building. Common servers are placed on another subnet attached to the switch. A primary and backup router are attached to the switch on yet another subnet.

Without Layer 3 IP routing on the switch, cross-subnet communication is relayed to the default gateway (in this case, the router) for the next level of routing intelligence. The router fills in the necessary address information and sends the data back to the switch, which then relays the packet to the proper destination subnet using Layer 2 switching.

With Layer 3 IP routing in place on the switch, routing between different IP subnets can be accomplished entirely within the switch. This leaves the routers free to handle inbound and outbound traffic for this group of subnets.

Using VLANs to Segregate Broadcast Domains

If you want to control the broadcasts on your network, use VLANs to create distinct broadcast domains. Create one VLAN for each server subnet and one for the router.

Configuration Example

This section describes the steps used to configure the example topology shown in Figure 11 on page 393.

1. Assign an IP address for each router and each server.

   The following IP addresses are used:

   Table 34. *Subnet Routing Example: IP Address Assignments*

Subnet	Devices	IP Addresses
1	Default router	205.21.17.1
2	Web servers	100.20.10.2-254
3	Database servers	131.15.15.2-254
4	Terminal Servers	206.30.15.2-254

2. Assign an IP interface for each subnet attached to the switch.

   Since there are four IP subnets connected to the switch, four interfaces are needed:

   Table 35. *Subnet Routing Example: IP Interface Assignments*

Interface	Devices	IP Interface Address
Port 1	Default router	205.21.17.3
Port 2	Web servers	100.20.10.1
Port 3	Database servers	131.15.15.1
Port 4	Terminal Servers	206.30.15.1

3. Determine which switch ports belong to which VLANs.

   The following table adds port and VLAN information:

   Table 36. *Subnet Routing Example: Optional VLAN Ports*

Devices	LAG	Switch Ports	VLAN
Default router	1	22	1
Web servers	2	1 and 2	2
Database servers	3	3 and 4	3
Terminal Servers	4	5 and 6	4

   Note: To perform this configuration, you must be connected to the switch Command Line Interface (CLI) as the administrator.

4. Add the switch ports to their respective VLANs:

```
Switch(config)# vlan 1-4
Switch(config-vlan)# exit

Switch(config)# interface ethernet 1/22
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 1
Switch(config-if)# aggregation-group 1 mode on
Switch(config-if)# exit

Switch(config)# interface ethernet 1/1-2
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 2
Switch(config-if)# aggregation-group 2 mode on
Switch(config-if)# exit

Switch(config)# interface ethernet 1/3-4
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 3
Switch(config-if)# aggregation-group 3 mode on
Switch(config-if)# exit

Switch(config)# interface ethernet 1/5-6
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 4
Switch(config-if)# aggregation-group 4 mode on
Switch(config-if)# exit
```

5. Now that the ports are separated into LAGs, the VLANs are assigned to the appropriate interfaces for each subnet. From Table 36 on page 394, the settings are made as follows:

```
Switch(config)# interface vlan 1
Switch(config-if)# ip address 205.21.17.3/24
Switch(config-if)# exit

Switch(config)# interface vlan 2
Switch(config-if)# ip address 100.20.10.1/24
Switch(config-if)# exit

Switch(config)# interface vlan 3
Switch(config-if)# ip address 131.15.15.1/24
Switch(config-if)# exit

Switch(config)# interface vlan 4
Switch(config-if)# ip address 206.30.15.1/24
Switch(config-if)# exit
```

6. Configure the default gateway to the routers' addresses.

The default gateway allows the switch to send outbound traffic to the router:

```
Switch(config)# ip route 0.0.0.0/0 205.21.17.1
```

7. Verify the configuration.

```
Switch# display vlan

VLAN    Name                            Status  IPMC FLOOD Ports
======= ============================== ======= ========== ================
1       default                         ACTIVE  IPv4,IPv6  Ethernet1/1(u)
                                                           Ethernet1/2(u)
                                                           Ethernet1/6(u)
                                                           Ethernet1/7(u)
                                                           Ethernet1/8(u)
                                                           Ethernet1/9(u)
                                                           Ethernet1/11(u)
                                                           Ethernet1/12(u)
                                                           Ethernet1/13(u)
                                                           Ethernet1/17(u)
                                                           Ethernet1/18(u)
                                                           Ethernet1/19(u)
                                                           Ethernet1/21(u)
...
```

```
Switch# display interface brief

-------------------------------------------------------------------------
Ethernet     VLAN Type Mode   Status  Reason               Speed Port
Interface
Agg#
-------------------------------------------------------------------------
Ethernet1/1  1    eth  access down    Link not connected    auto  --
Ethernet1/2  1    eth  access down    Link not connected    auto  --
Ethernet1/3  19   eth  access up      none                  10000 --
Ethernet1/4  20   eth  access down    Administratively down auto  --
Ethernet1/5  --   eth  routed down    Administratively down auto  --
Ethernet1/6  1    eth  access down    Link not connected    auto  --
Ethernet1/7  1    eth  access down    Link not connected    auto  --
Ethernet1/8  1    eth  access down    Link not connected    auto  --
Ethernet1/9  1    eth  access down    Link not connected    auto  --
Ethernet1/10 --   eth  routed down    Administratively down auto  --
...
```

```
Switch# display ip interface brief

Interface        IP-Address      Admin-Status          Link-Status
Ethernet1/5      21.1.1.1        administratively down down
Ethernet1/10     191.1.1.1       administratively down down
Ethernet1/14     16.1.1.2        up                    up
Ethernet1/20     116.0.0.1       up                    down
Ethernet1/22     22.1.1.2        administratively down down
mgmt0            10.241.40.13    up                    up
Vlan17           17.1.1.2        up                    down
Vlan18           18.1.1.2        up                    down
Vlan19           19.1.1.1        up                    up
Vlan20           20.1.1.1        up                    down
Vlan23           23.1.1.2        up                    down
Vlan24           24.1.1.2        up                    down
```

Examine the resulting information. If any settings are incorrect, make the appropriate changes.

ECMP Static Routes

Equal-Cost Multi-Path (ECMP) is a forwarding mechanism that routes packets along multiple paths of equal cost. ECMP provides equally-distributed link load sharing across the paths. ECMP routes allow the switch to choose between several next-hops toward a given destination. The switch performs periodic health checks (ping) on each ECMP gateway. If a gateway fails, it is removed from the routing table and an SNMP trap is sent.

RIB Support for ECMP Routes

RIB supports the adding of multiple next-hops to the same route. Such routes are called Equal Cost Multiple Paths (ECMP) routes.

The maximum number of different next-hops permitted for a route can be changed. To configure the maximum number of different paths for an ECMP route, use the following command:

```
Switch(config)# maximum-paths <1-32>
```

By default, the switch allows up to 32 paths for a single route. To reset this number to its default value, use the following command:

```
Switch(config)# no maximum-paths
```

ECMP Hashing

In case of an ECMP route, the next-hop address will be chosen based on a hash algorithm. The hashing parameters used are:

- source IP address
- destination IP address
- source and destination IP addresses
- Layer 4 source port
- Layer 4 destination port
- Layer 4 source and destination ports

To configure the hashing parameters, use the following command:

```
Switch(config)# ip load-sharing <hashing parameters> [universal-id <1-4294967295>]
```

Note: The universal ID is used to randomize the hashing result for each pair of source and destination addresses.

By default, the hash algorithm utilizes the all four hashing parameters and has a universal ID of 1431655765. To reset the hash algorithm to its default settings, use the following command:

```
Switch(config)# no ip load-sharing
```

To view the current hash settings, use the following command:

```
Switch# display ip load-sharing
```

When ECMP is configured in a multi-tier network, where multiple redundant paths are present, the hash algorithm on neighboring switches, that have the same hashing parameters, may calculate the same path for outgoing traffic. This results in a congestion of traffic on a upstream switch, while leaving other ECMP routes completely unused, thus negating the redundancy of the network.

To avoid such scenarios, a hash offset can be introduced in the hash algorithm calculations. The hash offset is just a number that is taken into account when calculating the hashing result. By default, the hash offset is set to 0, therefore having no influence on the hash calculation. When it is set to a non-zero value, the hash offset ensures that a different ECMP route is chosen.

Use the following command to set an ECMP hash offset:

```
Switch(config)# hardware ecmp hash-offset <0-15>
```

To reset it to its default value, use the following command:

```
Switch(config)# no hardware ecmp hash-offset
```

Note: ECMP hashing is not available for IPv6 traffic.

Configuring ECMP Static Routes

To configure ECMP static routes, add the same route multiple times, each with the same destination IP address, but with a different gateway IP address. These routes become ECMP routes.

1. Add a static route (IP address, prefix length and gateway).

```
Switch(config)# ip route 10.10.1.1/32 100.10.1.1
```

2. Continue adding static routes with the same IP address and prefix length, but a different gateway address.

```
Switch(config)# ip route 10.10.1.1/32 200.20.2.2
...
```

3. Select an ECMP hashing method (optional).

```
Switch(config)# ip load-sharing source-dest-ip
```

You may add up to 32 gateways for each static route.

4. Use the following command to check the status of ECMP static routes:

```
Switch(config)# display ip route static
```

Weighted ECMP Routes

In traditional ECMP with next hops, each next hop is added to the ECMP multipath once so traffic is distributed equally among the next hops. However, in some scenarios, traffic may use one path more than others, causing congestion. A lack of balance can occur because the ECMP hashing algorithm only considers the source, destination, or both when selecting the path, but not the use of the link.

Weighted ECMP lets you configure the multipath based on the use of each link, thus avoiding congestion and obtaining a better balance of traffic. This is achieved by adding the next hops in the multipath from one to four times.

Requirements for Weighted ECMP

To use weighted ECMP routes, you must have the following:

- ECMP route support
- hardware multipath group size changes to at least 128 paths when the weighted ECMP feature is enabled.

Notes:

- The next hop weights are reset when you enable or disable weighted ECMP, or when you change the maximum number of paths.
- The ECMP route weights are set dynamically and are not part of the running or saved configuration.

Configure Weighted ECMP

To configure weighted ECMP routes:

1. Enable the ECMP weighted routes feature:

```
Switch(config)# ip ecmp weight enable
```

2. Set the next hop weight for a specific ECMP interface:

```
Switch(config)# ip ecmp weight {interface {<IF name> | ethernet
<slot>/<chassis> | port-aggregation <PA number>} | <IP address>} <weight>
```

where:

Parameter	Description
IF name	Interface name; a string.
slot/chassis	The ethernet slot and chassis numbers.
PA number	The port aggregation number; an integer from 1-4096
IP address	The IPv4 or IPv6 address for the next hop.
weight	The ECMP weight value; an integer from 1-4.

To display the next hop weight, enter:

```
Switch(config)# display ip ecmp weight {interface {<IF name> | ethernet
<chassis number>/<port number> | port-aggregation <PA number>} | <IP address>}
```

where:

Parameter	Description
IF name	Interface name; a string.
slot/chassis	The ethernet slot and chassis numbers.
PA number	The port aggregation number; an integer from 1-4096
IP address	The IPv4 or IPv6 address for the next hop.

To disable weighted ECMP routes, enter:

```
Switch(config)# no ip ecmp weight enable
```

Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol (DHCP) is a transport protocol that provides a framework for automatically assigning IP addresses and configuration information to other IP hosts or clients in a large TCP/IP network. Without DHCP, the IP address must be entered manually for each network device. DHCP allows a network administrator to distribute IP addresses from a central point and automatically send a new IP address when a device is connected to a different place in the network.

The switch accepts gateway configuration parameters if they have not been configured manually. The switch ignores DHCP gateway parameters if the gateway is statically configured.

DHCP is an extension of another network IP management protocol, Bootstrap Protocol (BOOTP), with an additional capability of being able to allocate reusable network addresses and configuration parameters for client operation.

Built on the client/server model, DHCP allows hosts or clients on an IP network to obtain their configurations from a DHCP server, thereby reducing network administration. The most significant configuration the client receives from the server is its required IP address.

The switch support DHCP for both IPv4 and IPv6 addressing.

To enable or disable DHCP, use the following command:

```
Switch(config)# [no] feature dhcp
```

By default, DHCP is enabled on the management port, but disabled on all other switch Layer 3 interfaces.

For more details about DHCP, see "DHCP IP Address Services" on page 55.

Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) is used to send error messages indicating, for example, that a network device is unreachable. ICMP messages are primarily used for diagnostic and control purposes.

ICMP errors are sent back to the source device of the IP packet. For example, when a packet crosses a network node, its TTL decreases by one. When the TTL reaches zero, the packet is discarded and an ICMP error message is sent to source device.

ICMP messages are used by many network utilities, such as Traceroute or Ping.

ICMP messages are identified by type and code, that is used to identify the type of message. Below are a few of the most common ICMP message types:

- 0 - echo reply
  An echo reply is generated when the switch receives an echo request (ping) and it indicates that the switch can be reached.

- 3 - destination unreachable
  It means that there is error along the path to the destination device. This type has multiple codes that indicate the cause.

- 5 - redirect
  The switch detects that the packet is not optimally forwarded to its destination.

- 11 - time exceeded
  The TTL of the packet has reached zero.

ICMP can be configured independently on each switch interface.

To view the current settings regarding ICMP, use the following command:

```
Switch# display ip interface

IP Interface Status for VRF (default)
 loopback0, Interface Status: link up/admin up
 IP MTU:1500 bytes (using link MTU)
 IP icmp redirects: enabled
 IP icmp unreachables (except port): disabled
 IP icmp port-unreachable: enabled

 Vlan1, Interface Status: link up/admin up
 IP address: 205.21.17.1, IP subnet: 205.21.17.0/24
 IP MTU:1500 bytes (using link MTU)
 IP icmp redirects: enabled
 IP icmp unreachables (except port): disabled
 IP icmp port-unreachable: enabled
...
```

ICMP Redirects

The switch generates an ICMP redirect message when it detects that a packet is not optimally forwarded to the destination and sends the message to inform the source device to send future packets along the optimal path across the network. This kind of ICMP messages are identified as type 5.

For example, the switch, a host device and a neighboring router are all directly connected with each other. The switch is configured as the default gateway for the host. The host sends the switch a packet with a destination that is reachable only through the router. The switch sends host an ICMP redirect error and informs the host to send subsequent packets directly to the router, thus reducing the number of hops to the destination by one.

By default, ICMP redirect errors are enabled on all interfaces.

To enable or disable ICMP redirect error messages, use the following command:

```
Switch(config-if)# [no] ip redirects
```

ICMP Port Unreachable

The switch generates an ICMP port unreachable message when the interface of destination device is unavailable and sends the message to inform the source device of this. This kind of ICMP messages are identified as type 3, having error code 3.

By default, ICMP port unreachable errors are enabled on all interfaces.

To enable or disable ICMP port unreachable error messages, use the following command:

```
Switch(config-if)# [no] ip port-unreachable
```

ICMP Unreachable (except Port)

The switch generates an ICMP unreachable message when the destination device is not available. This kind of ICMP messages are identified as type 3 and have multiple error codes, ranging from 0 to 15. For example, an error code 1 means that the destination device could not be reached, while an error code 2 means that the protocol for which the packet was intended is not available, but the destination device is reachable.

By default, ICMP unreachable errors, except port unreachable messages, are disabled on all interfaces.

To enable or disable ICMP unreachable error messages (except port unreachable), use the following command:

```
Switch(config-if)# [no] ip unreachables
```

Chapter 18. Routed Ports

By default, all ports on the switch behave as switch ports, which are capable of performing Layer 2 switch functions, such as VLANs, STP, or bridging. Switch ports also provide a physical point of access for the switch IP interfaces, which can perform global Layer 3 functions, such as routing for BGP or OSPF.

However, switch ports can also be configured as routed ports. Routed ports are configured with their own IP address belonging to a unique Layer 3 network and behave similar to a port on a conventional router. Routed ports are typically used for connecting to a server or to a router.

This section discusses the following topics:

- "Routed Ports Overview" on page 406
- "Configuring a Routed Port" on page 408

Routed Ports Overview

When a switch port is configured as a routed port, it forwards Layer 3 traffic and no longer performs Layer 2 switching functions.

By default, all ethernet ports are configured as switch access ports. To configure the port to operate as a routed port, see "Configuring a Routed Port" on page 408.

You can also assign an IP address to a routed port and configure OSPF to route IP traffic through the interface.

A routed port has the following characteristics:

- Does not participate in bridging.

- Does not belong to any user-configurable VLAN.

- Does not implement any Layer 2 functionality, such as the Spanning Tree Protocol (STP).

- Is always in a forwarding state.

- Can participate in IPv4 or IPv6 routing.

- Can be configured with basic IP protocols, such as Internet Control Message Protocol (ICMP) and with Layer 3 protocols, such as Open Shortest Path First (OSPF) or Virtual Router Redundancy Protocol (VRRP).

- Layer 3 applications can be enabled, such as Network Time Protocol (NTP) or Dynamic Host Configuration Protocol (DHCP).

- Layer 3 configuration is saved even when the interface is shutdown.

- MAC address learning is always enabled.

- Native VLAN tagging is disabled.

- Flooding is disabled.

- Bridge Protocol Data Unit (BPDU)-guard is disabled.

- Link Aggregation Control Protocol (LACP) is disabled.

- Multicast threshold is disabled.

Notes:

- Ports on which Link Aggregation Control Protocol (LACP) is enabled or are part of a Link Aggregation Group (LAG) cannot be changed to routed ports.

- Ports with configured static MAC addresses cannot be changed to routed ports.

When a switch port is configured as a routed port, the following configuration changes are automatically implemented:

- The port is removed from all the VLANs it belonged to.

- The port is added to the first available internal VLAN on which flooding is disabled. The ID of this internal VLAN will be between 4000 - 4093, starting in descending order from 4093. The internal VLAN is assigned to the Common Internal Spanning Tree (CIST). You cannot change the VLAN number assigned to the routed port.

- STP is disabled and the port is set to a forwarding state.

- All the Layer 2 configuration is lost.

- The port will be deleted from the bridge.

- MAC Access Control Lists (ACL) from the Layer 2 port will be removed.

When a routed port is changed back to a switch port, the following changes take place:

- All relevant Layer 3 configuration is lost. Common Layer 3 and Layer 2 settings will be preserved during the transition.

- The ARP entry corresponding to the IP address is lost.

- The switch port is added to the default VLAN (VLAN 1).

- STP is turned on and the port is added to the default STG (STG 1).

- The switch port can participate in STG and VLAN flooding.

- The switch port can participate in bridging.

- LACP port attributes are set to default.

- Multicast threshold remains disabled.

- BPDU guard remains disabled.

- QoS configurations is lost.

- IP Access Control Lists (ACL) are removed.

Note: When you configure a routed port back to a switch port, it does not restore the Layer 2 configuration it had before it was changed to a routed port.

Configuring a Routed Port

Use only the ISCLI to configure routed ports. Configurations made using SNMP cannot be saved or applied.

You cannot configure a management interface to be a routed port.

Following are the basic steps for configuring a routed port:

1. Enter the interface configuration mode for the port (for this example, ethernet interface 1/12 is used).

```
Switch(config)# interface ethernet 1/12
```

2. Enable routing.

```
Switch(config-if)# no bridge-port
```

3. Assign an IP address.

   a. an IPv4 address:

```
Switch(config-if)# ip address <IPv4 address>/<prefix length>
```

   b. an IPv6 address:

```
Switch(config-if)# ipv6 address <IPv6 address>/<prefix length>
```

4. Optionally, make sure the interface is in uplink state:

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

5. Optionally, you can set the maximum transmission unit (MTU) size in bytes for received or sent frames for the routed port:

```
Switch(config-if)# mtu <64-9216>
```

Note: The default value is 1500 bytes.

6. Check the interface configuration:

```
Switch(config-if)# display interface ethernet 1/12

Interface Ethernet1/12
  Hardware is Ethernet  Current HW addr: a897.dcde.2501
  Physical:a897.dcde.250e  Logical:(not set)
  index 11 metric 1 MTU 1500 Bandwidth 10000000 Kbit
  no bridge-port
  arp ageing timeout 1500
  <UP,BROADCAST,MULTICAST>
  VRF Binding: Not bound
  Speed 10000 Mb/s Duplex full
  DHCP client is disabled.
...
```

Configuring OSPF on Routed Ports

The following OSPF configuration commands are supported on routed ports:

```
Switch(config-if)# ip ospf ?

  A.B.C.D             Address of interface
  authentication      Enable authentication
  authentication-key  Authentication password (key)
  bfd                 Bidirectional Forwarding Detection (BFD)
  cost                Interface cost
  database-filter     Filter OSPF LSA during synchronization and
                      flooding
  dead-interval       Interval after which a neighbor is declared dead
  hello-interval      Time between HELLO packets
  message-digest-key  Message digest authentication password (key)
  mtu                 OSPF interface MTU
  mtu-ignore          Ignores the MTU in DBD packets
  network             Network type
  passive-interface   Suppress routing updates on an interface or on all
                      interfaces
  priority            Router priority
  retransmit-interval Time between retransmitting lost link state
                      advertisements
  shutdown            Shutdown OSPF
  transmit-delay      Link state transmit delay

Switch(config-if)# ip router ospf 0 ?

  area       Set the OSPF area ID
  multi-area  Set the multi-area-adjacency
```

See Chapter 23, "Open Shortest Path First," for details on the OSPF protocol and its configuration.

For a full description of the above OSPF commands, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

Note: OSPFv3 cannot be configured on routed ports.

OSPF Configuration Example

The following example includes the basic steps for configuring OSPF on a routed port:

1. Enable OSPF on the switch and configure the OSPF router ID:

```
Switch(config)# routing-protocol ospf
Switch(config-router)# router-id <IPv4 address>
Switch(config-router)# exit
Switch(config)#
```

2. Make the routed port part of OSPF area 0:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# ip router ospf 0 area 0
```

Chapter 19. Address Resolution Protocol

The Address Resolution Protocol (ARP) is a protocol that maps an IPv4 address to a MAC address. ARP uses a request and reply mechanism to determine the MAC address of the destination host.

This section discusses the following topics:

ARP Overview

When an IP packet is sent across the local network, from one device to another, the source host must know the MAC address of the destination host. If the address is known, the source device encapsulates the IP packet in a layer 2 frame containing the destination MAC address and then transmits the packet over the LAN. If the MAC address is unknown, the source device uses ARP to determine it.

The protocol sends an ARP broadcast message across the local network, requesting the MAC address of the device with the specified destination IP address. When the owner of the IP address receives the ARP broadcast, it will send the requesting device an ARP unicast message containing its MAC address. The source device will store the received MAC address in its ARP table and start sending traffic. The source MAC address is also kept in the ARP table of the destination device.

MAC addresses learned using ARP are kept in an ARP table, which is independent of the Forwarding Database (FDB). The ARP tables consists of MAC addresses and their associated IP addresses. The ARP table can contain up to a maximum of 48.000 entries.

To view the contents of the ARP table, use the following command:

```
Switch> display ip arp
```

ARP only operates in the boundaries of a single network and it is never routed across network nodes.

Notes:

- ARP only works in IPv4 local networks. In IPv6 environments, the functionality of ARP is provided by the Neighbor Discovery Protocol (NDP). For more details, see "Neighbor Discovery" on page 424.

- ARP can only be configured on routed ports or switch virtual interfaces (SVI). When a routed port using ARP changes to a switch port, any ARP entries (dynamic or static) associated with that port are removed from the ARP table.

ARP Aging Timer

Because the ARP table has a limited size, ARP periodically removes unused entries by assigning each entry an aging timer. When the timer expires, the entry is refreshed by sending an ARP request. If ARP does not receive any replies for three consecutive requests, the entry is removed from the ARP table.

The aging timer of an ARP entry is reset if the switch receives an IP packet from the device mapped to that entry before the timer expires.

To configure the aging time of ARP entries, in seconds, use the following command:

```
Switch(config)# ip arp timeout <60-28800>
```

By default, the aging time of ARP entries is 1500 seconds (25 minutes). To reset the aging time to its default value, use the following command:

```
Switch(config)# no ip arp timeout
```

Note: The aging time of ARP entries can be also individually configured for each routed port or switch virtual interface (SVI).

Note: Static ARP entries are permanent and are not affected by aging.

To view the aging time of ARP entries, use the following command:

```
Switch> display ip arp
```

ARP Inspection

Access Control Lists (ACLs) can be configured to filter ARP packets received on VLANs. An ARP ACL is used to deny or permit ARP requests or replies based on various fields from the ARP packet. For more details on how to configure an ARP ACL, see Chapter 7, "Access Control Lists".

To enable or disable ARP inspection, use the following command:

```
Switch(config)# [no] ip arp inspection filter <ARP ACL name> vlan <VLAN ID
(1-4093)>
```

For example:

```
Switch(config)# ip arp inspection filter arp-acl-22 vlan 100-105
```

To view the configured ARP filters, use the following command:

```
Switch> display ip arp inspection
```

Static ARP Entries

Up to a maximum of 512 static ARP entries can be configured on the switch.

Static ARP entries do not have an aging timer associated with them. They are permanently kept in the ARP table until they are manually removed. A static ARP entry will overwrite any dynamic entries associated with the specified IP address.

Static ARP entries are configured individually on each routed port of the switch.

To add a static ARP entry, use the following command:

```
Switch(config-if)# ip arp <IPv4 address> <MAC address>
```

To remove a static ARP entry, use the following command:

```
Switch(config-if)# no ip arp <IPv4 address>
```

Static ARP Configuration Example

To add a static ARP entry, use the following steps:

1. Configure the ethernet interface as a routed port (for this example, ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# no bridge-port
```

2. Configure the IPv4 address of the routed port:

```
Switch(config-if)# ip address <IPv4 address>/<prefix length>
```

For example:

```
Switch(config-if)# ip address 10.0.0.1/24
```

3. Configure the static ARP entry:

```
Switch(config-if)# ip arp <IPv4 address> <MAC address>
```

Note: The IP address used for the static ARP entry must be part of the same network as the routed port.

For example:

```
Switch(config-if)# ip arp 10.0.0.93 0080.c8e8.1efc
Switch(config-if)# exit
```

4. Verify the ARP table:

```
Switch(config)# exit
Switch# display ip arp static
```

ARP Entry States

ARP assigns a state to each entry in the ARP table. Depending on the state of each entry, ARP will take certain actions.

The following list describes each state and the actions taken by ARP:

- NOARP - is used for neighbors that do not require ARP, like interfaces connect using Point-to-Point Protocol (PPP).

- PERMANENT - is used for static ARP entries. ARP entries with this state are not affected by aging and are not removed from the ARP table, unless they are manually deleted.

- INCOMPLETE - is used for unmapped IP addresses. An ARP request has been sent and the protocol is waiting for an ARP reply, containing the MAC address that needs to be assigned to the incomplete entry.

- REACHABLE - is used for mapped IP addresses. When an ARP reply is received, the MAC address is associated with the destination IP address and the ARP entry is now complete. The aging timer is started for that entry. If an IP packet is received from a device mapped to an ARP entry before its aging timer expires, the timer is reset.

- STALE - is used when the base reachable timer for an ARP entry has expired. Although the entry is still present in the ARP table, its reachability has not been confirmed for the duration of the timer. The next time a packet is sent to the owner of the MAC address associated with a stale ARP entry, the reachability verification process will be started.

- DELAY - is used when an IP packet needs to be sent to a device mapped to a stale ARP entry. This is temporary, until the switch sends an ARP request.

- PROBE - is used when the protocol rechecks an expired entry. An ARP request is sent to determine the MAC address of the destination IP address.

- FAILED - is used for ARP entries that could not be rechecked. If the switch does not receive any ARP replies after three consecutive requests, the entry is marked as failed and it is removed when the ARP table is refreshed.

To view the state of ARP entries, use the following command:

```
Switch> display ip arp
```

ARP Table Refresh

Each ARP entry is checked periodically to determine its state. Based on the entry's state, ARP undertakes certain actions, like refreshing the entry or removing it from the ARP table.

An ARP table refresh can be manually triggered. The switch will recheck each dynamic ARP entry by placing it in the probe state and sending an ARP request. If the switch does not received any ARP replies after three consecutive requests, it places the ARP entry in the incomplete state. If the switch receives an ARP reply, the entry is placed in the reachable state. After all the ARP entries are checked, the switch removes the incomplete entries from the ARP table.

To manually trigger a refresh of the ARP table, use the following command:

```
Switch# remove ip arp
```

To enable refreshing the ARP table right after it expires, use the following command:

```
Switch(config)# ip arp timeout refresh
```

Note: This command does not reset the timeout to the default value.

To disable refreshing the ARP table right after it expires, use the following command:

```
Switch(config)# no ip arp timeout refresh
```

To disable refreshing the ARP table right after it expires and reset the timeout value to the default value, use the following command:

```
Switch(config)# no ip arp timeout
```

To delete all dynamic entries from the ARP table, use the following command:

```
Switch# remove ip arp force-delete
```

To display current configuration of ARP refresh and timeout, use the following command:

```
Switch# display ip arp
```

Proxy ARP

Proxy ARP is a technique in which a device on a given network answers ARP requests intended for another device. The proxy ARP is aware of the location of the traffic's destination and offers its own MAC address as the destination. The received traffic is then routed by the proxy device to the intended destination via another interface or a tunnel. Proxy ARP is primarily used when hosts in the connected subnet are separated by features such as a private VLAN. Proxy ARP is defined in RFC 1027.

CNOS supports restricted proxy ARP mode. This feature enables the proxy device to respond to ARP requests if the following are both true:

- The destination IP address is not on the same physical network as the source of the request.
- The proxy device has a route to the target address of the ARP request.

Proxy ARP Limitations

The following limitations apply to proxy ARP:

- Proxy ARP can only be enabled on routed interfaces (Ethernet or VLAN); it cannot be enabled on management or loopback interfaces.
- The proxy device must have an active route to the destination address of the ARP request.

Configure Proxy ARP

Proxy ARP is disabled by default. It can be enabled or disabled in Interface Configuration mode for a specific interface. To enable or disable proxy ARP, use the command:

```
Switch(config-if)# [no] ip proxy-arp
```

Chapter 20. Internet Protocol Version 6

Internet Protocol version 6 (IPv6) is a network layer protocol intended to expand the network address space. IPv6 is a robust and expandable protocol that meets the need for increased physical address space. The switch supports the following RFCs for IPv6-related features:

- RFC 1981

- RFC 2460

- RFC 4291

- RFC 4429

- RFC 4861

- RFC 4862

- RFC 4443

The following topics are discussed in this section:

IPv6 Address Format

The IPv6 address is 128 bits (16 bytes) long and is represented as a sequence of eight 16-bit hex values, separated by colons.

Each IPv6 address has two parts:

● Subnet prefix representing the network to which the interface is connected

● Local identifier, either derived from the MAC address or user-configured

The preferred hexadecimal format is as follows:

```
XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX
```

Example IPv6 address:

```
FEDC:BA98:7654:BA98:FEDC:1234:ABCD:5412
```

Some addresses can contain long sequences of zeros. A single contiguous sequence of zeros can be compressed to :: (two colons). For example, consider the following IPv6 address:

```
FE80:0:0:0:2AA:FF:FA:4CA2
```

The address can be compressed as follows:

```
FE80::2AA:FF:FA:4CA2
```

Unlike IPv4, a subnet mask is not used for IPv6 addresses. IPv6 uses the subnet prefix as the network identifier. The prefix is the part of the address that indicates the bits that have fixed values or are the bits of the subnet prefix. An IPv6 prefix is written in address/prefix-length notation. For example, in the following address, 64 is the network prefix:

```
21DA:D300:0000:2F3C::/64
```

IPv6 addresses can be either user-configured or automatically configured. Automatically configured addresses always have a 64-bit subnet prefix and a 64-bit interface identifier. In most implementations, the interface identifier is derived from the switch's MAC address, using a method called EUI-64.

Most Cloud NOS 10.6 features permit IP addresses to be configured using either IPv4 or IPv6 address formats. Throughout this manual, *IP address* is used in places where either an IPv4 or IPv6 address is allowed. In places where only one type of address is allowed, the type (IPv4 or IPv6) is specified.

IPv6 Address Types

IPv6 supports three types of addresses:

- unicast (one-to-one)
- multicast (one-to-many)
- anycast (one-to-nearest)

Multicast addresses replace the use of broadcast addresses.

Unicast Address

Unicast is a communication between a single host and a single receiver. Packets sent to a unicast address are delivered to the interface identified by that address. IPv6 defines the following types of unicast address:

- Global Unicast address: An address that can be reached and identified globally. Global Unicast addresses use the high-order bit range up to FF00, therefore all non-multicast and non-link-local addresses are considered to be global unicast. A manually configured IPv6 address must be fully specified. Autoconfigured IPv6 addresses are comprised of a prefix combined with the 64-bit EUI. RFC 4291 defines the IPv6 addressing architecture.

    The interface ID must be unique within the same subnet.

- Link-local unicast address: An address used to communicate with a neighbor on the same link. Link-local addresses use the format `FE80::EUI`

    Link-local addresses are designed to be used for addressing on a single link for purposes such as automatic address configuration, neighbor discovery, or when no routers are present.

    Routers must not forward any packets with link-local source or destination addresses to other links.

Multicast

Multicast is communication between a single host and multiple receivers. Packets are sent to all interfaces identified by that address. An interface may belong to any number of multicast groups.

A multicast address (FF00 - FFFF) is an identifier for a group interface. The multicast address most often encountered is a solicited-node multicast address using prefix `FF02::1:FF00:0000/104` with the low-order 24 bits of the unicast or anycast address.

The following well-known multicast addresses are pre-defined. The group IDs defined in this section are defined for explicit scope values, as follows:

```
FF00:::::::0  through  FF0F:::::::0
```

Anycast

Packets sent to an anycast address or list of addresses are delivered to the nearest interface identified by that address. Anycast is a communication between a single sender and a list of addresses.

Anycast addresses are allocated from the unicast address space, using any of the defined unicast address formats. Thus, anycast addresses are syntactically indistinguishable from unicast addresses. When a unicast address is assigned to more than one interface, thus turning it into an anycast address, the nodes to which the address is assigned must be explicitly configured to know that it is an anycast address.

IPv6 Interfaces

Each IPv6 interface supports multiple IPv6 addresses. You can manually configure up to 1024 IPv6 addresses for each interface.

You can manually configure up to 1024 IPv6 addresses for each interface, as follows:

- Initial IPv6 address is a global unicast or anycast address.

```
Switch(config)# interface {<interface name>|ethernet <slot/chassis number>
|loopback <interface number>|mgmt <interface number>|port-aggregation <LAG number>|
|vlan <VLAN ID (1-4093)>}
Switch(config-if)# ipv6 address <IPv6 address>
```

- Second IPv6 address can be a unicast or anycast address.

```
Switch(config-ip-if)# ipv6 address <IPv6 address> secondary
Switch(config-ip-if)# exit
```

Each IPv6 address can belong to only one VLAN interface. Each VLAN interface can support multiple IPv4/IPv6 addresses.

Neighbor Discovery

The switch uses Neighbor Discovery protocol (ND) to gather information about other router and host nodes, including the IPv6 addresses. Host nodes use ND to configure their interfaces and perform health detection. ND allows each node to determine the link-layer addresses of neighboring nodes and to keep track of each neighbor's information. A neighboring node is a host or a router linked directly to the switch. The switch supports Neighbor Discovery as described in RFC 4861.

You can configure up to 256 IPv6 static neighbors for each interface.

Neighbor Discovery Overview

Neighbor Discover messages allow network nodes to exchange information, as follows:

- *Neighbor Solicitations* allow a node to discover information about other nodes.

- *Neighbor Advertisements* are sent in response to Neighbor Solicitations. The Neighbor Advertisement contains information required by nodes to determine the link-layer address of the sender, and the sender's role on the network.

- IPv6 hosts use *Router Solicitations* to discover IPv6 routers. When a router receives a Router Solicitation, it responds immediately to the host.

- Routers uses *Router Advertisements* to announce its presence on the network, and to provide its address prefix to neighbor devices. IPv6 hosts listen for Router Advertisements, and uses the information to build a list of default routers. Each host uses this information to perform autoconfiguration of IPv6 addresses.

- *Redirect messages* are sent by IPv6 routers to inform hosts of a better first-hop address for a specific destination. Redirect messages are only sent by routers for unicast traffic, are only unicast to originating hosts, and are only processed by hosts.

ND configuration for general advertisements, flags, and interval settings, as well as for defining prefix profiles for router advertisements, is performed on a per-interface basis using the following commands:

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config)# interface vlan <VLAN ID (1-4093)>
Switch(config-if)# [no] ipv6 nd <arguments>
Switch(config-if)# exit
```

To add or remove entries in the static neighbor cache, use the following command:

```
Switch(config-if)# [no] ipv6 neighbor <IPv6 address>
```

To view the neighbor discovery information for the specified interface, use the following command:

```
Switch(config)# display ipv6 nd interface <name>
```

To clear the neighbor cache, use the following command:

```
Switch# remove ipv6 neighbor
```

Router Nodes

Each IPv6 interface can be configured as a router node. A router node's IP address is configured manually. Router nodes can send Router Advertisements. To configure the IPv6 interface as a routed port, see Chapter 18, "Routed Ports".

Note: When IPv6 forwarding is turned on, all IPv6 interfaces configured on the switch can forward packets.

Neighbor Table Threshold

When the common Neighbor Table (NT) is learned via neighbor protocol, the item is synchronized to the hardware services layer module and to the chip NT.

A packet cannot be routed through Layer 3 via a link-local neighbor because IPv6 link-local addresses are only used locally, so link-local neighbors do not need to be stored in the switch chip NT. While the hardware layer is receiving NT entries from the common NT, cache amount and threshold value are compared. If the switch cache amount is larger, synchronization will terminate and the item will be deleted from the common NT and synchronized to the switch's chip NT.

To prevent the NT from expanding beyond chip capacity, the NT threshold value is equal to the global threshold value. Table 37 shows the switch chip capacities and NT threshold values for devices supported by CNOS.

Table 37. *Switch Chip Capacities and Neighbor Table Threshold Values*

Switch	Switch Chip Capacity	Threshold value
G8272	40960 (40K)	40000
G8296	40960 (40K)	40000
G8332	40960 (40K)	40000
NE1032	34816 (33K)	32000
NE1032T	34816 (33K)	32000
NE1072T	34816 (33K)	32000
NE10032	20480 (20K)	16000
NE2572	20480 (20K)	16000

Some space needs to be reserved in the chip NT for the local Layer 3 interface and to prevent hash conflicts in the chip NT. The common NT capacity is 200K, which si shared by global and link-local NTs. The reserved space prevents triggering garbage collection early in building the common NT.

To view all IPv6 NT entries, use the command:

```
Switch> display ipv6 neighbor global
```

Supported Applications

The following applications have been enhanced to provide IPv6 support.

- **Ping**

  The **ping6** command supports IPv6 addresses. Use the following format to ping an IPv6 address:

  ping6 *<IPv6 address>* **[vrf {default|management}] [interface** *<destination interface>***|source** *<IPv6 source address>***] [count** *<number of pings>***] [interval** *<delay time between packets>***] [packet-size** *<length>***] [timeout** *<interval>***]**

- **Traceroute**

  The **traceroute6** command supports IPv6 addresses (but not link-local addresses). Use the following format to perform a traceroute to an IPv6 address:

  traceroute6 *<IPv6 address>* **[vrf {default|management}] [interface** *<destination interface>***] [source** *<IPv6 source>***]**

- **Telnet**

  The **telnet6** command supports IPv6 addresses (but not link-local addresses). Use the following format to Telnet to IPv6 address:

  telnet6 *<IPv6 address>* **[vrf {default|management}] [port** *<port number>***]**

- **SSH**

  Secure Shell (SSH) connections over IPv6 are supported (but not link-local addresses). The following syntax is required from the client:

  ssh6 *<IPv6 address>* **[vrf {default|management}] [port** *<port number>***]**

- **TFTP**

  The TFTP commands support both IPv4 and IPv6 addresses. Link-local addresses are not supported.

- **FTP**

  The FTP commands support both IPv4 and IPv6 addresses. Link-local addresses are not supported.

Configuration Guidelines

When you configure an interface for IPv6, consider the following guidelines:

- A single interface can accept multiple IPv6 addresses.

- Reserved IPv6 addresses are not supported on the interface (for example, 1::1/64).

- Health checks are not supported for IPv6 gateways.

- IPv6 interfaces support Path MTU Discovery. The CPU's MTU is fixed at 1500 bytes.

- Support for jumbo frames (1,500 to 9,216 byte MTUs) is limited. Any jumbo frames intended for the CPU must be fragmented by the remote node. The switch can re-assemble fragmented packets up to 9k. It can also fragment and transmit jumbo packets received from higher layers.

IPv6 Configuration Examples

This section provides steps to configure IPv6 on the switch.

IPv6 Example 1

Use the following example to configure IPv6 neighbor discovery prefix settings on the router.

1. Enable IPv6 on an interface.

```
Switch(config)# interface ethernet 1/2
Switch(config-if)# no bridge-port
Switch(config-if)# ipv6 address 12ef:4533::0f30/64
```

Neighbor discovery is automatically enabled.

2. Advertise the IPv6 prefix in the router-advertisement messages.

```
Switch(config-if)# ipv6 nd prefix
```

3. Verify the interface configuration.

```
Switch(config-if)# display ipv6 nd interface
```

IPv6 Example 2

Use the following example to configure IPv6 neighbor discovery reachable time.

1. Enable IPv6 on an interface.

```
Switch(config)# interface ethernet 1/2
Switch(config-if)# no bridge-port
Switch(config-if)# ipv6 address ee33:a239:22::bcd3/32
```

Neighbor discovery is automatically enabled.

2. Configure the time when a node considers a neighbor to be up.

```
Switch(config-if)# ipv6 nd reachable-time 10
```

3. Verify the configuration.

```
Switch(config-if)# display ipv6 nd interface
```

IPv6 Limitations

Most other Lenovo Cloud Network Operating System 10.6 features permit IP addresses to be configured using either IPv4 or IPv6 address formats. However, the following switch features support IPv4 only:

● Open Shortest Path First (OSPF) version 2.

Chapter 21. Internet Group Management Protocol

Internet Group Management Protocol (IGMP) is used by IPv4 Multicast routers to learn about the existence of host group members on their directly attached subnet. The IPv4 multicast routers get this information by broadcasting IGMP Membership Queries and listening for IPv4 hosts reporting their host group memberships. This process is used to set up a client/server relationship between an IPv4 multicast source that provides the data streams and the clients that want to receive the data.

The switch supports three versions of IGMP:

- IGMPv1: Defines the method for hosts to join a multicast group. However, this version does not define the method for hosts to leave a multicast group. See RFC 1112 for details.

- IGMPv2: Adds the ability for a host to signal its desire to leave a multicast group. See RFC 2236 for details.

- IGMPv3: Adds support for source filtering by which a host can report interest in receiving packets only from specific source addresses or from all but specific source addresses, sent to a particular multicast address. See RFC 3376 for details.

The switch can perform IGMP Snooping and supports both static and dynamic IGMP groups and multicast routers. The switch can act as a Querier and participate in the IGMP Querier election process.

The following topics are discussed in this chapter:

IGMP Terms

The following are commonly used IGMP terms:

- Multicast traffic: Flow of data from one source to multiple destinations.

- Group: A multicast stream to which a host can join. Multicast groups have IP addresses in the range: 224.0.2.0 to 239.255.255.255.

- IGMP Querier: A router or switch in the subnet that generates *Membership Queries.*

- IGMP Snooper: A Layer 3 device that forwards multicast traffic only to hosts that are interested in receiving multicast data. This device can be a router or a Layer 3 switch.

- Multicast Router: A router configured to make routing decisions for multicast traffic. The router identifies the type of packet received (unicast or multicast) and forwards the packet to the intended destination.

- Membership Report: A report sent by the host that indicates an interest in receiving multicast traffic from a multicast group.

- Leave: A message sent by the host when it wants to leave a multicast group.

- Fast Leave: A process by which the switch stops forwarding multicast traffic to a port as soon as it receives a Leave message.

- Membership Query: Message sent by the Querier to verify if hosts are listening to a group.

- General Query: A *Membership Query* sent to all hosts. The Group address field for general queries is 0.0.0.0 and the destination address is 224.0.0.1.

- Group-specific Query: A *Membership Query* sent to a specific multicast group.

- Group-and-Source-Specific Query: A *Membership Query* sent to a specific multicast address from any of a specified list of sources.

How IGMP Works

When IGMP is not configured, switches forward multicast traffic through all ports, increasing network load. When IGMP is configured on a switch, multicast traffic flows as follows:

- A server sends multicast traffic to a multicast group.

- The multicast router sends *Membership Queries* to the switch, which forwards them to all ports in a given VLAN.

- Hosts respond with *Membership Reports* if they want to join a group. The switch forwards these reports to the multicast router.

- The switch forwards multicast traffic only to hosts that have joined a group and to the multicast router.

- The multicast router periodically sends *Membership Queries* to ensure that a host wants to continue receiving multicast traffic. If a host does not respond, the IGMP Snooper stops sending traffic to the host.

- An IGMPv2 host can initiate the Leave process by sending an IGMPv2 Leave packet to the IGMP Snooper.

- When a host sends an IGMP Leave packet, the IGMP Snooper sends *Group-specific Queries* to find out if any other host connected to the port is interested in receiving the multicast traffic. If it does not receive a Join message in response, the IGMP Snooper removes the group entry and passes on the information to the multicast router.

The switch supports the following:

- IGMP version 1, 2, and 3

- 64 static multicast routers

- 64 dynamic multicast routers

- up to 128 static IGMP groups

- 8191 dynamic IGMP groups, if there are no reserved IGMP static group entries, or 8063, if there are 128 reserved IGMP static group entries

Note: Unknown multicast traffic is sent to all ports if the IPMC flood option is enabled and no Membership Report was learned for that specific IGMP group. If the flood option is disabled, unknown multicast traffic is discarded if no hosts are learned on a switch.

To enable or disable IP multicast (IPMC) flood, use the following commands:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# [no] flood [ipv4|ipv6]
```

For more details, see "IPMC Flooding" on page 236.

IGMP Capacity and Default Values

The following table lists the maximum and minimum values of the switch variables.

Table 38. *Switch IGMP Variables Table*

Variable	Maximum
VLANs - Snooping	4092
Static multicast routers	64
Dynamic multicast routers	64
Static Group - Snooping	128
Dynamic Groups - Snooping	8191

The following table lists the default settings for IGMP features and variables.

Table 39. *IGMP Default Configuration Settings*

Field	Default Value
IGMP Snooping	Enabled for all VLANs
IGMP Snooping Version	3
IPMC Flood	Enabled for all VLANs
IGMP FastLeave	Disabled for all VLANs
IGMP multicast router Timeout	255 Seconds
IGMP Group Timeout	260 Seconds
IGMP Report Suppression	Enabled for all VLANs
IGMP Query-Interval Variable	125 Seconds
IGMP Query-Max-Response-Time Variable	10 Seconds
IGMP Last-Member-Query-Interval Variable	1 Second
IGMP Robustness Variable	2

Note: The IGMP multicast router Timeout is calculated as:

$$robustness \times query\ interval + \frac{query\ max\ response\ time}{2} = 2 \times 125 + \frac{10}{2} = 255\ seconds$$

Note: The IGMP Group Timeout is calculated as:

$$robustness \times query\ interval + query\ max\ response\ time = 2 \times 125 + 10 = 260\ seconds$$

IGMP Snooping

IGMP Snooping allows a switch to listen to the IGMP conversation between hosts and multicast routers. With IGMP Snooping enabled, the switch learns the ports interested in receiving multicast data and forwards it only to those ports. Thus, IGMP Snooping conserves network resources.

The switch can sense IGMP *Membership Reports* from attached hosts and acts as a proxy to set up a dedicated path between the requesting host and a local IPv4 multicast router. After the path is established, the switch blocks the IPv4 multicast stream from flowing through any port that does not connect to a host member, thus conserving bandwidth.

If the IGMP Snooping switch receives a Query, it forwards the Query to all interfaces members of the VLAN on which is was received. Any Reports sent in reply to that Query are forwarded only on interfaces with multicast routers present.

By default, IGMP Snooping is globally enabled on the switch. To globally enable or disable IGMP Snooping, use the following command:

```
Switch(config)# [no] ip igmp snooping
```

IGMP Snooping can also be enabled or disabled for each VLAN. To enable or disable IGMP Snooping on a VLAN, use the following command:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# [no] ip igmp snooping
```

Note: If IGMP Snooping is disabled globally, it will also be disabled on any VLAN, regardless of the VLAN settings.

IGMP Snooping can monitor all version and types of IGMP messages (v1, v2 and v3). If the IGMP is configured to version 2, IGMPv3 packets will be ignored by the switch.

To change the IGMP version used on a VLAN, use the following command:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# ip igmp snooping version {2|3}
```

The default IGMP version is 3. You can reset the IGMP version used by a VLAN using the following command:

```
Switch(config-vlan)# no ip igmp snooping version
```

To view the current IGMP Snooping settings, use the following command:

```
Switch> display ip igmp snooping
```

IGMPv3 Snooping

IGMPv3 includes new Membership Report messages that extend IGMP functionality. The switch provides snooping capability for all types of IGMPv3 *Membership Reports*.

IGMPv3 supports Source-Specific Multicast (SSM). SSM identifies session traffic by both source and group addresses.

The IGMPv3 implementation keeps records on the multicast hosts present in the network. If a host is already registered, when it receives a new `IS_INC, TO_INC, ALLOW, BLOCK, IS_EXC` or `TO_EXC` report from same host, the switch makes the correct transition to new (port-host-group) registration based on the IGMPv3 RFC (RFC 3376). The registrations of other hosts for the same group on the same interface are not changed.

The switch supports the following IGMPv3 filter modes:

- INCLUDE mode: The host requests membership to a multicast group and provides a list of IPv4 addresses from which it wants to receive traffic.
- EXCLUDE mode: The host requests membership to a multicast group and provides a list of IPv4 addresses from which it does not want to receive traffic. This indicates that the host wants to receive traffic only from sources that are not part of the Exclude list.

IGMPv3 Snooping is compatible with IGMPv1 and IGMPv2 Snooping.

Spanning Tree Topology Change

If a Spanning Tree topology changes happens on an interface, the switch will receive a Topology Change Notification (TCN) message. After 30 seconds from the moment the TCN message is received, the IGMP Snooping switch will delete all dynamically learned IGMP multicast groups and multicast routers associated with that interface. After deleting the IGMP entries, the switch sends General Queries on all interface/VLAN pairs belonging to the Spanning Tree Group (STG) specified in the TCN message. The reports received in reply to the General Queries will enable the switch to re-learn IGMP multicast groups.

This mechanism happens even if the switch is not elected as an IGMP Querier. In this case, the source IPv4 address of the General Queries messages will be 0.0.0.0.

This mechanism can only be configured globally for the switch and it cannot be enabled or disabled only on a certain interface.

By default, this mechanism is enabled on the switch. To enable or disable it, use the following command:

```
Switch(config)# [no] ip igmp snooping tcn flood
```

IGMP Querier

For IGMP Snooping to function, you must have a multicast router on a VLAN to generate IGMP Membership Query packets. Enabling the IGMP Querier feature on the switch allows it to participate in the Querier election process. If the switch is elected as the Querier, it will send periodic IGMP Query packets for the VLAN. Hosts that want to receive multicast traffic will respond to the Membership Query packets with IGMP Report messages, which are used by the Querier to establish an appropriate forwarding list.

Querier Election

If multiple multicast routers exist on the VLAN, only one can be elected as a Querier. The multicast routers elect the one with the lowest source IPv4 address as the Querier. The Querier performs all periodic Membership Queries. All other multicast routers (non-Queriers) do not send IGMP Query packets.

Note: When IGMP Querier is enabled on a VLAN, the switch performs the role of an IGMP Querier only if it meets the IGMP Querier election criteria.

Each time the Querier switch sends an IGMP Query packet, it initializes a *general query timer*. If a Querier receives a General Query packet from a multicast router with a lower IP address, it transitions to a non-Querier state and initializes an *other querier present timer*. When this timer expires, the multicast router transitions back to the Querier state and sends a General Query packet.

While the Querier switch is in a non-Querier state, only Query packets that have a lower IP address than the elected Querier switch will be processed. This will cause an update of the Querier information.

By default, IGMP Snooping Querier is disabled on all VLANs.

Follow the steps below for a basic IGMP Querier configuration.

1. Configure the source IPv4 address for IGMP Querier on a VLAN.

   ```
   Switch(config)# vlan <VLAN ID (1-4093)>
   Switch(config-vlan)# ip igmp snooping querier <Querier IP address>
   ```

2. Verify the configuration.

   ```
   Switch> display ip igmp snooping querier
   ```

   To disable IGMP Querier for a VLAN, use the following command:

   ```
   Switch(config-vlan)# no ip igmp snooping querier
   ```

When IGMP Querier is enabled on a VLAN, the switch waits a random amount of time between zero and four seconds, and then will send a number of consecutive General Query packets, waiting a certain period of time between each General Query. The number of General Queries is called Startup Query Count and time interval between each packet is called Startup Query Interval.

After the IGMP Querier is enabled, the random time interval the switch waits is used to avoid sending too many IGMP Queries at the same time, if IGMP Querier is simultaneously enabled on multiple VLANs.

To configure the Startup Query Count, use the following command:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# ip igmp snooping startup-query-count <1-10>
```

The default value is the same as the configured robustness value, which by default is 2 (for more information about robustness, see "Robustness Variable" on page 452). You can reset the Startup Query Count to this value by using the following command:

```
Switch(config-vlan)# no ip igmp snooping startup-query-count
```

To configure the Startup Query Interval (in seconds), use the following command:

```
Switch(config-vlan)# ip igmp snooping startup-query-interval <1-18000>
```

The default value is a quarter of the Query Interval (which by default is 125 seconds, resulting in a default value of 31 seconds for the Startup Query Interval). You can reset the Startup Query Interval to this value by using the following command:

```
Switch(config-vlan)# no ip igmp snooping startup-query-interval
```

After the switch sends out the number of General Query packets as configured by the Startup Query Count and Interval, it will use Query Interval for sending further General Query packets if is elected as Querier.

To configure the Query Interval (in seconds), use the following command:

```
Switch(config-vlan)# ip igmp snooping query-interval <1-18000>
```

Note: When setting values for the Query Interval that are higher than 128 seconds, some of these values will be automatically adjusted by the switch to be consistent with the value used within the generated query packets. When this occurs, the switch will display a logging message.

The default value is 125 seconds. You can reset the Query Interval to this value by using the following command:

```
Switch(config-vlan)# no ip igmp snooping query-interval
```

Note: This command is not the same as configuring the Query Interval to 125 seconds. For IGMPv3, the Non-Querier switches get their Robustness and Query Interval values from the elected Querier switch. When the Query Interval is configured on a Non-Querier switch, the specified value is enforced regardless if it is the same as the default value or not (in this case, the default value is the Query Interval configured on the elected Querier switch). To configure the Non-Querier switch to take on the same value as the one configured on the elected Querier, the above command must be used.

The Querier Timeout is a time interval used by Non-Querier switches. After the IGMP Querier election process is over, if a Non-Querier switch does not receive a Query during a time interval equal to twice the configured Querier Timeout, it will trigger a new IGMP Querier election.

To configure the Querier Timeout (in seconds), use the following command:

```
Switch(config-vlan)# ip igmp snooping querier-timeout <1-65535>
```

The default value is 255 seconds and you can reset the Querier Timeout to this value by using the following command:

```
Switch(config-vlan)# no ip igmp snooping querier-timeout
```

Multicast Router Discovery

For each VLAN, IGMP Snooping keeps track of switch ports that are connected to Multicast Routers (multicast routers) by checking IGMP queries sent by multicast routers.

An IGMP Snooping switch forwards multicast traffic and IGMP control messages (report and leave messages) by using information about the presence of multicast routers on switch ports.

You can also configure a static multicast router on a specific ethernet port or Link Aggregation Group (LAG). For more details, see "Static Multicast Router" on page 454.

When a IGMP query packet with 0.0.0.0 as the source IPv4 address is received, the switch learns a multicast router on the receiving interface only if IGMP Snooping Querier is disabled.

If IGMP Querier is enabled, a multicast router is learned when receiving an IGMP query packet only if both of the following conditions are met:

- the source IP address is not 0.0.0.0;

- the query's source IP address is strictly lower than the Querier's IP address.

Multicast Routers can be discovered and dynamically learned by the switch using the Protocol Independent Multicast (PIM) Hello messages. Multicast routers learned through IGMP Queries have a higher precedence than multicast routers learned through PIM Hello messages.

A multicast router dynamically learned through PIM Hello messages has an expiration timer based on the PIM Hello hold timer.

IGMP Query Messages

An IGMP Snooping switch periodically receives General Query messages from the elected IGMP Querier on a VLAN and learns that the switch port on which such queries are received is a multicast router port. The switch forwards IGMP General Query messages to all interfaces that belong to the VLAN on which the messages have been received.

The IGMP Snooping switch keeps track of the Querier version and other parameters, like the Robustness Variable or the Query Interval. This information is gathered from IGMP General Query messages. After a Spanning Tree topology change happens, the switch will send General Queries using the same IGMP version as the one used by the Querier.

When a host sends an IGMP Report indicating its interest in receiving multicast traffic, the IGMP Snooping switch forwards the Report message only using ports that have a multicast router.

To configure the maximum response time (in seconds) during which a switch can reply after received IGMP Query message, use the following command:

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# ip igmp snooping query-max-response-time <1-25>
```

The default value is 10 seconds and you can reset the maximum response time to that value by using the following command:

```
Switch(config-vlan)# no ip igmp snooping query-max-response-time
```

IGMP Groups

One IGMP entry is allocated for each unique join multicast traffic request, based on the VLAN and IGMP group address. If multiple ports join the same IGMP group using the same VLAN, only a single IGMP entry is used.

The IGMP Snooping switch keeps track of group membership with group records as: (Group, group-timer, {Filter - Include/Exclude}, {Source-records}). For IGMPv1 and IGMPv2 group membership records, the filter and source records are set to the default values. The group membership state is updated when receiving IGMPv1, IGMPv2 and IGMPv3 Report messages that force the compatibility between different IGMP versions.

You can add or remove a static member of a multicast group for an ethernet port or LAG belonging to a VLAN (for this example, ethernet interface 1/12 is used):

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# [no] ip igmp snooping static-group <multicast group IP
address> interface ethernet 1/12
```

To configure the maximum number of IGMP multicast group table entries that are reserved for IGMP static groups, use the following command:

```
Switch(config)# ip igmp snooping static-group max-limit <0-128>
```

Notes:

- Configuring the maximum number of reserved IGMP static groups can result in a failure when the following conditions are met:
  - The limit is set to a value lower than the current number of configured IGMP static groups. To set the limit to the desired value, the excess IGMP static groups must be removed.
  - The limit is set to a value higher than the number of empty IGMP multicast group table entries. To set the limit to the desired value, the excess IGMP dynamic groups must be removed.
- An IGMPv3 static or dynamic multicast group can occupy more than one entry in the IGMP multicast group table. The previous command only limits the number of table entries and not the number of IGMP static groups.
- The reserved IGMP multicast group table entries are exclusively used for IGMP static groups and cannot be used to learn IGMP dynamic groups even if there are no IGMP static groups configured on the switch.

By default, the switch does not reserve any IGMP multicast group table entries for IGMP static groups. To reset the reserved number to its default value, use the following command:

```
Switch(config)# no ip igmp snooping static-group max-limit
```

To remove all IGMP static groups entries from the IGMP multicast group table, use the following command:

```
Switch(config)# no ip igmp snooping static-group all
```

The IGMP Snooping switch uses group membership state information to configure the multicast forwarding interfaces.

Static and dynamic IGMP entries for the same or different multicast groups can be present on an interface at the same time. When a group is present, both statically and dynamically, on the same interface, only one IGMP entry will be shown in the group table with both the static (S) and dynamic (D) flags. A static multicast group will not interfere with the dynamic learning of the same group on a different interface or in the same or different VLAN.

To view all IGMP multicast groups, use the following command:

```
Switch> display ip igmp snooping groups
```

You can delete all dynamic entries from the multicast group table of an IGMP Snooping switch by using the following command:

```
Switch(config)# remove ip igmp snooping group *
```

Note: When clearing an IGMPv2 multicast group, the switch automatically generates Leave messages for those groups and sends the messages to all interface with multicast routers present.

You can delete only a certain multicast group by using the following command:

```
Switch# remove ip igmp snooping group <multicast group IPv4 address>
```

You can also delete all dynamic IGMP multicast group entries associated with a VLAN by using the following command:

```
Switch# remove ip igmp snooping group vlan <VLAN ID (1-4093)>
```

IGMP Snooping Configuration Guidelines

Consider the following guidelines when you configure IGMP Snooping:

● When multicast traffic flood is disabled, the multicast traffic sent by the multicast server is discarded if no hosts are learned on the switch.

● The multicast router periodically sends IGMP Queries.

● The switch learns the multicast router on the interface connected to the router when it sees Query messages. The switch then floods the IGMP queries on all other interfaces, including Link Aggregation Groups (LAG).

● Multicast hosts send IGMP Membership Reports as a reply to the IGMP Queries sent by the multicast router.

● When an IGMP Snooping switch receives an IGMPv2 Leave message from a group member on a certain interface, it sends an IGMPv2 Group-specific Query on that interface to determine if there any more group members interested in receiving multicast traffic. If the switch does not receive any Membership Reports during a certain time interval, it removes the interface from the multicast group and forwards the IGMPv2 Leave message to all multicast routers.

IGMP Snooping Configuration Example

This section provides steps to configure IGMP Snooping on the switch. For this example, ethernet interface 1/12 and VLAN 100 are used.

1. Create VLAN 100:

```
Switch(config)# vlan 100
Switch(config-vlan)# exit
Switch(config)#
```

2. Set ethernet interface 1/12 as a member of VLAN 100:

   a. If the interface is configured as an access switch port:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# bridge-port access vlan 100
Switch(config)# exit
```

   b. If the interface is configured as a trunk switch port:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# bridge-port trunk allowed vlan add 100
Switch(config)# exit
```

3. (Optional) Enable IGMP Snooping on VLAN 100:

```
Switch(config)# vlan 100
Switch(config-vlan)# ip igmp snooping
```

4. Disable IP multicast (IPMC) flooding:

```
Switch(config-vlan)# no flood
```

5. (Optional) Enable IGMPv3 Snooping.

```
Switch(config-vlan)# ip igmp snooping version 3
```

6. Check IGMP Snooping settings:
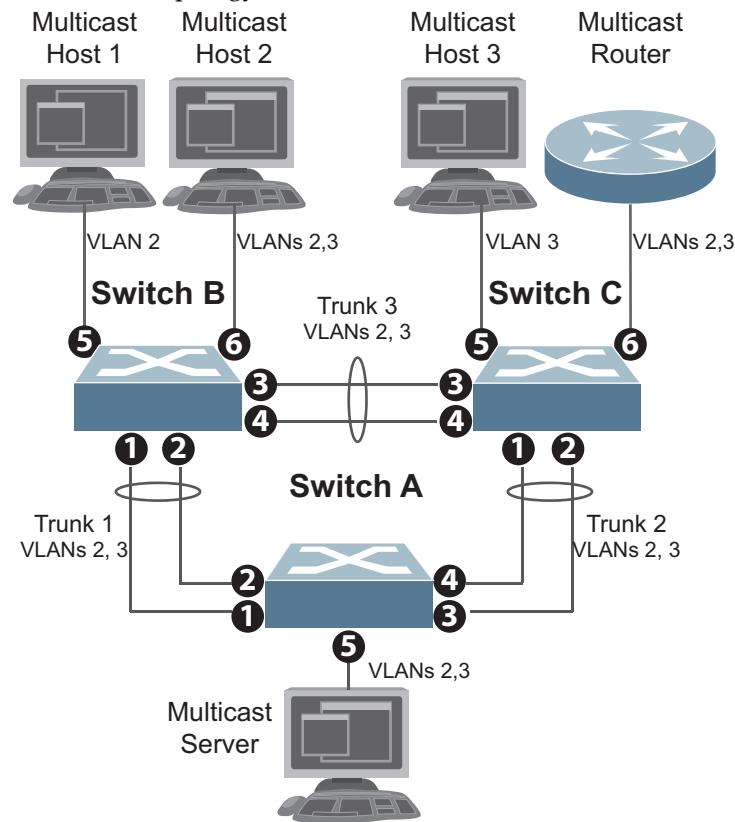
```
Switch> display ip igmp snooping

Global IGMP Snooping information
 IGMP Snooping Enabled
 IGMP Snooping V1/V2 Report Suppression Enabled
 General query transmission on TCN Enabled

IGMP Snooping information for Vlan1
 IGMP Snooping enabled
 IGMP Snooping Version: 3
 Robustness: 2 (operational: 2)
 Query Interval: 125 seconds (operational: 125 seconds)
 Group Membership Interval: 260 seconds
 Query Response Interval: 10 seconds
 Last Member Query Count: 2
 Last Member Query Interval: 1000 milliseconds
 IGMPv2 fast-leave: disabled
 IGMPv1/v2 Report suppression: enabled
 IGMPv3 Report suppression: disabled
 Router port detection using: IGMP Queries
 Snooping Querier disabled
 Querier timeout: 255 seconds (default, operational: 255 seconds)
 Querier Startup Query Count: 2
 Querier Startup Query Interval: 31 seconds
 Number of router-ports: 0
 Number of Groups: 0
 Number of Joins: 0
 Number of Leaves: 0
 Active Ports:
```

Advanced IGMP Snooping Configuration Example

Figure 12 shows an example topology. Switches B and C are configured with IGMP Snooping.

Figure 12. Topology

Devices in this topology are configured as follows:

- STG 2 includes VLAN 2; STG 3 includes VLAN 3.

- The multicast server sends IP multicast traffic for the following groups:
    - VLAN 2, 225.10.0.11 – 225.10.0.12, Source: 22.10.0.11
    - VLAN 2, 225.10.0.13 – 225.10.0.15, Source: 22.10.0.13
    - VLAN 3, 230.0.2.1 – 230.0.2.2, Source: 22.10.0.1
    - VLAN 3, 230.0.2.3 – 230.0.2.5, Source: 22.10.0.3

- The multicast router sends IGMP Query packets in VLAN 2 and VLAN 3. The multicast router's IP address is 10.10.10.10.

- The multicast hosts send the following IGMP Reports:
    - IGMPv2 Report, VLAN 2, Group: 225.10.0.11, Source: *
    - IGMPv2 Report, VLAN 3, Group: 230.0.2.1, Source: *
    - IGMPv3 IS_INCLUDE Report, VLAN 2, Group: 225.10.0.13, Source: 22.10.0.13
    - IGMPv3 IS_INCLUDE Report, VLAN 3, Group: 230.0.2.3, Source: 22.10.0.3

- The hosts receive multicast traffic as follows:
  - Host 1 receives multicast traffic for groups (*, 225.10.0.11), (22.10.0.13, 225.10.0.13)
  - Host 2 receives multicast traffic for groups (*, 225.10.0.11), (*, 230.0.2.1), (22.10.0.13, 225.10.0.13), (22.10.0.3, 230.0.2.3)
  - Host 3 receives multicast traffic for groups (*, 230.0.2.1), (22.10.0.3, 230.0.2.3)
- The multicast router receives all the multicast traffic.

Prerequisites

Before you configure IGMP Snooping, ensure you have performed the following actions:

- Configured VLANs.
- Enabled IGMP on the VLANs.
- Configured a switch or multicast router as the Querier.
- Identified the IGMP version(s) you want to enable.

IGMP Configuration

This section provides the configuration details of the switches shown in .

Switch A Configuration

1. Create VLANs 2 and 3.

```
Switch(config)# vlan 2,3
Switch(config-vlan)# exit
```

2. Configure ethernet interfaces 1/1 - 1/5 as switch trunk ports and add VLANs 2 and 3 to their allowed VLAN lists:

```
Switch(config)# interface ethernet 1/1-5
Switch(config-if-range)# bridge-port mode trunk
Switch(config-if-range)# bridge-port trunk allowed vlan 2,3
Switch(config-if-range)# exit
```

3. Assign a bridge priority lower than the default value to enable the switch to become the STP root in STG 2 and 3:

```
Switch(config)# spanning-tree mst 2 priority 4096
Switch(config)# spanning-tree mst 3 priority 4096
```

4. Configure ethernet ports 1, 2, 3 and 4 as LACP LAG members:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if-range)# aggregation-group 1 mode active
Switch(config-if-range)# exit

Switch(config)# interface ethernet 1/3-4
Switch(config-if-range)# aggregation-group 2 mode active
Switch(config-if-range)# exit
```

Switch B Configuration

1. Create VLANs 2 and 3:

```
Switch(config)# vlan 2,3
Switch(config-vlan)# exit
```

2. Configure ethernet interfaces 1/1 - 1/4 and 1/6 as switch trunk ports and add VLANs 2 and 3 to their allowed VLAN lists:

```
Switch(config)# interface ethernet 1/1-4, ethernet 1/6
Switch(config-if-range)# bridge-port mode trunk
Switch(config-if-range)# bridge-port trunk allowed vlan 2,3
Switch(config-if-range)# exit
```

3. Configure ethernet interface 1/5 as a switch access port and add VLAN 2 as its access VLAN:

```
Switch(config)# interface ethernet 1/5
Switch(config-if)# bridge-port mode access
Switch(config-if)# bridge-port access vlan 2
Switch(config-if)# exit
```

4. Configure STP edge port:

```
Switch(config)# interface ethernet 1/5-6
Switch(config-if-range)# spanning-tree port type edge
Switch(config-if-range)# exit
```

5. Configure ethernet ports 1, 2, 3 and 4 as LACP LAG members:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if)# aggregation-group 1 mode active
Switch(config-if)# exit

Switch(config)# interface ethernet 1/3-4
Switch(config-if)# aggregation-group 3 mode active
Switch(config-if)# exit
```

6. Configure IGMP Snooping:

```
Switch(config)# vlan 2,3
Switch(config-vlan)# no flood
Switch(config-vlan)# ip igmp snooping
Switch(config-vlan)# ip igmp snooping version 3
Switch(config-vlan)# exit
```

Switch C Configuration

1. Create VLANs 2 and 3:

```
Switch(config)# vlan 2,3
Switch(config-vlan)# exit
```

2. Configure ethernet interfaces 1/1 - 1/4 and 1/6 as switch trunk ports and add VLANs 2 and 3 to their allowed VLAN lists:

```
Switch(config-vlan)# interface ethernet 1/1-4, ethernet 1/6
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 2,3
Switch(config-if)# exit
```

3. Configure ethernet interface 1/5 as a switch access port and add VLAN 3 as its access VLAN:

```
Switch(config)# interface ethernet 1/5
Switch(config-if)# bridge-port mode access
Switch(config-if)# bridge-port access vlan 3
Switch(config-if)# exit
```

4. Configure STP edge ports:

```
Switch(config)# interface ethernet 1/5-6
Switch(config-if)# spanning-tree port type edge
Switch(config-if)# exit
```

5. Configure ethernet ports 1, 2, 3 and 4 as LACP LAG members:

```
Switch(config)# interface ethernet 1/1-2
Switch(config-if)# aggregation-group 2 mode active
Switch(config-if)# exit

Switch(config)# interface ethernet 1/3-4
Switch(config-if)# aggregation-group 3 mode active
Switch(config-if)# exit
```

6. Configure IGMP Snooping:

```
Switch(config)# vlan 2,3
Switch(config-vlan)# no flood
Switch(config-vlan)# ip igmp snooping
Switch(config-vlan)# ip igmp snooping version 3
Switch(config-vlan)# exit
```

Troubleshooting

This section provides the steps to resolve common IGMP Snooping configuration issues. The topology described in Figure 12 is used as an example.

Multicast traffic from non-member groups reaches the host or multicast router

● Check if traffic is unregistered. For unregistered traffic, an IGMP entry is not displayed in the IGMP groups table.

```
Switch> display ip igmp snooping groups
```

● Ensure IPMC flooding is disabled.

```
Switch(config)# vlan <VLAN ID (1-4093)>
Switch(config-vlan)# no flood
```

● Check the egress port's VLAN membership. The ports to which the hosts and multicast router are connected must be used only for VLAN 2 and VLAN 3.

```
Switch> display vlan id 2,3
```

Note: To avoid such a scenario, disable IPMC flooding for all VLANs enabled on the switches (if this is an acceptable configuration).

● Check IGMP Reports on switches B and C for information about the IGMP groups.

```
Switch> display ip igmp snooping groups
```

If the non-member IGMP groups are displayed in the table, close the application that may be sending the IGMP Reports for these groups.

Identify the traffic source by using a sniffer on the hosts and reading the source IP/MAC address. If the source IP/MAC address is unknown, check the port statistics to find the ingress port.

```
Switch> display interface ethernet <chassis number/port number> counters
```

● Ensure no static multicast MACs, static multicast groups or static multicast routers are configured.

Not all multicast traffic reaches the appropriate receivers.

● Ensure hosts are sending IGMP Reports for all the groups. Check the VLAN on which the groups are learned.

```
Switch# display ip igmp snooping groups
```

If some of the groups are not displayed, ensure the multicast application is running on the host device and the generated IGMP Reports are correct.

● Ensure multicast traffic reaches the switch to which the host is connected.

Close the application sending the IGMP Reports. Clear the IGMP groups by disabling, then re-enabling the port.

Note: To clear all IGMP groups, use the following command:

```
Switch# remove ip igmp snooping group *
```

However, this will clear all the IGMP groups and will influence other hosts.

● Ensure multicast server is sending all the multicast traffic.

● Ensure no static multicast MACs, static multicast groups or static multicast routes are configured.

IGMP queries sent by the multicast router do not reach the host.

● Ensure the multicast router is learned on switches B and C.

```
Switch> display ip igmp snooping mrouter
```

If it is not learned on switch B but is learned on switch C, check the link state of the LAG, VLAN membership and STP convergence.

If it is not learned on any switch, ensure the multicast application is running and is sending correct IGMP Query packets.

If it is learned on both switches, check the link state, VLAN membership and STP port states for the ports connected to the hosts.

IGMP Reports/Leaves sent by the hosts do not reach the multicast router

● Ensure IGMP Queries sent by the multicast router reach the hosts.

● Ensure the multicast router is learned on both switches. Note that the multicast router may not be learned on switch B immediately after a LAG failover/failback.

```
Switch> display ip igmp snooping mrouter
```

● Ensure the host's multicast application is started and is sending correct IGMP Reports/Leaves.

```
Switch> display ip igmp snooping groups
Switch> display ip igmp snooping statistics
```

A host receives multicast traffic from the incorrect VLAN

● Check port VLAN membership.

● Check IGMP Reports sent by the host.

● Check multicast data sent by the server.

The multicast router is learned on the incorrect LAG

● Check link state. LAG 1 might be down or in STP discarding state.

● Check STP convergence.

● Check port VLAN membership.

Hosts receive multicast traffic at a lower rate than normal

- Ensure a storm control is not configured on the LAGs.

```
Switch(config)# interface ethernet <chassis number/port number>
Switch(config-if)# no storm-control multicast level
```

- Check link speeds and network congestion.

Additional IGMP Features

The following topics are discussed in this section:

- "Report Suppression" on page 452
- "Robustness Variable" on page 452
- "Fast Leave" on page 453
- "Static Multicast Router" on page 454

Report Suppression

An IGMP Snooping switch will forward to a multicast router all IGMPv1 and IGMPv2 reports. To reduce IGMP traffic on a VLAN, you can enable the suppression of such reports.

Report suppression will forward only the first occurrences of IGMPv1 and IGMPv2 reports for any multicast group. Any subsequent reports from the same multicast group are discarded by the IGMP Snooping switch and not forwarded to the multicast router.

By default, report suppression is enabled on the switch for all VLANs. Report suppression can be enabled or disabled globally or for each VLAN.

To globally enable or disable report suppression, use the following command:

```
Switch(config)# [no] ip igmp snooping report-suppression
```

To enable or disable report suppression on a VLAN, use the following command:

```
Switch(config-vlan)# [no] ip igmp snooping report-suppression
```

Note: If report suppression is disabled globally, any individual VLAN configuration is ignored and report suppression is disabled for all VLANs.

Note: IGMPv3 reports cannot be suppressed.

Robustness Variable

The Robustness Variable allows tuning for the expected packet loss on a network. In this scenario, the robustness variable can be increased to reduce the impact of traffic loss.

You can increase this value by using the following command:

```
Switch(config-vlan)# ip igmp snooping robustness-variable <1-7>
```

You can reset IGMP Robustness to its default value by using the following command:

```
Switch(config-vlan)# no ip igmp snooping robustness-variable
```

Fast Leave

When an IGMP Snooping switch receives an IGMPv2 Leave message from a group member on a certain interface, it sends an IGMPv2 Group-specific Query on that interface to determine if there any more group members interested in receiving multicast traffic. If the switch does not receive any Membership Reports during a time interval called last-member-query-time, it removes the interface from the multicast group and forwards the IGMPv2 Leave message to all multicast routers.

The IGMP Snooping switch will send a particular number of IGMPv2 Group-specific Queries waiting a certain period of time between each query. The number of queries is called last-member-query-count and the period of time between each query is called last-member-query-interval.

The last-member-query-time is defined as the last-member-query-interval multiplied by the last-member-query-count. While the last-member-query-time is not directly configurable, you can configure its components.

To configure the last-member-query-interval (in seconds), use the following command:

```
Switch(config-vlan)# ip igmp snooping last-member-query-interval <1-25>
```

The default value is one second and you can reset the last-member-query-interval to this value by using the following command:

```
Switch(config-vlan)# no ip igmp snooping last-member-query-interval
```

The last-member-query-count is equal to the value of the robustness variable. To modify the robustness variable, see "Robustness Variable" on page 452.

Note: When host generates an IGMPv2 Leave message, the source IPv4 address of the message will be 0.0.0.0.

If Fast Leave is enabled on a VLAN, the IGMP Snooping switch immediately removes an interface from the multicast group when receiving an IGMPv2 Leave message on that interface and forwards the Leave message to all multicast routers.

Note: Only IGMPv2 supports Fast Leave. Enable Fast Leave on ports that have only one host connected. If more than one host is connected to a port, you may lose some hosts unexpectedly.

To enable or disable Fast Leave on a VLAN, use the following command:

```
Switch(config-vlan)# [no] ip igmp snooping fast-leave
```

Static Multicast Router

A static multicast router can be configured for a particular ethernet interface or Link Aggregation Group (LAG) on a particular VLAN. Any static multicast routers do not have to be learned through IGMP Snooping, but they can be also learned this way.

An multicast router is kept as an entry in the multicast group table of a IGMP Snooping switch. If the multicast router is both statically configured and dynamically learned, it will appear only as one IGMP entry, but will have both static (S) and dynamic (D) flags.

To add or remove a static multicast router on an interface of a VLAN, use the following command (for this example, ethernet interface 1/12 is used):

```
Switch(config-vlan)# [no] ip igmp snooping mrouter interface ethernet 1/12
```

To remove all static multicast routers from a VLAN interface, use the following command:

```
Switch(config-vlan)# no ip igmp snooping mrouter all
```

To view all static and dynamic multicast routers, use the following command:

```
Switch> display ip igmp snooping mrouter
```

Chapter 22. Border Gateway Protocol

Border Gateway Protocol (BGP) is an Internet protocol that enables routers on a network to advertise routing information with each other about the segments of the IP (IPv4 or IPV6) address space they can access within their network and with routers on external networks. BGP allows you to decide what is the "best" route for a packet to take from your network to a destination on another network rather than simply setting a default route from your border router(s) to your upstream provider(s). BGP is defined in RFC 4271.

The switch can advertise their IP interfaces and IP addresses using BGP and take BGP feeds from as many as 96 BGP router peers. This allows more resilience and flexibility in balancing traffic from the Internet.

The following topics are discussed in this section:

BGP Overview

BGP is an inter-Autonomous System routing protocol. The primary function of BGP is to exchange network reachability information with other BGP systems. This information is sufficient for constructing a graph of Autonomous System (AS) connectivity for this reachability, from which routing loops may be pruned and, at the AS level, policy decisions may be enforced.

Routing information exchanged via BGP supports the destination-based forwarding paradigm, which assumes a router forwards a packet based solely on the destination address carried in the IP header of the packet. This, in turn, determines the policy decisions that can and cannot be enforced using BGP.

BGP supports only those policies conforming to the destination-based forwarding paradigm over IPv4 as described in RFC 4271.

BGP Router Identifier

To establish BGP sessions between peers, BGP must have a router ID, which is sent to BGP peers in the OPEN message when a BGP session is established. The BGP router ID is a 32-bit value that is represented by an IPv4 address. You can configure the router ID.

By default, Cloud NOS can select a router ID among all the IP addresses of all interfaces automatically. If there are loopback interfaces with configured IPv4 address, the highest address among them is chosen as the router ID. If no loopback interfaces exist, the highest IPv4 address whose interface is in the up state will become the router ID. If that interface goes down, the current router ID is kept.

The BGP router ID must be unique to the BGP peers in a network. If BGP does not have a router ID, it cannot establish any peering sessions with BGP peers.

To configure the router ID, use the following command:

```
Switch(config)# router-id <IPv4 address>
```

To view the router ID, use the following command:

```
Switch# display router-id

Router ID: 191.1.1.1 (automatic)
```

```
Switch# display router-id

Router ID: 26.4.4.4 (config)
```

To reset the router ID to its default value (the one automatically chosen by the switch), use the following command:

```
Switch(config)# no router-id
```

Internal Routing Versus External Routing

To ensure effective processing of network traffic, every router on your network needs to know how to send a packet (directly or indirectly) to any other location/destination in your network. This is referred to as *internal routing* and can be done with static routes or using active, internal dynamic routing protocols, such as OSPF.
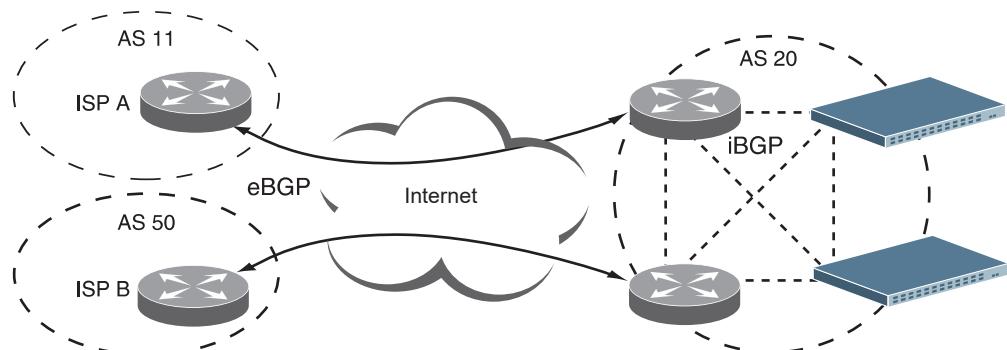
Static routes must have a higher degree of precedence than dynamic routing protocols. If the destination route is not in the route cache, the packets are forwarded to a default gateway which may be incorrect if a dynamic routing protocol is enabled.

It is also useful to tell routers outside your network (upstream providers or *peers*) about the routes you can access in your network. External networks (those outside your own) that are under the same administrative control are referred to as *autonomous systems* (AS). Sharing of routing information between autonomous systems is known as *external routing*.

External BGP (eBGP) is used to exchange routes between different autonomous systems whereas internal BGP (iBGP) is used to exchange routes within the same autonomous system. An iBGP is a type of internal routing protocol you can use to do active routing inside your network. It also carries AS path information, which is important when you are an ISP or doing BGP transit.

The iBGP peers have to maintain reciprocal sessions to every other iBGP router in the same AS (in a full-mesh manner) to propagate route information throughout the AS. If the iBGP session shown between the two routers in AS 20 was not present (as indicated in Figure 13), the top router would not learn the route to AS 50, and the bottom router would not learn the route to AS 11, even though the two AS 20 routers are connected via the switch.

Figure 13. iBGP and eBGP

When there are many iBGP peers, having a full-mesh configuration results in large number of sessions between the iBGP peers. In such situations, configuring a route reflector eliminates the full-mesh configuration requirement, prevents route propagation loops, and provides better scalability to the peers. For details, see "Route Reflector" on page 459.

Chapter 22: Border Gateway Protocol **457**

Typically, an AS has one or more *border routers*—peer routers that exchange routes with other ASs—and an internal routing scheme that enables routers in that AS to reach every other router and destination within that AS. When you *advertise* routes to border routers on other autonomous systems, you are effectively committing to carry data to the IPv4 space represented in the route being advertised. For example, if you advertise 192.204.4.0/24, you are declaring that if another router sends you data destined for any address in 192.204.4.0/24, you know how to carry that data to its destination.

Route Reflector

The Cloud NOS implementation conforms to the BGP Route Reflection specification defined in RFC 4456.

As per RFC 4271 specification, a route received from an iBGP peer cannot be advertised to another iBGP peer. This makes it mandatory to have full-mesh iBGP sessions between all BGP routers within an AS. A route reflector—a BGP router— breaks this iBGP loop avoidance rule. It does not affect the eBGP behavior.

Route reflection is an alternative to using a full mesh. This approach allows a BGP speaker known as a route reflector (RR) to advertise iBGP learned routes to certain iBGP peers. It represents a change in the commonly understood concept of iBGP, and the addition of two new optional non-transitive BGP attributes to prevent loops in routing updates.
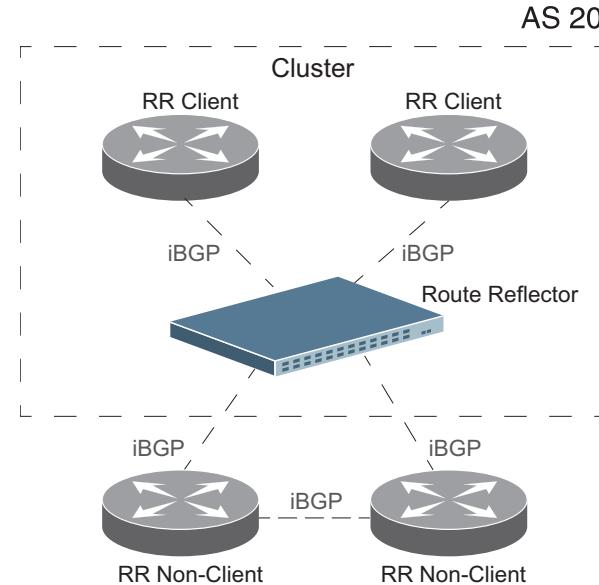
A route reflector has two groups of internal peers: clients and non-clients. A route reflector reflects between these groups and among the clients. The non-client peers must be fully meshed. The route reflector and its clients form a cluster.

When a route reflector receives a route from an iBGP peer, it selects the best path based on its path selection rule. It then does the following based on the type of peer it received the best path from:

● A route received from a non-client iBGP peer is reflected to all clients.

● A route received from an iBGP client peer is reflected to all iBGP clients and iBGP non-clients.

In Figure 14, the switch is configured as a route reflector. All clients and non-clients are in the same AS.

Figure 14. iBGP Route Reflector

The following attributes are used by the route reflector functionality:

- **Originator ID**

  BGP identifier (BGP router ID) of the route originator in the local AS. If the route does not have the originator ID attribute (it has not been reflected before), the router ID of the iBGP peer from which the route has been received is copied into the Originator ID attribute.This attribute is never modified by subsequent route reflectors. When a router identifies its own ID as the originator ID, it ignores the route.

- **Cluster list**

  Sequence of the cluster ID (the router ID) values representing the reflection path that the route has passed. The value can be configured with the following command:

  ```
  Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
  Switch(config-router)# cluster-id {<ID (1-4294967295)>|<IPv4 address>}
  ```

  The value configured with this command (or the router ID of the route reflector if cluster-id is not configured) is prepended to the cluster list attribute. When a route reflector detects its own cluster ID in the cluster list, it ignores the route.

Route Reflection Configuration Example

To configure BGP route reflection, follow the steps below:

1. Configure an AS.

   ```
   Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
   Switch(config-router)#
   ```

2. Configure a route reflector client.

   ```
   Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
   Switch(config-router-neighbor)# address-family {ipv4|ipv6} unicast
   Router(config-router-neighbor-af)# route-reflector-client
   Switch(config-router-neighbor-af)# exit
   Switch(config-router-neighbor)# exit
   Switch(config-router)# exit
   ```

Note: When a client is configured on the switch, the switch automatically gets configured as a route reflector.

3. Verify configuration.

```
Switch(config)# display ip bgp neighbors <neighbor address>

BGP neighbor is 10.1.1.2, remote AS 400, local AS 400, internal link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
  Last read 00:00:03, hold time is 180, keepalive interval is 60 seconds
  Received 121 messages, 0 notifications, 0 in queue
  Sent 123 messages, 3 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 5 seconds
 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 0
  Index 1, Offset 0, Mask 0x2
  Route-Reflector Client
  0 accepted prefixes, maximum limit 15870
  Threshold for warning message 75(%)
  0 announced prefixe
...
```

Note: The BGP neighbors displayed will only be IPv4 or IPv6 neighbors, depending upon the type of *neighbor address* you enter.

Once configured as a route reflector, the switch, by default, passes routes between clients. If required, you can disable this by using the following command:

```
Switch# config [device]
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# address-family {ipv4|ipv6} unicast
Switch(config-router-af)# no client-to-client reflection
Switch(config-router-af)# exit
Switch(config-router)# exit
```

You can view the route reflector BGP attributes attached to a BGP route using the following command:

```
Switch(config)# display ip bgp attribute-info

attr[3] nexthop 0.0.0.0
```

Restrictions

Consider the following restriction when configuring route reflection functionality:

- When a route reflector client is enabled/disabled, the session is restarted.
- When a Cluster ID is changed, all iBGP sessions are restarted.

Forming BGP Peer Routers

Two BGP routers become peers or neighbors once you establish a TCP connection between them. You can configure BGP peers statically or dynamically. While it may be desirable to configure static peers for security reasons, dynamic peers prove to be useful in cases where the remote address of the peer is unknown. For example in B-RAS applications, where subscriber interfaces are dynamically created and the address is assigned dynamically from a local pool or by using RADIUS.

For each route removed from the route table, if the route has already been sent to a peer, an update message containing the route to withdraw is sent to that peer.

For each Internet host, you must be able to send a packet to that host, and that host has to have a path back to you. This means that whoever provides Internet connectivity to that host must have a path to you. Ultimately, this means that they must "hear a route" which covers the section of the IP space you are using; otherwise, you will not have connectivity to the host in question.

BGP Peers and Dynamic Peers

A BGP speaker does not discover another BGP speaker automatically. You must configure the relationships between BGP speakers. A BGP peer is a BGP speaker that has an active TCP connection to another BGP speaker.

Cloud NOS accepts an IP address range to establish BGP sessions. For example, if you configure BGP to use IPv4 prefix 192.168.2.0/24, BGP will establishes a session with 192.168.1.2 but will reject a session from 192.168.2.2.

Static Peers

You can configure BGP static peers by using the following commands:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# address-family {ipv4|ipv6} unicast
```

Static peers always take precedence over dynamic peers. Consider the following:

- If the remote address of an incoming BGP connection matches both a static peer address and an IP address from a dynamic group, the peer is configured statically and not dynamically.

- If a new static peer is enabled while a dynamic peer for the same remote address exists, BGP automatically removes the dynamic peer.

- If a new static peer is enabled when the maximum number of BGP peers were already configured, then BGP deletes the dynamic peer that was last created and adds the newly created static peer. A syslog will be generated for the peer that was deleted.

Dynamic Peers

To configure dynamic peers, you must define a range of IP addresses (in the following format: IP address/subnet mask). BGP waits to receive an open message initiated from BGP speakers within that range. Dynamic peers are automatically created when a peer from the same subnet initiates a TCP connection. Dynamic peers are passive. When they are not in the established state, they accept inbound connections but do not initiate outbound connections.

When the BGP speaker receives an open message from a dynamic peer, the AS number from the packet must match the configured remote AS number.

All static BGP peer attributes can also be configured for dynamic peers.

To set the maximum number of dynamic peers for a group that can simultaneously be in an established state, enter the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address>/<prefix> remote-as <AS number>
Switch(config-router-neighbor)# maximum-peers <1-96>
```

If you reset this limit to a lower number, and if the dynamic peers already established for the group are higher than this new limit, then BGP deletes the last created dynamic peer(s) until the new limit is reached.

Note: The maximum number of static and dynamic peers established simultaneously cannot exceed the maximum peers (96) that the switch can support.

The following are the basic commands for configuring dynamic peers:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address>/<prefix> remote-as <AS number>
```

You cannot remove dynamic peers manually. However, you can stop a dynamic peer using the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# no neighbor <neighbor address>/<prefix>
```

The stop command interrupts the BGP connection until the peer tries to reestablish the connection.

Also, when a dynamic peer state changes from established to idle, BGP removes the dynamic peer.

Loopback Interfaces

In many networks, multiple connections may exist between network devices. In such environments, it may be useful to employ a loopback interface for a common BGP router address, rather than peering the switch to each individual interface.

Note: To ensure that the loopback interface is reachable from peer devices, it must be advertised using an interior routing protocol (such as OSPF), or a static route must be configured on the peer.

To configure an existing loopback interface for BGP neighbor, use the following commands:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address>
Switch(config-router-neighbor)# update-source loopback <0-7>
```

What is a Route Map?
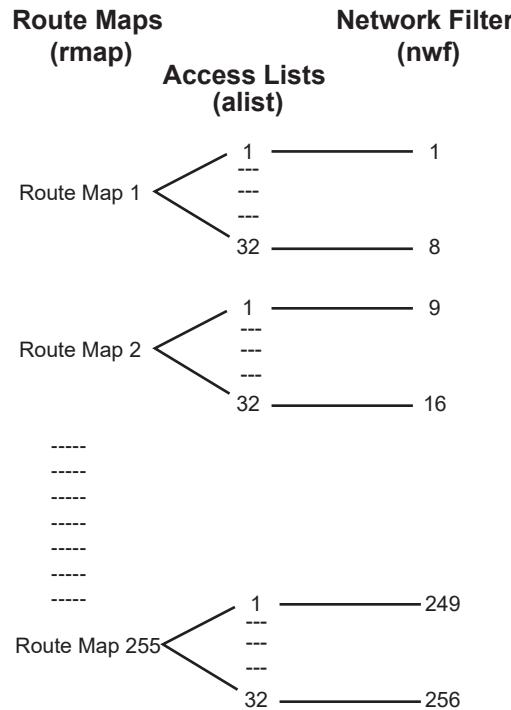
A route map is used to control and modify routing information. Route maps define conditions for redistributing routes from one routing protocol to another or controlling routing information when injecting it in and out of BGP. For example, a route map is used to set a preference value for a specific route from a peer router and another preference value for all other routes learned via the same peer router. For example, the following command is used to enter the Route Map mode for defining a route map:

```
Switch(config)# route-map <route map name>
Switch(config-route-map)# ?
```

A route map allows you to match attributes, such as metric, network address, and AS number. It also allows users to overwrite the local preference metric and to append the AS number in the AS route. See "BGP Failover Configuration" on page 491.

Cloud NOS allows you to configure up to 255 route maps. Each route map can have up to 32 access lists. Each access list consists of a network filter. A network filter defines an IP address and subnet mask of the network that you want to include in the filter. Figure 15 illustrates the relationship between route maps, access lists, and network filters.

Figure 15. Distributing Network Filters in Access Lists and Route Maps

Next Hop Peer IP Address

Next hop peer IP address can be configured only for route maps used in BGP. When a route map is applied on ingress, the next hop of learned routes is replaced with peer IP address. When applied on egress, the next hop of the redistributed routes is replaced with the local IP address.

```
Switch(config)# route-map <route map name>
Switch(config-route-map)# apply ip next-hop peer-address
```

Incoming and Outgoing Route Maps

A BGP peer router can be configured to support up to two simultaneous route maps; one incoming route map and one outgoing route map.

If a route map is not configured in the incoming route map list, the router imports all BGP updates. If a route map is configured in the incoming route map list, the router ignores all unmatched incoming updates. If you set the action to deny, you must add another route map to permit all unmatched updates.

Route maps in an outgoing route map list behave similar to route maps in an incoming route map list. If a route map is not configured in the outgoing route map list, all routes are advertised or permitted. If a route map in the outgoing route map list is set to permit, matched routes are advertised and unmatched routes are ignored.

Precedence

You can set a priority to a route map by specifying a precedence value with the following command (Route Map mode):

```
Switch(config)# route-map <route map name> <sequence number (1-65535)>
```

The smaller the value the higher the precedence. If two route maps have the same precedence value, the smaller number has higher precedence.

Configuration Overview

To configure route maps:

1. Create the IP address prefix-list, specifying the IPv4 address and subnet mask of the network that you want to match.

```
Switch(config)# ip prefix-list <prefix list name> permit <IP address>/<prefix length>
```

2. Define a network filter.

```
Switch(config)# route-map <route map name>
Switch(config-route-map)# match ip address prefix-list <prefix list name>
```

Enable the network filter. You can distribute up to 256 network filters among 255 route maps each containing 32 access lists.

3. (Optional) Configure the AS filter attributes.

```
Switch(config)# ip as-path access-list <ACL name> permit <Regular expression to match>
Switch(config)# route-map <route map name>
Switch(config-route-map)# match as-path <ACL name>
```

4. (Optional) Set up the BGP attributes.

If you want to overwrite the attributes that the peer router is sending, define the following BGP attributes:

- Specify AS paths that you want to prepend to a matched route and the local preference for the matched route.
- Specify the metric [Multi Exit Discriminator (MED)] for the matched route.

```
Switch(config-route-map)# apply as-path prepend <AS number> [<AS number> ...]
Switch(config-route-map)# apply local-preference <local preference value>
Switch(config-route-map)# apply metric <metric value>
Switch(config-route-map)# exit
```

5. Assign the route map to a peer router.

Select the peer router and then add the route map to the incoming route map list,

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address>
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# route-map <route map name> in
```

or to the outgoing route map list.

```
Switch(config-router-neighbor-af)# route-map <route map name> out
```

Aggregating Routes

Aggregation is the process of combining several different routes in such a way that a single route can be advertised, which minimizes the size of the routing table. You can configure aggregate routes in BGP either by redistributing an aggregate route into BGP or by creating an aggregate entry in the BGP routing table.

To define an aggregate route in the BGP routing table, use the following commands:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# address-family {ipv4|ipv6} unicast
Switch(config-router-af)# aggregate-address <IP address>/<prefix length>
```

The origin code for the aggregated route is either IGP or INCOMPLETE.

An example of creating a BGP aggregate route is shown in "Default Redistribution and Route Aggregation Example" on page 493.

Redistributing Routes

BGP injects local routes in two different ways:

- Using network configuration commands. These commands list networks that are candidates if they appear in the routing table.

- Using redistribution from another routing protocol.

In addition to running multiple routing protocols simultaneously, CNOS software can redistribute information from one routing protocol to another. For example, you can instruct the switch to use BGP to re-advertise static routes. This applies to all of the IP-based routing protocols.

You can also conditionally control the redistribution of routes between routing domains by defining a method known as route maps between the two domains. For more information on route maps, see "What is a Route Map?" on page 465. Redistributing routes is another way of providing policy control over whether to export OSPF routes, fixed routes, and static routes. For an example configuration, see "Default Redistribution and Route Aggregation Example" on page 493.

Default routes can be configured using the following methods:

- Import

- Originate—The router sends a default route to peers if it does not have any default routes in its routing table.

- Redistribute—Default routes are either configured through the default gateway or learned via other protocols and redistributed to peer routers. If the default routes are from the default gateway, enable the static routes because default routes from the default gateway are static routes. Similarly, if the routes are learned from another routing protocol, make sure you enable that protocol for redistribution.

- None

The origin code for the redistributed routes is INCOMPLETE.

The metrics of the non-BGP-learned routes are transferred to the metrics or MEDs of the new BGP route.

A route map can be used for each redistributed protocol to control which routes are redistributed. A route map can also be used to modify the BGP attributes of the redistributed routes. The switch can redistribute directly connected routes, static routes or routes learned through OSPF.

To configure route redistribution, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# address-family {ipv4|ipv6} unicast
Switch(config-router-af)# redistribute {direct|ospf|static} [route-map
<route map name>]
```

To view statistics for redistributed routes, use the following command:

```
Switch# display bgp {ipv4|ipv6} {unicast|multicast} policy statistics
redistribute {direct|static|all}
```

This command displays the following information:

- Route-map: The route-map name (if route-map is applied at redistribution)

- Compared count: How many static and connected routes were considered for redistribution into BGP

- Matched count: How many of these routes actually matched the route-map

- Total compared count: How many static and connected routes can be found in the routing table (including management routes)

- Total matched count: How many static and connected routes were actually redistributed by BGP (all routes minus management routes)

To clear statistics for redistributed routes, use the following command:

```
Switch# remove bgp {ipv4|ipv6} {unicast|multicast} policy statistics
redistribute {direct|static|all}
```

BGP Communities

BGP communities are attribute tags that allow controlled distribution of routing information based on an agreement between BGP peers. Communities are commonly used by transit service providers to enable peering customers to choose specific routing destinations for their outgoing routes. The transit service provider would typically publish a list of well-known or proprietary communities along with their descriptions, and take it upon itself to advertise incoming routes accordingly. For instance, an ISP may advertise that incoming routes tagged with community XY:01 will be advertised only to European peers while incoming routes tagged with community XY:02 will be advertised only to Asian peers.

The switch can be configured to manage the community tags applied to the outgoing route updates. It does not, however, modify any routing decisions based on the community tags.

Up to 32 community tags can be applied to prefixes that pass a route-map. Valid values are between 0:0 and 65535:65535. Newly added communities will replace the existing communities. To set communities to prefixes that pass the route-map, use the following commands:

```
Switch(config)# route-map <route map name>
Switch(config-route-map)# apply community <AA:NN (autonomous system number:
community number)>
```

To remove all community tags from prefixes that pass the route-map, use the following command:

```
Switch(config-route-map)# apply community none
```

To propagate prefixes that pass the route-map with their original community settings, use the following command:

```
Switch(config-route-map)# no apply community <AA:NN>
```

To enable or disable community tags forwarding for specific neighbors or neighbor groups, use the following commands:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address>[/<prefix>]
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# send-community
```

BGP Community

A *community* is a group of destinations which share some common property; this may be added to each prefix. Communities are transitive optional attributes. The community attribute can be thought of as a flat, 32-bit value that can be applied to any set of prefixes. It can be read as a 32-bit value or split into two portions, the first two bytes representing an ASN and the last two bytes as a value with a predetermined meaning. The community attribute is always used to do routing policy.

The values 0x00000000 through 0x0000FFFF and 0xFFFF0000 through 0xFFFFFFFF are reserved. Most modern router software displays communities as ASN:VALUE. In this format the communities 1:0 through 65534:65535 are available for use. The convention is to use the ASN of your own network as the leading 16 bits for your internal communities and communities with which you accept and send data.

Each autonomous system administrator may define which communities a destination belongs to. By default, all destinations belong to the general Internet community.

Three communities are defined in RFC 1997 and are standard within BGP implementations:

- **NO-EXPORT (0xFFFFFF01)**

  NO-EXPORT is commonly used within an AS to instruct routers not to export a prefix to eBGP neighbors.

- **NO-ADVERTISE (0xFFFFFF02)**

  NO-ADVERTISE instructs a BGP-speaking router not to send the tagged prefix to any other neighbor, including other iBGP routers.

- **NO-ADVERTISE-SUBCONFED (0xFFFFFF03).**

  NO-ADVERTISE-SUBCONFED is used to prevent a prefix from being advertised to external BGP peers (this includes peers in other members autonomous systems inside a BGP confederation).

BGP community attribute exchange between BGP peers is enabled when the send-community command is configured for the specified neighbor.

When multiple values are configured in the same community list statement, a logical AND condition is created. All community values must match to satisfy this AND condition. When multiple values are configured in separate community list statements, a logical OR condition is created. The first list that matches a condition is processed.

To add or remove a community list, use the following command:

```
Switch(config)# [no] ip community-list {standard|expanded} <community list
name> {deny|permit} {<AA:NN>|internet|local-as|no-advertise|no-export}
```

For example, the following command adds a community list called 'myCommList', which permits the advertising of routes to the internet community.

```
Switch(config)# ip community-list standard myCommList internet
```

To view the configuration, use the following command:

```
Switch# display ip community-list

Named Community standard list myCommList
    permit internet
```

BGP Extended Community

An upgrade to the standard community with a transitive-optional attribute (Type 16) is called the *extended community*. The current Internet RFC defines the extended community as an 8-octet value, while the community is only a 4-octet value. The first octet specifies the type; an optional second value can specify a subtype. This value dictates the structure given to the remaining octets.

The Type field gives the community some immediate flexibility. The first is the use of bit 0 to represent whether the community is registered with the IANA. The second bit gives the Extended Community a coarse scope, either Transitive, meaning it may be passed between ASs, or Non-Transitive, meaning it should be carried only within the local AS.

BGP extended community attribute exchange between BGP peers is enabled when the send-community command is configured for the specified neighbor.

When multiple values are configured in the same extended community list statement, a logical AND condition is created. All extended community values must match to satisfy this AND condition. When multiple values are configured in separate extended community list statements, a logical OR condition is created. The first list that matches a condition is processed.

To add or remove an extended community list, use the following command:

```
Switch(config)# [no] ip extcommunity-list {standard|expanded} <extended
community list name> {deny|permit} {<AA:NN>|rt|soo}
```

For example, the following commands adds an extended community list called 'denyRT-100-SOO-133', which denies routes from route target 34400:100 and site of origin 2500:133:

```
Switch(config)# ip extcommunity-list standard denyRT-100-SOO-133 deny rt
34400:100 soo 2500:133
```

To view the configuration, use the following command:

```
Switch# display ip extcommunity-list

Named extended community standard list denyRT-100-SOO-133
    deny 34400:100 2500:133
```

BGP Confederation

A fully meshed iBGP network becomes complex as the number of iBGP peers grows. You can reduce the iBGP mesh by dividing the autonomous system into multiple sub-autonomous systems and grouping them into a single confederation. A *confederation* is a group of iBGP peers that use the same autonomous system number (confederation identifier) to communicate to external networks. Each sub-autonomous system is fully meshed within itself and has a few connections to other sub-autonomous systems in the same confederation. The peers in sub-autonomous systems communicate with each other using sub-autonomous AS numbers.

To configure the confederation AS, use the following command:

```
Switch(config)# routing-protocol bgp <AS number>
Switch(config-router)# confederation identifier <confederation AS (1-65535)>
```

To delete the confederation AS, use the following command:

```
Switch(config-router)# no confederation identifier
```

To configure the peer member AS number to be part of the BGP confederation, use the following command:

```
Switch(config)# routing-protocol bgp <AS number>
Switch(config-router)# confederation peers <member AS (1-65535)> ...
```

Notes:

● You can add multiple confederation members with a single command.

● The member AS number is only visible with the BGP confederation.

BGP Path Attributes

Path attributes are the characteristics of an advertised BGP route. BGP routing policy is set and communicated using the path attributes.

There are four kinds of BGP path attributes:

- Well-known mandatory
- Well-known discretionary
- Optional transitive
- Optional non-transitive

Well-Known Mandatory

Well-known, mandatory attributes must appear in every UPDATE message. It must be supported by all BGP software implementations. If a well-known, mandatory attribute is missing from an UPDATE message, a NOTIFICATION message must be sent to the peer.

Cloud NOS supports the following well-known, mandatory attributes:

- AS_PATH: the sequence of autonomous systems (AS) through the network from the switch to the destination;
- ORIGIN: the origin of the BGP route (IGP, EGP or incomplete);
- NEXT_HOP: the IP address of the next hop on the route.

Well-Known Discretionary

Well-known, discretionary attributes may or may not appear in an UPDATE message, but they must be supported by any BGP software implementation.

Cloud NOS supports the following well-known, discretionary attributes:

- LOCAL_PREF: the preference value of a certain path;
- ATOMIC_AGGREGATE: informs the neighboring AS that the originating routes has aggregated routes.

Optional Transitive

Optional, transitive attributes may or may not be supported in all BGP implementations. If an optional, transitive attribute is sent in an UPDATE message but is not recognized by the receiver, it must be passed on to the next AS.

Cloud NOS supports the following optional, transitive attributes:

- AS4_PATH: informs that the AS number is stored in four-octet format;
- AS4_AGGREGATOR: informs that the aggregator carries an AS number in four-octet format;
- AGGREGATOR: specifies the IP address and AS number of route aggregating BGP peer;
- COMMUNITY: a common policy used by BGP peers that are in the same group;
- EXTENDED COMMUNITIES: an extended range BGP community;

Optional Non-Transitive

Optional, non-transitive attributes may not be supported and will not be passed on if advertised. If an optional, non-transitive attribute is received, the router does not have to pass it on and may safely and quietly ignore the attribute.

Cloud NOS supports the following optional, non-transitive attributes:

- MULTI_EXIT_DISC (MED): indicates the preferred entry point into an AS, when multiple entry points are available;
- ORIGINATOR_ID: the router ID of route origin;
- CLUSTER_LIST: the sequence of the CLUSTER ID (the router ID) values representing the reflection path that the route has passed;
- MP_REACH_NLRI: Multiprotocol Reachable NLRI - advertises a feasible route to a peer or permits a router to advertise the Network Layer address of the BGP peer that should be used as the next hop address to the destinations listed in the Network Layer Reachability Information field;
- MP_UNREACH_NLRI: Multiprotocol Unreachable NLRI - withdraws multiple unfeasible routes from service.

Best Path Selection Logic

A BGP speaker may have several paths to the same destination. The local speaker can select the best paths, and then install it into the routing information base (RIB). The best path selection logic includes the following:

- Exclude paths that matches any of the following conditions.
  - Paths for which the NEXT_HOP is inaccessible
  - Paths for which it's punished by damping
  - Paths that are not synchronized if synchronization is configured
  - Paths from an external BGP (eBGP) neighbor if the local autonomous system (AS) appears in the AS_PATH

- If you enable `bgp enforce-first-as` and the UPDATE does not contain the AS of the neighbor as the first AS number in the AS_SEQUENCE, the router sends a notification and closes the session.

- Paths that are marked as `received-only` in the `display ip bgp longer-prefixes` output have been rejected by the policy. However, the router has stored the paths because you have configured `soft-reconfiguration inbound` for the neighbor that sent the path.

BGP Best Path Selection

If there are multiple paths, BGP compares the best path with the next path until BGP reaches the end of valid paths. The following rules determine the best path:

- Prefer the path with the highest WEIGHT.

- Prefer the path with the highest LOCAL_PREF.

- Prefer the path that was locally originated via a network command, then the path that was locally aggregated.

- Prefer the path with the shortest AS_PATH.

- Prefer the path with the lowest origin type. Interior Gateway Protocol (IGP) is lower than Exterior Gateway Protocol (EGP), and EGP is lower than INCOMPLETE.

- Prefer the path with the lowest multi-exit discriminator (MED).

- Prefer eBGP over Internal Border Gateway Protocol (iBGP) paths.

- Prefer the path with the lowest IGP metric to the BGP next hop.

- When both paths are external, prefer the path that was received first (the oldest one).

- Prefer the route that comes from the BGP router with the lowest router ID.

- If the originator or router ID is the same for multiple paths, prefer the path with the minimum cluster list length.

- Prefer the path that comes from the lowest neighbor address.

BGP Weight

The weight is assigned locally to each individual router, and the value is only understood by that particular router. The value is neither propagated nor carried through in route updates.

A weight can be any integer from 0 to 65535. Routes with a higher weight value have preference when there are multiple routes to the same destination. Routes learned through another BGP peer have a default weight of 0, and routes sourced by the local router (the paths that the router originated) have a default weight of 32768.

To configure the weight of a BGP peer, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# weight <0-65535>
```

To remove the weight associated with a BGP peer, use the following command:

```
Switch(config-router-neighbor)# no weight
```

Local Preference

When there are multiple paths to the same destination, the local preference attribute indicates the preferred path. The path with the higher preference is preferred (the default value of the local preference attribute is 100). Unlike the weight attribute, which is only relevant to the local router, the local preference attribute is part of the routing update and is exchanged among routers in the same AS.

The following commands use the route map local preference method, which affects both inbound and outbound directions.

```
Switch(config)# route-map <route map name>
Switch(config-route-map)# apply local-preference <0-4294967295>
```

Metric (Multi-Exit Discriminator) Attribute

This attribute is a hint to external neighbors about the preferred path into an AS when there are multiple entry points. A lower metric value is preferred over a higher metric value. The default value of the metric attribute is 0.

Unlike local preference, the metric attribute is exchanged between ASs; however, a metric attribute that comes into an AS does not leave the AS.

When an update enters the AS with a certain metric value, that value is used for decision making within the AS. When BGP sends that update to another AS, the metric is reset to 0.

Unless otherwise specified, the router compares metric attributes for paths from external neighbors that are in the same AS.

Next Hop

BGP routing updates sent to a neighbor contain the next hop IP address used to reach a destination. In eBGP, the edge router, by default, sends its own IP address as the next hop address. The edge router receiving this next hop attribute, if it has any iBGP connection, will advertise this same IP address as the next hop attribute to other iBGP routers. However, this can sometimes cause routing path failures in Non-Broadcast Multiaccess Networks (NBMA). In some cases the routers inside this AS do not have a direct route to the external IP address in this next hop attribute, causing the route to not be installed in the hardware.

To avoid routing failures, you can manually configure the next hop IP address. When using NBMA networks, you can configure the external BGP speaker to advertise its own IP address as the next hop. With iBGP updates, you can configure the edge iBGP router to send its IP address as the next hop.

Next hop can be configured on a BGP peer or a peer group. Use the following commands:

● Next Hop for a BGP Peer:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor IP address>
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# next-hop-self
```

Best Path Selection Tuning

There are several ways of tuning the BGP path selection. They are available by using the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# bestpath ?

  always-compare-med        Allow comparing MED from different
                            neighbors
  as-path                   AS-path attribute
  compare-confed-aspath     Allow comparing confederation AS path
                            length
  compare-routerid          Compare router-id for identical EBGP paths
  dont-compare-originator-id Don't Compare originator-id for BGP
  med                       MED attribute
  tie-break-on-age          Prefer old route when compare-route-id is
                            not set
```

● **bestpath always-compare-med**

In best path selection section, BGP speaker compare the MED of 2 paths only when they are from same AS. When customer enables this option, BGP speaker will compare MED even from different AS.

- **bestpath med**

  Use this command to specify two MED (Multi Exit Discriminator) attributes:

  o **bestpath med confed**

  Enables MED comparison among paths learned from confederation peers. The MEDs are compared only if there is no external Autonomous System (an AS not within the confederation) in the path. If there is an external autonomous system in the path, the MED comparison is not made.

  For example in the following paths, the MED is not compared with Path3 as it is not in the confederation. MED is compared for Path1 and Path2 only.

  ```
  Path1 = 32000 32004, med=4
  Path2 = 32001 32004, med=2
  Path3 = 32003 1, med=1
  ```

  o **bestpath med missing-as-worst**

  Considers a missing MED attribute in a path as having a value of infinity, making the path without a MED value the least desirable path. If `missing-as-worst` is disabled, the missing MED is assigned the value of 0, making the path with the missing MED attribute the best path.

  o **bestpath med non-deterministic**

  When you enabled, BGP route selection process does not group routes from the same AS, the best route will depend on the order of routes receiving.

  o **bestpath med remove-recv-med**

  Removes the receive MED attribute from the MED attribute.

  o **bestpath med remove-send-med**

  Removes the send MED attribute from the MED attribute.

- **bestpath compare-routerid**

  Use this command to compare router IDs for identical eBGP paths. When comparing similar routes from peers, the BGP router does not consider the router ID of the routes. By default, it selects the first received route. This command causes BGP to include the router ID in the selection process. Similar routes are compared, and the route with lowest router ID is selected.

- **bestpath tie-break-on-age**

  Use this command to always select the older preferred route when the switch does not compare router IDS for identical eBGP paths.

- **bestpath compare-confed-aspath**

  Use this command to allow comparison of the confederation AS path length. This command specifies that the AS confederation path length must be used when available in the BGP best path decision process.

- **bestpath dont-compare-originator-id**

  Use this command to compare "not" an originator ID for an identical eBGP path.

- **bestpath as-path multipath-relax**

  Use this command to have the BGP speaker handle the paths received from different autonomous systems for multipath if their AS-path lengths are the same and all other multipath conditions are met.

- **bestpath as-path ignore**

  Ignores the AS path length when selecting a BGP route.

BGP ECMP

The BGP multipath feature is used for balancing forwarding loads on eBGP or iBGP routers when the destination route can be reached through multiple Equal Cost next hops. Equal cost is determined via the best path selection mechanism, and all paths that have equal cost are considered Multipath Candidates. To be candidates for multipath, paths to the same destination must have the following characteristics equal to the best-path characteristics:

- Weight

- Local preference

- AS path length

- Origin

- MED

- One of the following:
  - Neighboring AS or sub-AS
  - AS path

BGP Features and Functions

BGP on CNOS has the following features and functions.

AS-Path Filter

When the BGP speaker uses the `as-path` parameter to filter inbound or outbound routes, it compares the regular express of AS-path to the AS path of the inbound outbound routes. If they match and the action is `permit`, the routes are installed or sent to peers. If they match and the action is `deny`, the routes are denied or not sent to peers.

To configure the AS path filter, use the following command:

```
Switch(config)# ip as-path access-list <ACL name> {deny|permit} <regular
expression to match>
```

BGP Capability Code

A BGP speaker can negotiate capability by sending a capability code in an OPEN message when establishing the BGP session. CNOS supports the following capability codes:

Table 40. *BGP Capability Codes*

Capability Code	Capability Name
1	Multiprotocol Extensions for BGP-4
2	Route Refresh Capability for BGP-4
64	Graceful Restart Capability
65	Support for 4-octet AS number capability
67	Support for Dynamic Capability (capability specific)

Administrative Distance

An *administrative distance* is a rating of the trustworthiness of a routing information source. The administrative distance does not influence the BGP path selection algorithm, but it does influence whether BGP-learned routes are installed in the IP routing table.

TTL-Security Check

The BGP TTL Security Check feature protects the eBGP peering session by comparing the value in the TTL field of received IP packets against a hop count that is configured locally for each eBGP peering session:

- If the value in the TTL field of the incoming IP packet is greater than or equal to the locally configured value, the IP packet is accepted and processed normally.

- If the TTL value in the IP packet is less than the locally configured value, the packet is silently discarded, and no ICMP message is generated.

This is designed behavior; a response to a forged packet is unnecessary.

By default, eBGP does not support eBGP multi-hop. To enable or disable the exterior Border Gateway Protocol (eBGP) time-to-live (TTL) value to support eBGP multi-hop, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# [no] ebgp-multihop <1-255>
```

By default, there is no limit to how many hops can separate two BGP peers. To configure the maximum number of hops that separate two peers, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# ttl-security hops <1-254>
```

To reset this limit to its default value (0), use the following command:

```
Switch(config-router-neighbor)# no ttl-security hops
```

Note: When this feature is configured for a multi-hop peering session, the **ebgp-multihop** command cannot be used. If you attempt to use both commands for the same peering session, an error message will be displayed on the console.

Local-AS

The local AS feature allows a router to appear to be a member of a second autonomous system (AS) in addition to its real AS. Local AS allows two ISPs to merge without modifying peering arrangements. Routers in the merged ISP become members of the new autonomous system but continue to use their old AS numbers for their customers. You can only use the Local AS feature with true eBGP peers; you cannot use this feature for two peers that are members of different confederation sub-autonomous systems.

To enable or disable eBGP to prepend the local AS number to the AS path attribute, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# [no] local-as <AS number (1-4294967295)>
```

BGP Authentication

You can configure MD5 authentication between two BGP peers; each segment sent on the TCP connection between the peers is verified. MD5 authentication must be configured with the same password on both BGP peers or the connection between them will not be made. Configuring MD5 authentication causes the BGP speaker to generate and check the MD5 digest of every segment sent on the TCP connection.

To enable or disable BGP to use MD5 authentication when communicating with a certain neighbor, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# [no] password <neighbor password>
```

Note: An unencrypted password can be configured by using the following command:

```
Switch(config-router-neighbor)# [no] password 0 <neighbor password>
```

Note: For dynamic BGP peers, MD5 authentication is not supported.

Originate Default Route

By default, BGP sends the known default routes to neighbors, but will not originate a default route. If a route is configured instead of learned, it will not be sent. To originate default routes to neighbors, the default-originate feature must be enabled.

To enable or disable a BGP routing process to distribute a default route, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# address-family {ipv4|ipv6} unicast
Switch(config-router-neighbor-af)# [no] default-originate
```

Optionally, a route map can be specified to allow the default route to be injected conditionally. To do this, use the following command:

```
Switch(config-router-neighbor-af)# [no] default-originate route-map <route
map name>
```

IP Prefix-List Filter

There are several ways to filter BGP routes:

- IP prefix-list filter
- as-path filter
- route-map filter

When the BGP speaker uses the IP prefix list to filter inbound routes or outbound routes, it compares the prefix from the prefix list to the prefix of the inbound routes or outbound routes. If they match and the action is permit, the routes are installed or sent to peers. If they match and the action is deny, the routes are denied or not sent to peers.

To create or delete an IP prefix list, use the following command:

```
Switch(config)# ip prefix-list <prefix list name> ?

  deny         Specify packets to reject
  description  Prefix-list specific description
  permit       Specify packets to forward
  seq          Sequence number of an entry
```

For more details about the above command, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

For example, the following command adds an IP prefix list called 'pref-list-35' that forwards BGP routes that match IPv4 address 35.0.0.0 with prefix length 8:

```
Switch(config)# ip prefix-list pref-list-35 permit 35.0.0.0/8
```

To view prefix list information, use the following command:

```
Switch# display ip prefix list

ip prefix-list pref-list-35: 1 entries
    seq 5 permit 35.0.0.0/8
```

To enable or disable using IP prefix lists to filter BGP routes, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# address-family {ipv4|ipv6} unicast
Switch(config-router-neighbor-af)# [no] prefix-list <prefix list name> {in|out}
```

For example, to filter inbound BGP routes using the prefix list previously created (pref-list-35), use the following command:

```
Switch(config-router-neighbor-af)# prefix-list pref-list-35 in
```

Dynamic Capability

Dynamic Capability allows the dynamic update of capabilities over an established BGP session without disrupting the session.

By advertising Dynamic Capability to a peer in the open, a BGP speaker tells the peer that the speaker is capable of receiving and properly handling the CAPABILITY message from the peer after the BGP session has been established.

By default, Dynamic Capability is disabled. To enable or disable it on the switch, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# [no] dynamic-capability
```

BGP Graceful Restart

Usually when BGP on a router restarts, all the BGP peers detect that the session went down and then came up. This "down/up" transition is called *route flapping*. This causes BGP route re-computation, generation of BGP routing updates, and unnecessary churn to the forwarding tables. It can spread across multiple routing domains and create transient forwarding black holes and transient forwarding loops.

BGP graceful restart specifies a mechanism that allows a BGP speaker to preserve the forwarding state during BGP restart. BGP works in helper mode to help the connected router to perform a graceful restart.

The router can help the peer restart but it cannot preserve its own routing information when restarting. Thus, the peer neighbor cannot preserve the routing information advertised by a BGP router in only graceful restart helper mode.

By default, BGP graceful restart helper functionality is disabled. To enable or disable it on the switch, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# [no] graceful-restart-helper
```

To configure the maximum time (in seconds) to keep a restarting peer's stale routes, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# graceful-restart stalepath-time <1-3600>
```

By default, the maximum time the switch will keep a restarting peer' stale routes is 360 seconds. To reset this value to its default, use the following command:

```
Switch(config-router)# [no] graceful-restart stalepath-time
```
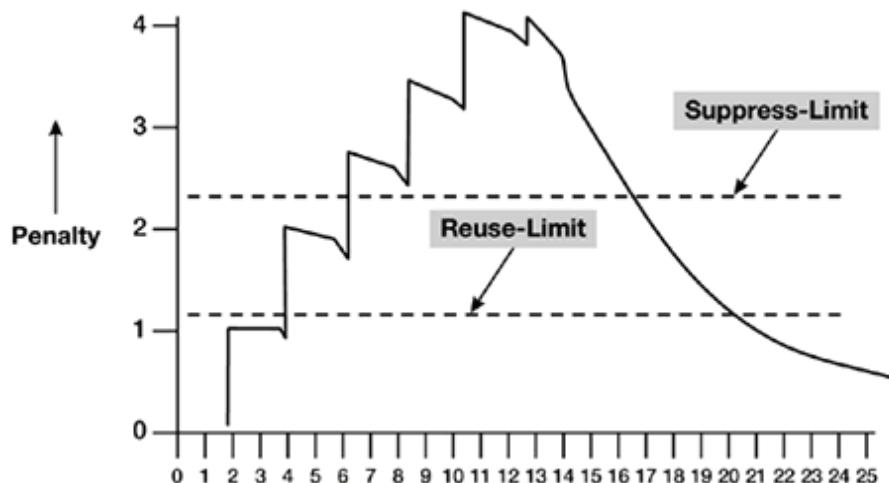
BGP Damping

Route flapping is a symptom of route instability. A route in the routing table may disappear and reappear intermittently when BGP sends a routing table, then withdraws it. A route that appears and disappears intermittently and repeatedly propagates BGP UPDATE and WITHDRAWN messages onto the Internet. If the prefix is constantly flapping, it can result in wasted CPU, bandwidth, and other network resources.

RFC 2439, *BGP Route Flap Damping*, defines a mechanism by which the amount of routing state propagated via BGP can be reduced by reducing the scope of route flap propagation. BGP Route Flap Dampening categorizes routes as either well-behaved or ill-behaved. A well-behaved route shows a high degree of stability during an extended period of time (as determined by configurable limits). An ill-behaved route is no longer advertised; it is suppressed until there is an indication that the route has become stable.

Route flap dampening requires BGP speakers to maintain a history of the advertisement and withdraw of each BGP path implementations provide the capability to tune the suppress-limit, reuse-limit, and half-life values to control how long a misbehaving prefix is suppressed. The general operation of BGP dampening is depicted in Figure 16.

Figure 16. BGP Route Flap Dampening

- A fixed penalty is assessed for each flap. A flap can be either a route withdraw or update with an attribute change.

- If the penalty exceeds the suppress-limit, the route is not advertised.

- The penalty is exponentially decayed based on the half-life value.

- Once the penalty is decayed below the reuse limit, the route is advertised.

The BGP flap dampening feature affects only external BGP routes (those in different ASes). You can also apply it within a confederation, between confederation member ASes. Because routing consistency within an AS is important, do not apply flap damping to iBGP routes, for it will be ignored.)

A value for reachability half time and a different value for unreachability half time is supported. The unreachability half time value must be higher. In this case, when computing the maximum suppress penalty, BGP uses this higher unreachabilty value.

By default, BGP route flap dampening is disabled. To enable or disable it, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# address-family {ipv4|ipv6} unicast
Switch(config-router-af)# [no] dampening [<half-life> [<reuse> <suppress> <maximum
duration> [<unreachability half-life>]]|routemap <route map name>]
```

Note: If the **dampening** command is used without any additional parameters, the switch will assume the default values for those parameters. For more details about this command, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

Soft Reconfiguration Inbound

This feature triggers routing updates for the specified peer without resetting the session. You can use this option if you change an inbound route policy. Soft reconfiguration inbound saves a copy of all routes received from the peer before processing the routes through the inbound route policy. If you change the inbound route policy, Cloud NOS passes these stored routes through the modified inbound route policy to update the route table without tearing down existing peering sessions. Soft reconfiguration inbound can use significant memory resources to store the unfiltered BGP routes.

This feature is disabled by default. To enable or disable it for a BGP peer, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# address-family {ipv4|ipv6} unicast
Switch(config-router-neighbor-af)# [no] soft-configuration inbound
```

BGP Route Refresh

BGP route refresh updates the inbound routing tables dynamically by sending route refresh requests to supporting peers when you change an inbound route policy. The remote BGP peer responds with a new copy of its routes that the local BGP speaker processes with the modified route policy.

The device supports both old and new route refresh capability. BGP peers advertise the route refresh capability as part of the BGP capability negotiation when establishing the BGP peer session.

To initiate a BGP route refresh, use the following command:

```
Switch# remove ip bgp <neighbor address> soft
```

BGP Multiple Address Families

You can manually configure the address families that BGP supports:

- Use the **address-family** command in BGP Configuration Mode to configure features for an address family.

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# address-family {ipv4|ipv6} unicast
Switch(config-router-af)#
```

- Use the **address-family** command in BGP Neighbor Configuration Mode to configure the specific address family for the neighbor.

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# address-family {ipv4|ipv6} unicast
Switch(config-router-neighbor-af)#
```

You must configure address families if you are using route redistribution, address aggregation and other advanced features.

BGP and BFD

Bidirectional forwarding detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for media types, encapsulations, topologies, and routing protocols. You can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different protocol hello mechanisms. BFD makes network profiling and planning easier and reconvergence time consistent and predictable.

BFD provides subsecond failure detection between two BGP peers.

By default, BFD is disabled for BGP peers. To enable or disable it for a BGP peer, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# neighbor <neighbor address> remote-as <AS number>
Switch(config-router-neighbor)# [no] bfd
```

If the BGP peer is multiple hops away from the switch, use the following command to enable or disable BFD for that peer:

```
Switch(config-router-neighbor)# [no] bfd multihop
```

BFD is supported with BGP dynamic peers.

BGP Next Hop Tracking

Next hop tracking (NHT) is used to notify the BGP process asynchronously whenever there is any change in the IGP routes. NHT reduces the convergence time of BGP routes when IGP routes are changed.

The BGP process can wait for a configured or a default time after receiving a next hop change trigger from NSM, before it updates the BGP RIB. You can set and unset the time delay.

This feature is enabled by default. To configure BGP next hop tracking, use the following command:

```
Switch(config)# routing-protocol bgp <AS number (1-4294967295)>
Switch(config-router)# address-family {ipv4|ipv6} unicast
Switch(config-router-af)# nexthop trigger-delay critical <delay value
(1-4294967295 milliseconds)> non-critical <delay value (1-4294967295 milliseconds)>
```

To disable BGP next hop tracking, use the following command:

```
Switch(config-router-af)# nexthop trigger-delay
```

BGP Tuning

The following commands enable you to tune BGP protocols in Routing-protocol BGP Neighbor Address Family Configuration Mode:

- **allowas-in**
- **as-origination-interval**
- **attribute-unchanged**
- **connection-retry-time**
- **description**
- **disallow-infinite-holdtime**
- **enforce-first-as**
- **fall-over**
- **fast-external-fallover**
- **log-neighbor-changes**
- **synchronization**
- **timer**
- **transport connection-mode passive**
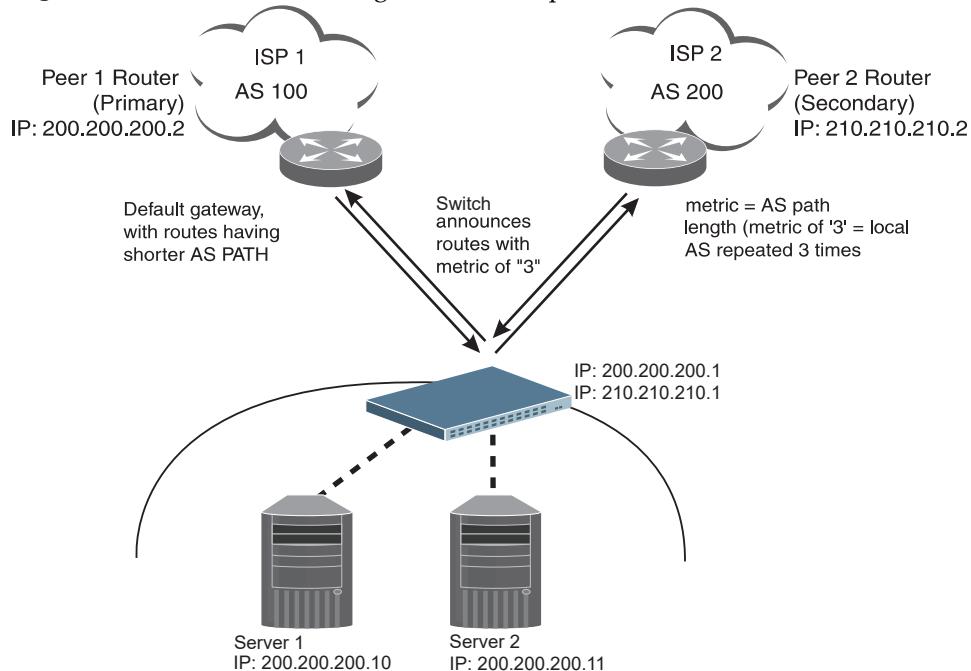
For more information about these commands, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

BGP Failover Configuration

Use the following example to create redundant default gateways for a switch at a Web Host/ISP site, eliminating the possibility, if one gateway goes down, that requests will be forwarded to an upstream router unknown to the switch.

As shown in Figure 17, the switch is connected to ISP 1 and ISP 2. The customer negotiates with both ISPs to allow the switch to use their peer routers as default gateways. The ISP peer routers will then need to announce themselves as default gateways to the switch.

Figure 17. BGP Failover Configuration Example

On the switch, one peer router (the secondary one) is configured with a longer AS path than the other, so that the peer with the shorter AS path will be seen by the switch as the primary default gateway. ISP 2, the secondary peer, is configured with a metric of "3," thereby appearing to the switch to be three router *hops* away.

1. Define the VLANs and IP interfaces using IPv4 addresses.

   The switch will need an IP interface for each default gateway to which it will be connected. Each interface must be placed in the appropriate VLAN. These interfaces will be used as the primary and secondary default gateways for the switch.

   ```
   Switch(config)# interface vlan 1
   Switch(config-if)# ip address 200.200.200.1/24
   Switch(config-if)# exit

   Switch(config)# interface vlan 2
   Switch(config-if)# ip address 200.200.200.1/24
   Switch(config-if)# exit
   ```

2. (Optional) Enable IP forwarding. IP forwarding is enabled on switch by default and is used for VLAN-to-VLAN (non-BGP) routing. Make sure IP forwarding is enabled if the default gateways are on different subnets, or if the switch is connected to different subnets that need to communicate through the switch.

```
Switch(config)# ip forwarding
```

3. Configure BGP peer router 1 and 2 with IPv4 addresses.

```
Switch(config)# routing-protocol bgp <local AS number>
Switch(config-router)# neighbor 200.200.200.2 remote-as 100
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit
Switch(config-router)# exit

Switch(config)# routing-protocol bgp <local AS number>
Switch(config-router)# neighbor 210.210.210.2 remote-as 200
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit
Switch(config-router)# exit
```

Note: Only one AS can be configured on the switch. When configuring BGP, choose the local AS number.

4. Verify the configuration:
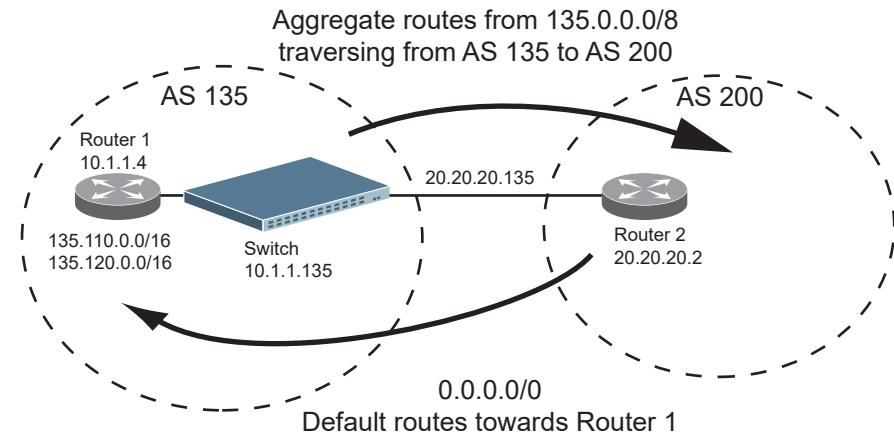
```
Switch# display ip bgp neighbors
```

Default Redistribution and Route Aggregation Example

This example shows you how to configure the switch to redistribute information from one routing protocol to another and create an aggregate route entry in the BGP routing table to minimize the size of the routing table.

As illustrated in Figure 18, you have two routers: one in AS 135 and the other in As 200. Configure the switch to redistribute the default routes from AS 200 to AS 135. At the same time, configure for route aggregation to allow you to condense the number of routes traversing from AS 135 to AS 200.

Figure 18. Route Aggregation and Default Route Redistribution

1. Configure the IP interfaces.

2. Configure the AS number (AS 135) and router ID (10.1.1.135) on the switch.

```
Switch(config)# router-id 10.1.1.135
Switch(config)# routing-protocol bgp 135
Switch(config-router)#
```

3. Configure Router 1 with IPv4 addresses.

```
Router1(config)# router bgp 135
Router1(config-router)# router-id 10.1.1.4
```

4. Configure redistribution for Router 1.

```
Router1(config-router)# address-family ipv4 unicast
Router1(config-router-af)# redistribute static
```

5. Configure aggregation policy control on Router 1 by configuring the IPv4 routes that you want aggregated.

```
Router1(config-router-af)# aggregate-address 135.0.0.0/8
Router1(config-router-af)# exit
```

6. Configure Router 2 with IPv4 addresses.

```
Router2(config)# routing-protocol bgp 200
Router2(config-router)# router-id 20.20.20.2
```

7. Configure redistribution for Router 2.

```
Router2(config-router)# address-family ipv4 unicast
Router2(config-router-af)# redistribute static
```
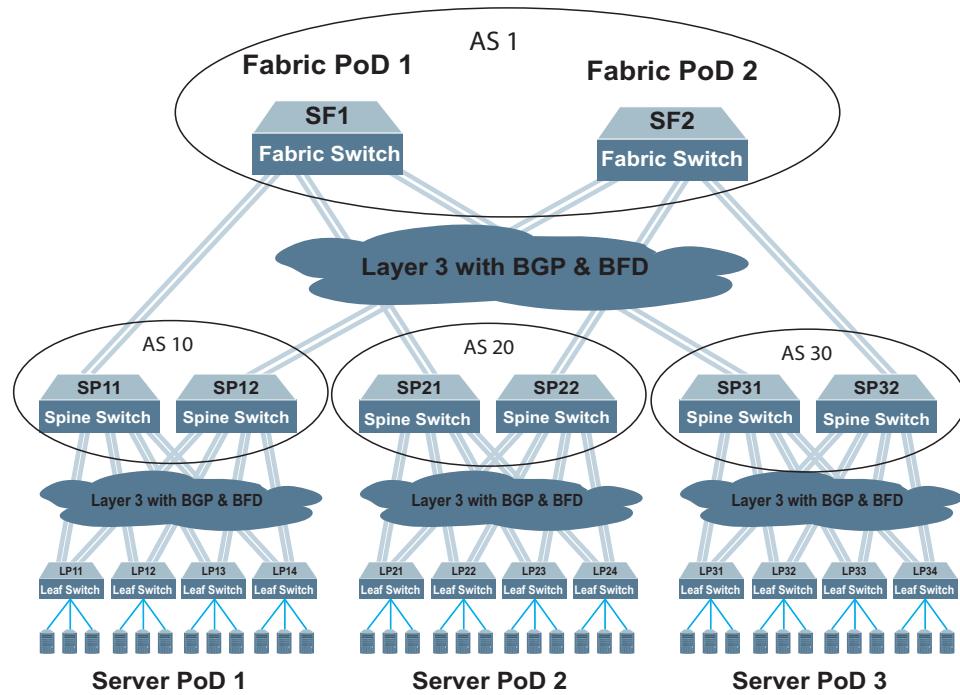
Designing a Clos Network Using BGP

A Clos network is a type of non-blocking, multistage switching architecture that reduces the number of ports required in an interconnected fabric. Instead of a hierarchically oversubscribed system of clusters, a Clos network turns your configuration into a high-performance network.

Clos networks are named after Bell Labs researcher Charles Clos, who determined that throughput is increased in a switching array or fabric if the switches are organized in a leaf-spine hierarchy.

In a leaf-spine topology, a series of leaf switches that form the access layer are fully meshed to a series of spine switches. Figure 19 shows a sample leaf-spine topology with a three-stage multi-7Point of Delivery (PoD) L3 Clos network.

Figure 19. Clos Network Topology

Each fabric switch (SF1 and SF2) serves as a fabric PoD that connects to one of the spine switches in each server PoD using BGP and Bidirectional Forwarding Detection (BFD). The spine switches each connect to four leaf switches using BGP and BFD. Using this kind of configuration, you can add capacity to your network by adding another server PoD instead of reconfiguring your existing infrastructure.

For more information on BFD, see "Bidirectional Forwarding Detection" on page 387.

Clos Network BGP Configuration Example

In this example, from each tier of the network topology, a single switch configuration is covered:

- for the Fabric tier: fabric switch SF1

- for the Spine tier: spine switch SP11

- for the Leaf tier: leaf switch LP11

The configurations for the rest of the switches in each tier are similar to the those presented. Only the IP addresses of the switch interfaces, the neighbor addresses and AS numbers of the BGP peers are different.

Autonomous system (AS) membership is as follows:

- AS 1 includes:
  - SF1
  - SF2

- AS 10 includes:
  - SP11
  - SP12

- AS 20 includes:
  - SP21
  - SP22

- AS 30 includes:
  - SP31
  - SP32

- AS 110 includes LP11

- AS 120 includes LP12

- AS 130 includes LP13

- AS 140 includes LP14

- AS 210 includes LP21

- AS 220 includes LP22

- AS 230 includes LP23

- AS 240 includes LP24

- AS 310 includes LP31

- AS 320 includes LP32

- AS 330 includes LP33

- AS 340 includes LP34

To configure BGP for the Clos Network topology shown in , follow the steps in the next section.

Configure Fabric Switch SF1

1. Configure the switch ethernet interfaces that are connected to its BGP peers (SP11, SP21 and SP31) as routing ports and assign them IP addresses.

```
Switch(config)# interface ethernet 1/49          (connected to SP11)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 1.11.1.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/50          (connected to SP11)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 1.11.2.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/51          (connected to SP21)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 1.21.1.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/52          (connected to SP21)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 1.21.2.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/53          (connected to SP31)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 1.31.1.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/54          (connected to SP31)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 1.31.2.1/30
Switch(config-if)# exit
```

2. Configure a loopback interface on the switch as a routing port and define its control plane service policy:

```
Switch(config)# interface loopback 0
Switch(config-if)# no bridge-port
Switch(config-if)# service-policy input copp-system-policy
Switch(config-if)# exit
```

3. Configure the BGP AS number for the switch, the path selection method and the maximum eBGP paths:

```
Switch(config)# routing-protocol bgp 1
Switch(config-router)# bestpath as-path multipath-relax
Switch(config-router)# address-family ipv4 unicast
Switch(config-router-af)# maximum-paths ebgp 32
Switch(config-router-af)# exit
```

4. Define its BGP peers:

```
Switch(config-router)# neighbor 1.11.1.2 remote-as 10    (BGP peer SP11)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 1.11.2.2 remote-as 10    (BGP peer SP11)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 1.21.1.2 remote-as 20    (BGP peer SP21)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 1.21.2.2 remote-as 20    (BGP peer SP21)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 1.31.1.2 remote-as 30    (BGP peer SP31)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 1.31.2.2 remote-as 30    (BGP peer SP31)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit
```

Configure Spine Switch SP11

1. Configure the switch ethernet interfaces that are connected to its BGP peers (SF1, LP11, LP12, LP13 and LP14) as routing ports and assign them IP addresses.

```
Switch(config)# interface ethernet 1/49        (connected to SF1)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 1.11.1.2/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/50        (connected to SF1)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 1.11.2.2/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/1         (connected to LP11)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.11.1.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/2         (connected to LP11)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.11.2.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/3         (connected to LP12)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.12.1.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/4         (connected to LP12)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.12.2.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/5         (connected to LP13)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.13.1.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/6         (connected to LP13)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.13.2.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/7         (connected to LP14)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.14.1.1/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/8         (connected to LP14)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.14.2.1/30
Switch(config-if)# exit
```

2. Configure a loopback interface on the switch as a routing port and define its control plane service policy:

```
Switch(config)# interface loopback 0
Switch(config-if)# no bridge-port
Switch(config-if)# service-policy input copp-system-policy
Switch(config-if)# exit
```

3. Configure the BGP AS number for the switch, the path selection method and the maximum eBGP paths:

```
Switch(config)# routing-protocol bgp 10
Switch(config-router)# bestpath as-path multipath-relax
Switch(config-router)# address-family ipv4 unicast
Switch(config-router-af)# maximum-paths ebgp 32
Switch(config-router-af)# exit
```

4. Define its BGP peers:

```
Switch(config-router)# neighbor 1.11.1.1 remote-as 1      (BGP peer SF1)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 1.11.2.1 remote-as 1      (BGP peer SF1)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 2.11.1.2 remote-as 110   (BGP peer LP11)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 2.11.2.2 remote-as 110   (BGP peer LP11)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 2.12.1.2 remote-as 120   (BGP peer LP12)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 2.12.2.2 remote-as 120   (BGP peer LP12)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit
```

```
Switch(config-router)# neighbor 2.13.1.2 remote-as 130     (BGP peer LP13)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 2.13.2.2 remote-as 130     (BGP peer LP13)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 2.14.1.2 remote-as 140     (BGP peer LP14)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 2.14.2.2 remote-as 140     (BGP peer LP14)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit
```

Configure Leaf Switch LP11

1. Configure the switch ethernet interfaces that are connected to its BGP peers (SP11 and SP12) as routing ports and assign them IP addresses.

```
Switch(config)# interface ethernet 1/1            (connected to SP11)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.11.1.2/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/2            (connected to SP11)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.11.2.2/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/3            (connected to SP12)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.11.3.2/30
Switch(config-if)# exit

Switch(config)# interface ethernet 1/4            (connected to SP12)
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 2.11.4.2/30
Switch(config-if)# exit
```

2. Configure a loopback interface on the switch as a routing port and define its control plane service policy:

```
Switch(config)# interface loopback 0
Switch(config-if)# no bridge-port
Switch(config-if)# service-policy input copp-system-policy
Switch(config-if)# exit
```

3. Configure the BGP AS number for the switch, the path selection method and the maximum eBGP paths:

```
Switch(config)# routing-protocol bgp 110
Switch(config-router)# bestpath as-path multipath-relax
Switch(config-router)# address-family ipv4 unicast
Switch(config-router-af)# maximum-paths ebgp 32
Switch(config-router-af)# exit
```

4. Define its BGP peers:

```
Switch(config-router)# neighbor 2.11.1.1 remote-as 10      (BGP peer SP11)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 2.11.2.1 remote-as 10      (BGP peer SP11)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 2.11.3.1 remote-as 20      (BGP peer SP12)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit

Switch(config-router)# neighbor 2.11.4.1 remote-as 20      (BGP peer SP12)
Switch(config-router-neighbor)# advertisement-interval 0
Switch(config-router-neighbor)# bfd
Switch(config-router-neighbor)# address-family ipv4 unicast
Switch(config-router-neighbor-af)# exit
Switch(config-router-neighbor)# exit
```
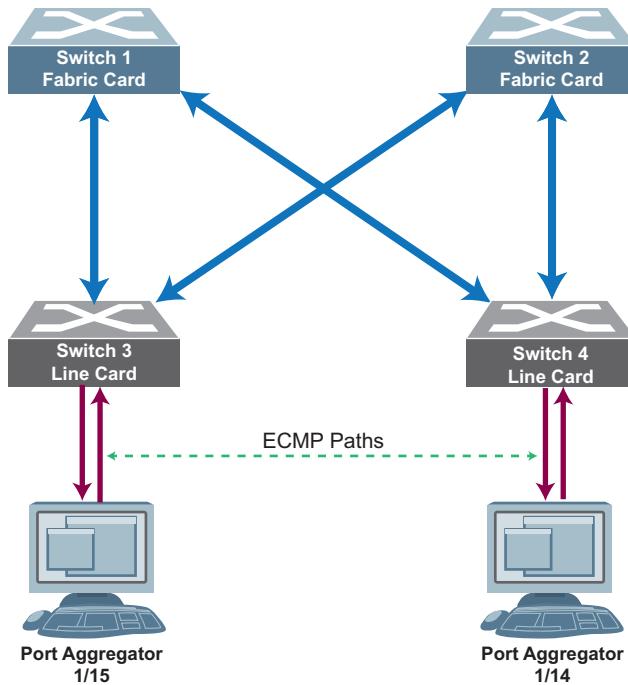
Configuring BGP Unnumbered

The BGP unnumbered feature is useful for quick setup of large configurations for CLOS based network design. In a multi-chassis system you have a set of lower layer, or *line card* switches that all interconnect through a different set of upper layer, or *fabric card* switches to the *fabric chassis*. Consider the following topology:

Figure 20. BGP Unnumbered Topology

Switch 1 and Switch 2 are fabric card switches that connect to the fabric chassis. The fabric card switches interconnect with the lower layer, or line card switches, which in this example are Switch 3 and Switch 4. The line card switches connect to ethernet or portchannel interfaces.

This topology is applicable on both top-of-rack platforms and Clos chassis routers. This feature lets you configure routing between the line card switches and fabric card switches using the BGP protocol with minimal work. Once you enable BGP unnumbered, the line card switches can automatically generate BGP peering sessions with the fabric card switches.

Once the BGP neighbors are established, the routes are propagated, and traffic is routed via ECMP between servers.

Logical Link Discovery Protocol (LLDP) advertises the local MAC, BGP AS number, and BGP interface name on the links that have BGP unnumbered enabled. Once the peer side receives this information via LLDP, it indicates this to BGP. BGP computes the peer IPv6 link-local address from the MAC and interface name received from LLDP and starts to configure the link-local peer with the remote AS received from the LLDP message.

The BGP IPv6 link-local peer neighbors will be configured with addresses in the IPv4 family to transport IPv4 routes with IPv6 link-local next hops. BGP unnumbered supports Extension of AFI/SAFI Definitions for the IPv4 address Family to include IPv6 next hop address using multiprotocol reachable Network Layer Reachability Information (NLRI). This capability is enabled when feature is set and will be negotiated when receiving open messages from neighbors.

Configure BGP Unnumbered

BGP unnumbered on CNOS works with BGPD and LLDP, and only works with routed ports. To configure BGP unnumbered, you must:

● Enable BGP routing protocol.

● Configure the AS number and router ID.

● Enable BGP unnumbered routing.

● Enable BGP unnumbered routing for the desired routed port interface.

● Enable LLDP by default on each link.

● Support BGP.

● Configure BGP unnumbered only on routed port interfaces.

● Have default IPv6 capability on interfaces

Each routed port has a default IPv6 autoconfigured link-local address that is derived from the MAC address + the host's Extended Unique Identifier 64-Bit IPv6 interface identifier (EUI-64) to ensure the uniqueness of the address.

For example, to configure BGP unnumbered on Ethernet port 1/1 with AS number 42, router ID 10.11.12.13, and up to 32 EBGP paths:

1. Set up BGP on the router:

```
Switch(config)# routing-protocol bgp
Switch(config-router)# router-id 10.11.12.13
Switch(config-router)# bestpath as-path multipath-relax
Switch(config-router)# address-family ipv4 unicast
Switch(config-router-af)# exit
Switch(config-router)# address-family ipv6 unicast
Switch(config-router-af)# exit
Switch(config-router)# exit
```

2. Configure the port as unnumbered:

```
Switch(config)# interface ethernet 1/1
Switch(config-if)# no bridge-port
Switch(config-if)# bgp unnumbered
```

BGP Unnumbered Limitations

The following limitations apply to the BGP unnumbered feature:

- This feature only works on routed ports

- The IPv4 routes with link-local nexthop will be installed only in hardware and not in the kernel so traffic is routed over hardware only.

- Programs that have packets that need to be routed by the CPU, such as ping or traceroute, will not work with packets with a destination associated with a hardware-only route.

Note: There is no hardware limitation related to this feature.

Differentiated Services and BGP

BGP works with the differentiated services (DS) computer networking architecture. You can use differentiated services with BGP to provide low-latency to critical network traffic, such as VOIP, while providing best-effort service to non-critical services.

A Differentiated Services Code Point (DSCP) is a method used to classify the way IP packets are queued before being forwarded by a router. Per RFCs 2474 and 2475, DS uses the 6-bit DSCP in the 8-bit DS field in the IP header to classify packets. The DSCP field and ECN (Explicit Congestion Notification) fields replace the IPv4 Type Of Service field in the IPv4 packet header.

When a router receives an IP header with a DSCP, it assigns one of 64 possible forwarding behaviors known as Per Hop Behaviors (PHBs). A PHB provides a particular service level, such as queuing or bandwidth, in accordance with network policy.

Table 41 contains commonly used DSCP values.

Table 41. *Commonly Used DSCP Values*

DSCP Value	Hex Value	Decimal value	Meaning	Drop probability	Equivalent IP precedence value
101 110	0x2e	46	Expedited forwarding (EF)	N/A	101 Critical
000 000	0x00	0	Best effort	N/A	000 - Routine
001 010	0x0a	10	AF11	Low	001 - Priority
001 100	0x0c	12	AF12	Medium	001 - Priority
001 110	0x0e	14	AF13	High	001 - Priority
010 010	0x12	18	AF21	Low	010 - Immediate
010 100	0x14	20	AF22	Medium	010 - Immediate
010 110	0x16	22	AF23	High	010 - Immediate
011 010	0x1a	26	AF31	Low	011 - Flash
011 100	0x1c	28	AF32	Medium	011 - Flash
011 110	0x1e	30	AF33	High	011 - Flash
100 010	0x22	34	AF41	Low	100 - Flash override
100 100	0x24	36	AF42	Medium	100 - Flash override
100 110	0x26	38	AF43	High	100 - Flash override

Commands for Using DS with BGP

To set a DSCP value with BGP, enter:

```
Switch(config-router)# dscp <value>
```

where *value* is the DSCP value; an integer from 0-63.

To display the DSCP value, enter:

```
Switch# display running-config bgp
```

Note: Configuring a DSCP value with BGP will reset BGP adjacencies.

DS with BGP Example

The following example sets BGP AS 100 to use DSCP 45 and then shows the changes on the switch:

```
Switch(config)# routing-protocol bgp 100
Switch(config-router)# dscp 45
Switch(config-router)# exit
Switch(config)# display running-config bgp
!
routing-protocol bgp 100
 router-id 10.30.30.25
 bestpath as-path multipath-relax
 dscp 45
 log-neighbor-changes
 address-family ipv4 unicast
...
!
```

Chapter 23. Open Shortest Path First

Lenovo Cloud Network Operating System supports the Open Shortest Path First (OSPF) routing protocol. The Cloud NOS implementation conforms to the OSPF version 2 specifications detailed in Internet RFC 2328. The following sections discuss OSPF support for the switch:

- "OSPFv2 Overview" on page 510. This section provides information on OSPFv2 concepts, such as types of OSPF areas, types of routing devices, neighbors, adjacencies, link state database, authentication, and internal versus external routing.

- "OSPFv2 Implementation in Cloud NOS" on page 514. This section describes how OSPFv2 is implemented in CNOS, such as configuration parameters, electing the designated router, summarizing routes, defining route maps and so forth.

- "OSPFv2 Configuration Examples" on page 524. This section provides step-by-step instructions on configuring different OSPFv2 examples:

  - Creating a simple OSPF domain
  - Creating virtual links
  - Summarizing routes

OSPFv2 Overview

OSPF is designed for routing traffic within a single IP domain called an Autonomous System (AS). The AS can be divided into smaller logical units known as *areas*.

All routing devices maintain link information in their own Link State Database (LSDB). OSPF allows networks to be grouped together into an area. The topology of an area is hidden from the rest of the AS, thereby reducing routing traffic. Routing within an area is determined only by the area's own topology, thus protecting it from bad routing data. An area can be generalized as an IP subnetwork.

The following sections describe key OSPF concepts.
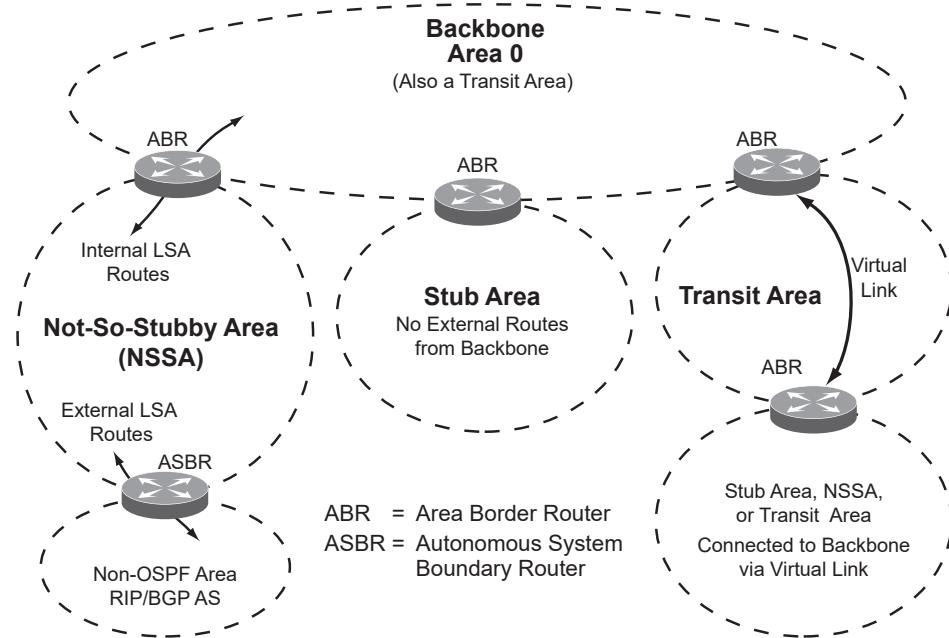
Types of OSPF Areas

An AS can be broken into logical units known as *areas*. In any AS with multiple areas, one area must be designated as area 0, known as the *backbone*. The backbone acts as the central OSPF area. All other areas in the AS must be connected to the backbone. Areas inject summary routing information into the backbone, which then distributes it to other areas as needed.

As shown in Figure 21, OSPF defines the following types of areas:

- Stub Area—an area that is connected to only one other area. External route information is not distributed into stub areas.

- Not-So-Stubby-Area (NSSA)—similar to a stub area with additional capabilities. Routes originating from within the NSSA can be propagated to adjacent transit and backbone areas. External routes from outside the AS can be advertised within the NSSA but can be configured to not be distributed into other areas.

- Transit Area—an area that carries data traffic which neither originates nor terminates in the area itself.

Figure 21. OSPF Area Types
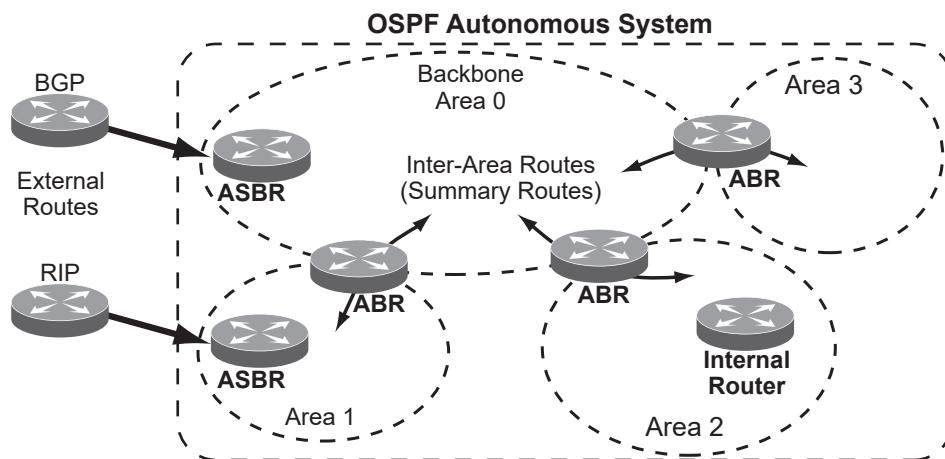
Types of OSPF Routing Devices

As shown in Figure 22, OSPF uses the following types of routing devices:

- Internal Router (IR)—a router that has all of its interfaces within the same area. IRs maintain LSDBs identical to those of other routing devices within the local area.

- Area Border Router (ABR)—a router that has interfaces in multiple areas. ABRs maintain one LSDB for each connected area and disseminate routing information between areas.

- Autonomous System Boundary Router (ASBR)—a router that acts as a gateway between the OSPF domain and non-OSPF domains, such as RIP, BGP, and static routes.

Figure 22. OSPF Domain and an Autonomous System

Neighbors and Adjacencies

In areas with two or more routing devices, *neighbors* and *adjacencies* are formed.

Neighbors are routing devices that maintain information about each others' state. To establish neighbor relationships, routing devices periodically send hello packets on each of their interfaces. All routing devices that share a common network segment, appear in the same area, and have the same health parameters (`hello` and `dead` intervals) and authentication parameters respond to each other's hello packets and become neighbors. Neighbors continue to send periodic hello packets to advertise their health to neighbors. In turn, they listen to hello packets to determine the health of their neighbors and to establish contact with new neighbors.

The hello process is used for electing one of the neighbors as the network segment's Designated Router (DR) and one as the network segment's Backup Designated Router (BDR). The DR is adjacent to all other neighbors on that specific network segment and acts as the central contact for database exchanges. Each neighbor sends its database information to the DR, which relays the information to the other neighbors.

The BDR is adjacent to all other neighbors (including the DR). Each neighbor sends its database information to the BDR just as with the DR, but the BDR merely stores this data and does not distribute it. If the DR fails, the BDR will take over the task of distributing database information to the other neighbors.

The Link-State Database

OSPF is a link-state routing protocol. A *link* represents an interface or routable path from the routing device. By establishing an adjacency with the DR, each routing device in an OSPF area maintains an identical Link-State Database (LSDB) describing the network topology for its area.

Each routing device transmits a Link-State Advertisement (LSA) on each of its *active* interfaces. LSAs are entered into the LSDB of each routing device. OSPF uses *flooding* to distribute LSAs between routing devices. Interfaces may also be *passive*. Passive interfaces send LSAs to active interfaces, but do not receive LSAs, hello packets, or any other OSPF protocol information from active interfaces. Passive interfaces behave as stub networks, allowing OSPF routing devices to be aware of devices that do otherwise participate in OSPF (either because they do not support it, or because the administrator chooses to restrict OSPF traffic exchange or transit).

When LSAs result in changes to the routing device's LSDB, the routing device forwards the changes to the adjacent neighbors (the DR and BDR) for distribution to the other neighbors.

OSPF routing updates occur only when changes occur, rather than periodically. For each new route, if a neighbor is interested in that route (for example, if configured to receive static routes and the new route is static), an update message containing the new route is sent to the adjacency. For each route removed from the route table, if the route has already been sent to a neighbor, an update message containing the route to withdraw is sent.

The Shortest Path First Tree

The routing devices use a link-state algorithm (Dijkstra's algorithm) to calculate the shortest path to all known destinations, based on the cumulative *cost* required to reach the destination.

The cost of an individual interface in OSPF is an indication of the overhead required to send packets across it.

Internal Versus External Routing

To ensure effective processing of network traffic, every routing device on your network needs to know how to send a packet (directly or indirectly) to any other location or destination in your network. This is referred to as *internal routing* and can be done with static routes or using active internal routing protocols, such as OSPF, RIP, or RIPv2.

It is also useful to tell routers outside your network (upstream providers or *peers*) about the routes you have access to in your network. Sharing of routing information between autonomous systems is known as *external routing*.

Typically, an AS will have one or more border routers (peer routers that exchange routes with other OSPF networks) as well as an internal routing system enabling every router in that AS to reach every other router and destination within that AS.

When a routing device *advertises* routes to boundary routers on other autonomous systems, it is effectively committing to carry data to the IP space represented in the route being advertised. For example, if the routing device advertises 192.204.4.0/24, it is declaring that if another router sends data destined for any address in the 192.204.4.0/24 range, it will carry that data to its destination.

OSPFv2 Implementation in Cloud NOS

CNOS supports a single instance of OSPF and up to 8k routes on the network. The following sections describe OSPF implementation in CNOS:

Configurable Parameters

In CNOS, OSPF parameters can be configured through the Industry Standard Command Line Interfaces (ISCLI), or through SNMP. For more information, see Chapter 2, "Switch Administration."

The ISCLI supports the following parameters: interface output cost, interface priority, dead and hello intervals, retransmission interval, and interface transmit delay.

In addition to the preceding parameters, you can specify the following:

- OSPF traps—Traps produce messages upon certain events or error conditions, such as missing a hello, failing a neighbor, or recalculating the SPF.

- Link-State Database size—The size of the LSA database can be specified to help manage the memory resources on the switch.

- Stub area metric—A stub area can be configured to send a numeric metric value such that all routes received via that stub area carry the configured metric to potentially influence routing decisions.

- Default routes—Default routes with weight metrics can be manually injected into transit areas. This helps establish a preferred route when multiple routing devices exist between two areas. It also helps route traffic to external networks.

- Passive—When enabled, the interface sends LSAs to upstream devices, but does not otherwise participate in OSPF protocol exchanges.

- Point-to-Point—For LANs that have only two OSPF routing agents (the switch and one other device), this option allows the switch to significantly reduce the amount of routing information it must carry and manage.

Defining Areas

If you are configuring multiple areas in your OSPF domain, one of the areas must be designated as area 0, known as the *backbone*. The backbone is the central OSPF area and is usually physically connected to all other areas. The areas inject routing information into the backbone which, in turn, disseminates the information into other areas.

Since the backbone connects the areas in your network, it must be a contiguous area. If the backbone is partitioned (possibly as a result of joining separate OSPF networks), parts of the AS will be unreachable, and you will need to configure *virtual links* to reconnect the partitioned areas (see "Virtual Links" on page 519).

Up to 20 OSPF areas can be connected to the switch with CNOS software. To configure an area, the OSPF number must be defined and then attached to a network interface on the switch. The full process is explained in the following sections.

An OSPF area is defined by assigning an *area ID*. The command to define and enable an OSPF area is as follows:

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area {<area ID (1-4294967295)>|<area ID as IPv4 address>}
```

Using the Area ID to Assign the OSPF Area Number

The OSPF area number can be defined as a decimal number ranging from 1 to 4294967295, or as an IPv4 address. The IPv4 address format is used for compatibility with two different systems of notation used by other OSPF network vendors. There are two valid ways to designate an area ID:

● Single Number

  Most common OSPF vendors express the area ID number as a single number. For example, the Cisco IOS-based router command "network 1.1.2.0 0.0.0.255 area 1" defines the area number simply as "area 1."

● Multi-octet (*IP address*): Placing the area number in the last octet (0.0.0.*n*)

  Some OSPF vendors express the area ID number in multi-octet format. For example, "area 0.0.0.2" represents OSPF area 2 and can be specified directly on the switch as "area 0.0.0.2".

On the switch, using the last octet in the area ID, "area 1" is equivalent to "area 0.0.0.1".

Note: Although both types of area ID formats are supported, be sure that the area IDs are in the same format throughout an area.

Attaching an Area to a Network

Once an OSPF area has been defined, it must be associated with a network. To attach the area to a network, you must assign the OSPF area ID to an IP interface that participates in the area. The format for the command is as follows:

```
Switch(config)# interface ethernet <chassis number>/<port number>
Switch(config-if)# no bridge-port
Switch(config-if)# ip address <IP address/mask>
Switch(config-if)# ip router ospf 0 area <area ID> [secondaries none]
Switch(config-if)# exit
```

For example, the following commands could be used to add a network area 0.0.0.10 to the OSPFv2 instance:

```
Switch(config)# interface ethernet 1/2
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 192.0.2.1/16
Switch(config-if)# ip router ospf 0 area 0.0.0.10
Switch(config-if)# exit
```

Note: OSPFv2 supports IPv4 only.

Interface Cost

The OSPF link-state algorithm (Dijkstra's algorithm) places each routing device at the root of a tree and determines the cumulative *cost* required to reach each destination. You can manually enter the cost for the output route with the following command (Interface IP mode):

```
Switch(config)# interface ethernet <chassis number>/<port number>
Switch(config-if)# ip ospf cost <cost value (1-65535)>
```

Electing the Designated Router and Backup

In any broadcast type subnet, a Designated Router (DR) is elected as the central contact for database exchanges among neighbors. On subnets with more the one device, a Backup Designated Router (BDR) is elected in case the DR fails.

DR and BDR elections are made through the hello process. The election can be influenced by assigning a priority value to the OSPF interfaces on the switch. The command is as follows:

```
Switch(config)# interface ethernet <chassis number>/<port number>
Switch(config-if)# ip ospf priority <priority value (0-255)>
```

A priority value of 255 is the highest, and 1 is the lowest. A priority value of zero (0) specifies that the interface cannot be used as a DR or BDR. In case of a tie, the routing device with the highest router ID wins. Interfaces configured with a priority of zero (0) do not participate in the DR or BDR election process. Layer 3 switches with OSPF enabled that are connected between each other by passive interfaces will also not participate in the DR or BDR election process.

Summarizing Routes

Route summarization condenses routing information. Without summarization, each routing device in an OSPF network would retain a route to every subnet in the network. With summarization, routing devices can reduce some sets of routes to a single advertisement, reducing both the load on the routing device and the perceived complexity of the network. The importance of route summarization increases with network size.

Summary routes can be defined for up to 16 IP address ranges using the following command:

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area <area ID> range {<area range prefix> <area range prefix mask>|
|<area range prefix>/<prefix length>} [advertise|non-advertise]
```

where:

- *area ID* is written as a number from 0 to 4294967295 or as an IPv4 address

- *area range prefix* is the base IP address for the range

- *area range prefix mask* is the IP address mask for the range

For a detailed configuration example, see "Example 3: Summarizing Routes" on page 528.

A filter can be configured to advertise summary routes on an Area Border Router (ABR). This command suppresses incoming and outgoing summary routes between a specific area and the others:
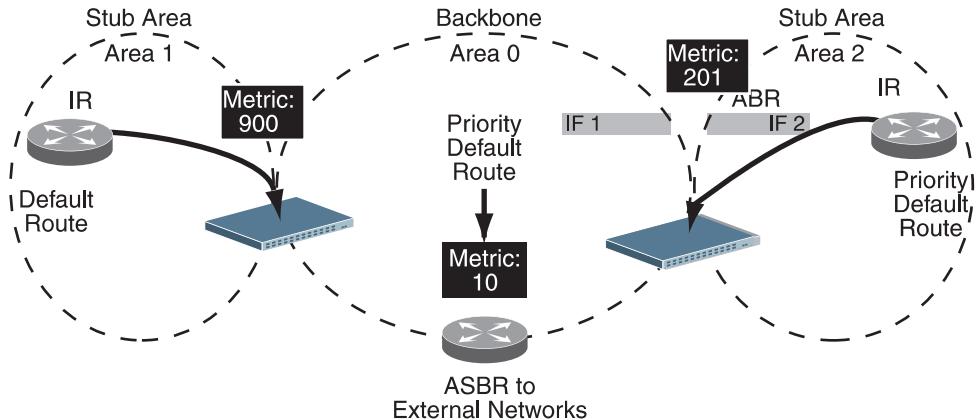
```
Switch(config)# routing-protocol ospf
Switch(config-router)# area <area id> filter-list route-map <map name>
{in|out}
```

Default Routes

When an OSPF routing device encounters traffic for a destination address it does not recognize, it forwards that traffic along the *default route*. Typically, the default route leads upstream toward the backbone until it reaches the intended area or an external router.

Each switch acting as an ABR automatically inserts a default route into stub. In simple OSPF stub areas with only one ABR leading upstream (see Area 1 in Figure 23), any traffic for IP address destinations outside the area is forwarded to the switch's IP interface, and then into the connected transit area (usually the backbone). Since this is automatic, no further configuration is required for such areas.

Figure 23. Injecting Default Routes

If the switch is in a transit area and has a configured default gateway, it can inject a default route into rest of the OSPF domain. Use the following command to configure the switch to inject OSPF default routes:

```
Switch(config)# routing-protocol ospf
Switch(config-router)# default-information originate [always] [metric
<metric value (0-16777214)>] [metric-type {1|2}] [route-map <map-name>]
```

In the above command, the `metric` is used to calculate the cost of the default route. The metric value is the inverse proportional of the bandwidth of the default route. A higher bandwidth means a lower metric value, and a lower bandwidth means a higher metric value. A metric value of one (1) informs OSPF that the route has the highest bandwidth, and a metric value of zero (0) informs OSPF that the route is directly connected to the switch.

OSPF supports two types of external metrics for default routes:

- `type 1` - uses the cost of the external route and the cost to reach the autonomous system boundary router (ASBR).

- `type 2` - uses only the cost of the external route, and ignores the cost to reach the ASBR.

By default, OSPF uses type 2 external metric. If both types of metric are present simultaneously in the AS, type 1 external metrics always have precedence over type 2.

When the switch is configured to inject a default route, an AS-external LSA with link state ID 0.0.0.0 is propagated throughout the OSPF routing domain. This LSA is sent with the configured metric value and metric type.

The OSPF default route configuration can be removed with the command:

```
Switch(config)# routing-protocol ospf
Switch(config-router)# no default-information originate
```

Virtual Links

Usually, all areas in an OSPF AS are physically connected to the backbone. In some cases where this is not possible, you can use a *virtual link*. Virtual links are created to connect one area to the backbone through another non-backbone area (see Figure 21 on page 511).

The area which contains a virtual link must be a transit area and have full routing information. Virtual links cannot be configured inside a stub area or NSSA.

To set a link between two backbone areas that are physically separated through other non backbone area, use the following command:

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area <area ID> virtual link <IP address>
Switch(config-router-vlink)#
```

To configure an authentication type and key between virtual-link neighbors, use the following command:

```
Switch(config-router-vlink)# authentication [message-digest|null]
Switch(config-router-vlink)# authentication [0] <string>
Switch(config-router-vlink)# message-digest-key <key ID> md5 [0] <string>
```

where `message-digest` enables MD5 authentication and *string* is the password string.

For a detailed configuration example on Virtual Links, see "Example 2: Virtual Links" on page 526.

Router ID

Routing devices in OSPF areas are identified by a router ID. The router ID is expressed in IP address format. The IP address of the router ID is not required to be included in any IP interface range or in any OSPF area, and may even use the switch loopback interface.

The router ID can be configured in one of the following two ways:

- Dynamically—OSPF protocol configures the highest IP interface IP address as the router ID (loopback interface has priority over the IP interface). This is the default.
- Statically—Use the following command to manually configure the router ID:

```
Switch(config)# routing-protocol ospf
Switch(config-router)# router-id <IPv4 address>
```

If there is a loopback interface, its IP address is always preferred as the router ID, instead of an IP interface address. The **router-id** command is the preferred method to set the router ID and it is always used in preference to the other methods.

- o To modify the router ID from static to dynamic, delete the Router ID using the **no router-id** command and re-initialize the OSPF process.
- o To view the router ID, use the following command:

```
Switch# display router-id
```

Authentication
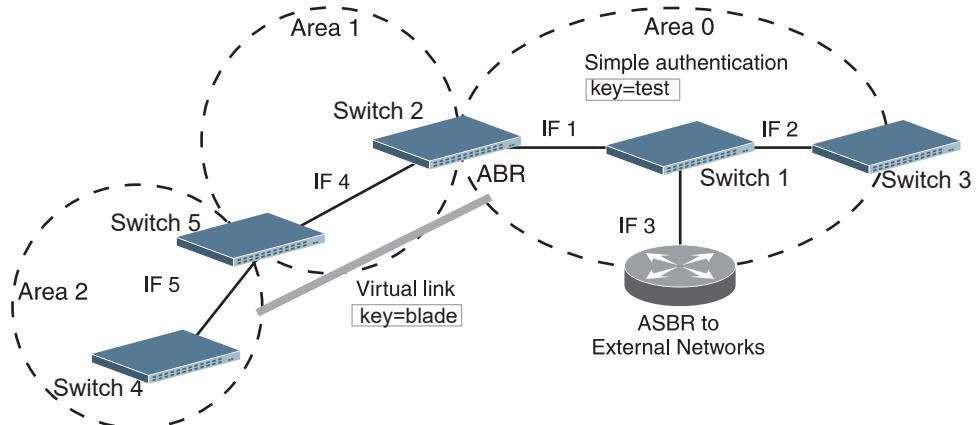
OSPF protocol exchanges can be authenticated so that only trusted routing devices can participate. This ensures less processing on routing devices that are not listening to OSPF packets.

OSPF allows packet authentication and uses IP multicast when sending and receiving packets. Routers participate in routing domains based on pre-defined passwords. CNOS supports simple password (type 1 plain text passwords) and MD5 cryptographic authentication. This type of authentication allows a password to be configured per interface.

We strongly recommend that you implement MD5 cryptographic authentication as a best practice.

Figure 24 shows authentication configured for area 0 with the password test. Simple authentication is also configured for the virtual link between area 2 and area 0. Area 1 is not configured for OSPF authentication.

Figure 24. OSPF Authentication

Configuring Plain Text OSPF Passwords

To configure simple plain text OSPF passwords on the switches shown in Figure 24 use the following commands:

1. Enable OSPF authentication for Area 0 on switches 1, 2, and 3.

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area 0 authentication
Switch(config-router)# exit
```

2. Configure a simple text non-encrypted password for each OSPF interface in Area 0 on switches 1, 2, and 3.

```
Switch(config)# interface vlan 1
Switch(config-if)# ip ospf authentication-key 0 test
Switch(config-if)# exit

Switch(config)# interface vlan 2
Switch(config-if)# ip ospf authentication-key 0 test
Switch(config-if)# exit

Switch(config)# interface vlan 3
Switch(config-if)# ip ospf authentication-key 0 test
Switch(config-if)# exit
```

3. Enable OSPF authentication for Area 2 on switch 4.

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area 2 authentication
```

4. Configure a simple text non-encrypted password for the virtual link between Area 2 and Area 0 on switches 2 and 4.

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area 2 virtual-link 1.2.3.4
Switch(config-router-vlink)# authentication
Switch(config-router-vlink)# authentication-key 0 blade
```

Configuring MD5 Authentication

Use the following commands to configure MD5 authentication on the switches shown in Figure 24:

1. Enable OSPF MD5 authentication for Area 0 on switches 1, 2, and 3.

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area 0 authentication message-digest
Switch(config-router)# exit
```

2. Assign MD5 key ID to OSPF interfaces on switches 1, 2, and 3.

```
Switch(config)# interface vlan 1
Switch(config-if)# ip ospf message-digest-key 11 md5 MyAuthKey
Switch(config-if)# exit

Switch(config)# interface vlan 2
Switch(config-if)# ip ospf message-digest-key 22 md5 MyAuthKey
Switch(config-if)# exit

Switch(config)# interface vlan 3
Switch(config-if)# ip ospf message-digest-key 33 md5 MyAuthKey
Switch(config-if)# exit
```

Note: In the above configuration example, the password MyAuthKey is encrypted. To not encrypt the password, use the following command:

```
Switch(config-if)# ip ospf message-digest-key <key ID> md5 0 MyAuthKey
```

Loopback Interfaces in OSPF

A loopback interface is an IP interface which has an IP address, but is not associated with any particular physical port. The loopback interface is thus always available to the general network, regardless of which specific ports are in operation. Because loopback interfaces are always available on the switch, loopback interfaces may present an advantage when used as the router ID.

If dynamic router ID selection is used (see "Router ID" on page 519) loopback interfaces can be used to force router ID selection. If a loopback interface is configured, its IP address is automatically selected as the router ID, even if other IP interfaces have lower IP addresses. If more than one loopback interface is configured, the lowest loopback interface IP address is selected.

Loopback interfaces can be advertised into the OSPF domain by specifying an OSPF host route with the loopback interface IP address.

To enable OSPF on an existing loopback interface:

```
Switch(config)# interface loopback <0-7>
Switch(config-if)# ip router ospf 0 area <area ID>
Switch(config-if)# exit
```

Graceful Restart Helper

The CNOS design provides a complete separation of its control plane from the forwarding plane, thus allowing the restart or upgrade of control plane software without disturbing forwarding. Such a restart/upgrade is called *graceful restart*.

In graceful restart, a restarting router is helped by neighbors by announcing links in their LSAs. These neighbors are considered to be in *helper mode* for the duration (grace period) of graceful restart.

The feature is enabled by default. To disable it, use the following command:

```
Switch(config)# graceful-restart ospf helper never
```

OSPF and BFD

Bidirectional forwarding detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for media types, encapsulations, topologies, and routing protocols. You can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different protocol hello mechanisms. BFD makes network profiling and planning easier and reconvergence time consistent and predictable.

BFD can work together with OSPF to increase route convergence as an alternative to adjusting the OSPF Hello Interval and Dead Interval. BFD improves the speed of failure detection by having shorter timer limits than the OSPF failure detection mechanisms.

By default, BFD is disabled for OSPF. To enable or disable BFD to work together with OSPF, use the following command:

```
Switch(config)# routing-protocol ospf
Switch(config-router)# [no] bfd
```

OSPFv2 Configuration Examples

A summary of the basic steps for configuring OSPF on the switch is listed here. Detailed instructions for each of the steps is covered in the following sections:

1. Configure IP interfaces.

   One IP interface is required for each desired network (range of IP addresses) being assigned to an OSPF area on the switch.

2. (Optional) Configure the router ID.

3. Enable OSPF on the switch.

4. Define the OSPF areas.

5. Configure OSPF interface parameters.

   IP interfaces are used for attaching networks to the various areas.

6. (Optional) Configure route summarization between OSPF areas.
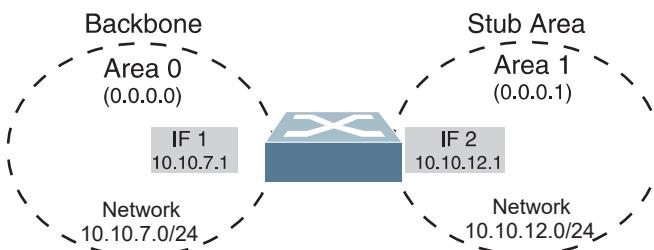
7. (Optional) Configure virtual links.

8. (Optional) Configure host routes.

Example 1: Simple OSPF Domain

In this example, two OSPF areas are defined—one area is the backbone and the other is a stub area. A stub area does not allow advertisements of external routes, thus reducing the size of the database. Instead, a default summary route of IP address 0.0.0.0 is automatically inserted into the stub area. Any traffic for IP address destinations outside the stub area will be forwarded to the stub area's IP interface, and then into the backbone.

Figure 25. A Simple OSPF Domain

Follow this procedure to configure OSPF support as shown in Figure 25:

1. Configure IP interfaces on each network that will be attached to OSPF areas.

   In this example, two IP interfaces are needed:

   - Interface 1 for the backbone network on 10.10.7.0/24
   - Interface 2 for the stub area network on 10.10.12.0/24

   ```
   Switch(config)# interface vlan 1
   Switch(config-if)# ip address 10.10.7.1/24
   Switch(config-if)# exit

   Switch(config)# interface vlan 2
   Switch(config-if)# ip address 10.10.12.1/24
   Switch(config-if)# exit
   ```

   Note: OSPFv2 supports IPv4 only.

2. Enable OSPF.

   ```
   Switch(config)# routing-protocol ospf
   Switch(config-router)#
   ```

3. Define the stub area.

   ```
   Switch(config-router)# area 1 stub
   Switch(config-router)# exit
   ```

4. Attach the network interface to the backbone.

   ```
   Switch(config)# interface vlan 1
   Switch(config-if)# ip router ospf 0 area 0
   Switch(config-if)# exit
   ```
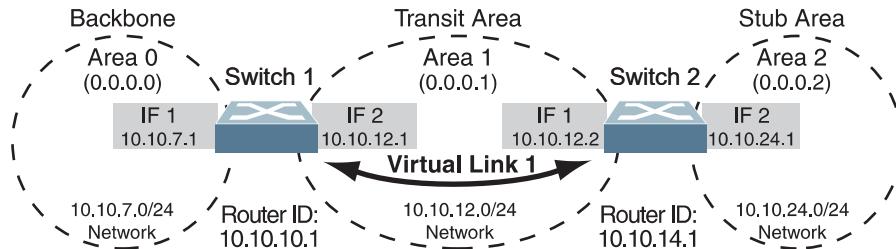
5. Attach the network interface to the stub area.

   ```
   Switch(config)# interface vlan 2
   Switch(config-if)# ip router ospf 0 area 1
   Switch(config-if)# exit
   ```

Example 2: Virtual Links

In the example shown in Figure 26, area 2 is not physically connected to the backbone as is usually required. Instead, area 2 will be connected to the backbone via a virtual link through area 1. The virtual link must be configured at each endpoint.

Figure 26. Configuring a Virtual Link

Note: OSPFv2 supports IPv4 only.

Configuring OSPF for a Virtual Link on Switch 1

1. Configure IP interfaces on each network that will be attached to the switch.

   In this example, two IP interfaces are needed:

   - Interface 1 for the backbone network on 10.10.7.0/24

   - Interface 2 for the transit area network on 10.10.12.0/24

   ```
   Switch(config)# interface vlan 1
   Switch(config-if)# ip address 10.10.7.1/24
   Switch(config-if)# exit

   Switch(config)# interface vlan 2
   Switch(config-if)# ip address 10.10.12.1/24
   Switch(config-if)# exit
   ```

2. Configure the router ID.

   A router ID is required when configuring virtual links. Later, when configuring the other end of the virtual link on Switch 2, the router ID specified here will be used as the target virtual neighbor (nbr) address.

   ```
   Switch(config)# routing-protocol ospf
   Switch(config-router)# router-id 10.10.10.1
   ```

3. Attach the network interface to the backbone.

   ```
   Switch(config)# interface vlan 1
   Switch(config-if)# ip router ospf 0 area 0
   Switch(config-if)# exit
   ```

4. Attach the network interface to the transit area.

```
Switch(config)# interface vlan 2
Switch(config-if)# ip router ospf 0 area 1
Switch(config-if)# exit
```

5. Configure the virtual link.

The nbr router ID configured in this step must be the same as the router ID that will be configured for Switch #2 in .

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area 1 virtual-link 10.10.14.1
```

Configuring OSPF for a Virtual Link on Switch 2

1. Configure IP interfaces on each network that will be attached to OSPF areas.

In this example, two IP interfaces are needed:

- Interface 1 for the transit area network on 10.10.12.0/24
- Interface 2 for the stub area network on 10.10.24.0/24

```
Switch(config)# interface vlan 1
Switch(config-ip-if)# ip address 10.10.12.2/24
Switch(config-ip-if)# exit

Switch(config)# interface vlan 2
Switch(config-ip-if)# ip address 10.10.24.1/24
Switch(config-ip-if)# exit
```

2. Configure the router ID.

A router ID is required when configuring virtual links. This router ID must be the same one specified as the target virtual neighbor (nbr) on switch 1 in .

```
Switch(config-router)# router-id 10.10.14.1
```

3. Enable OSPF.

```
Switch(config)# routing-protocol ospf
```

4. Define the stub area.

```
Switch(config-router)# area 2 stub
Switch(config-router)# exit
```

5. Attach the network interface to the transmit area.

```
Switch(config)# interface vlan 1
Switch(config-ip-if)# ip router ospf 0 area 1
Switch(config-ip-if)# exit
```

6. Attach the network interface to the stub area.

```
Switch(config)# interface vlan 2
Switch(config-ip-if)# ip router ospf 0 area 2
Switch(config-ip-if)# exit
```

7. Configure the virtual link.

The nbr router ID configured in this step must be the same as the router ID that was configured for switch #1 in

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area 1 virtual-link 10.10.10.1
```

Other Virtual Link Options

- You can use redundant paths by configuring multiple virtual links.
- Only the endpoints of the virtual link are configured. The virtual link path may traverse multiple routers in an area as long as there is a routable path between the endpoints.
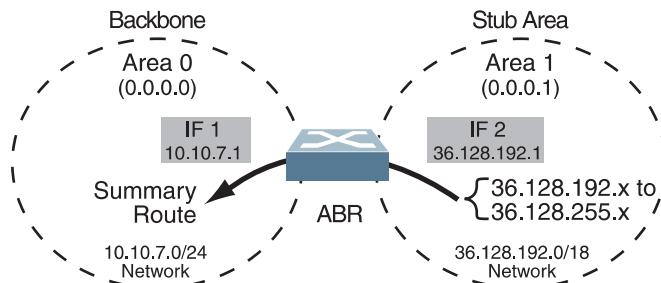
Example 3: Summarizing Routes

By default, ABRs advertise all the network addresses from one area into another area. Route summarization can be used for consolidating advertised addresses and reducing the perceived complexity of the network.

If network IP addresses in an area are assigned to a contiguous subnet range, you can configure the ABR to advertise a single summary route that includes all individual IP addresses within the area.

The following example shows one summary route from area 1 (stub area) injected into area 0 (the backbone). The summary route consists of all IP addresses from 36.128.192.0 through 36.128.255.255 except for the routes in the range 36.128.200.0 through 36.128.200.255.

Figure 27. Summarizing Routes

Notes:

- OSPFv2 supports IPv4 only.
- You can specify a range of addresses to prevent advertising by using the hide option. In this example, routes in the range 36.128.200.0 through 36.128.200.255 are kept private.

Use the following procedure to configure OSPF support as shown in Figure 27:

1. Configure IP interfaces for each network which will be attached to OSPF areas.

```
Switch(config)# interface vlan 1
Switch(config-if)# ip address 10.10.7.1/24
Switch(config-if)# exit

Switch(config)# interface vlan 2
Switch(config-if)# ip address 36.128.192.1/18
Switch(config-if)# exit
```

2. Enable OSPF.

```
Switch(config)# routing-protocol ospf
```

3. Attach the network interface to the backbone.

```
Switch(config)# interface vlan 1
Switch(config-if)# ip router ospf 0 area 0
Switch(config-if)# exit
```

4. Attach the network interface to the stub area.

```
Switch(config)# interface vlan 2
Switch(config-if)# ip router ospf 0 area 1
Switch(config-if)# exit
```

5. Configure route summarization by specifying the starting address and mask of the range of addresses to be summarized.

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area 1 range 36.128.192.0/18
Switch(config-router)# exit
```

6. Use the hide command to prevent a range of addresses from advertising to the backbone.

```
Switch(config)# routing-protocol ospf
Switch(config-router)# area 1 range 36.128.200.0/24 not-advertise
Switch(config-router)# exit
```

Verifying OSPF Configuration

Use the following commands to verify the OSPF configuration on your switch:

```
Switch> display ip ospf
Switch> display ip ospf border-routers
Switch> display ip ospf neighbor
Switch> display ip ospf database
```

Refer to the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6* for information on the preceding commands.

Chapter 24. Route Maps

Route maps are used to define route policy by permitting or denying certain routes based on configured set of rules. Each route map consists of multiple clauses that are ordered by their sequence number.

This section discusses the following topics:

Route Maps Overview

Route maps define conditions for redistributing routes from one routing protocol to another. Also, route maps are widely used by routing protocols such as BGP or OSPF, which apply route maps to filter traffic.

The more common applications of route maps are listed below:

- route filtering during redistribution between routing protocols;
- route control and attribute modification of BGP routes;
- route metric modification during redistribution between routing protocols.

Route maps can be thought as conditional statements that the switch will consider before examining its routing table. They take precedence over routing protocols.

For example, you can define and use a route map when you want traffic to follow a certain path across the network. This path can be different from the usual path that the routing protocol forwards traffic on. The switch will examine the route map and if traffic meets the specified conditions, it will forward traffic across the selected network path. If traffic does not meet the conditions, it will be forwarded according to the routing table.

To create or modify a route map, use the following command:

```
Switch(config)# route-map <route map name>
Switch(config-route-map)#
```

Note: After creating a route map, you will enter Route Map Configuration mode.

Note: By default, if the rule (permit or deny) is not specified, the switch will configure the route map as a permit rule. Also, if the sequence number is not specified, the switch will configure sequence 10.

To view information about a route map, use the following command:

```
Switch(config)# display route-map <route map name>
```

To delete a route map, use the following command:

```
Switch(config)# no route-map <route map name>
```

Permit and Deny Rules

Route maps contain conditional clauses (match and apply) that can be used to permit or deny routes.

A permit rule means that a route must meet all the match clauses to be allowed. A deny rule means that a route that meets at least one of the match clauses will be dropped.

For example, route map is configured to use a prefix-list as a match clause. If the route map is set as a permit rule, then routes that are allowed by the prefix-list will be redistributed.

If the route map is set as a deny rule, then routes that are allowed by the prefix-list will not be redistributed.

To configure a route map permit rule, use the following command:

```
Switch(config)# route-map <route map name> permit
Switch(config-route-map)#
```

Note: After running the above command, you will enter Route Map Configuration mode, where you can create match and apply clauses that will be used to check if a route will be redistributed or not.

Rules are configured with a sequence number. The switch will check routes against permit or deny rules in ascending order of their sequence number. Rules with a lower sequence number will be checked before rules with a higher sequence number. A route map can have up to a maximum of 65535 sequences.

Note: If the sequence number is not specified, the switch will configure match and apply clauses for sequence 10.

Note: It is recommended that you use multiples of 10 when configuring sequences, in case you require to add more rules between already existing ones.

To configure a route map permit rule and specify its sequence number, use the following command:

```
Switch(config)# route-map <route map name> permit <1-65535 (sequence number)>
```

To configure a route map deny rule and specify its sequence number, use the following command:

```
Switch(config)# route-map <route map name> deny <1-65535>
```

Note: A route map cannot have two rules that share the same sequence number. The newer rule will replace the older one.

For example, sequence 20 is configured as a deny rule. If you then set it up as a permit rule, the new configuration will overwrite the old one. Thus, sequence 20 will become a permit rule.

To delete a rule (permit or deny), use the following command:

```
Switch(config)# no route-map <route map name> {permit|deny} <1-65535>
```

Match and Apply Clauses

A rule consists of two types of clauses: match and apply. You can consider these clauses in terms of 'IF' and 'THEN' statements from many computer programming languages. If a match clause is true, then a apply clause is executed.

To configure a clause, you must enter Route Map Configuration mode. To achieve this, use the following command:

```
Switch(config)# route-map <route map name> {deny|permit} <1-65535 (sequence number)>
Switch(config-route-map)#
```

Match clauses or conditions are used to create permit or deny rules. Only routes that meet the specified match clauses will be redistributed or rejected, depending on the type of the rule.

If multiple rules are created, the switch checks every rule in ascending order of their sequence number. If the first match clause is not met by the route, the clause is not applicable to the route and the switch checks the route against the next match clause.

This goes on until the route meets the conditions of a rule or until the switch has checked all sequences. If a route did not meet all the conditions of a single permit rule, it will be rejected. If a route did not met any of the conditions of any deny rule, it will be redistributed.

Note: Multiple match or apply clauses can be configured for each rule. Each clause will be checked in the order they where configured.

To add or remove a match clause, use the following command:

```
Switch(config-route-map)# [no] match <condition>
```

You can create a match clause for the conditions:

- match BGP AS path list;
- match BGP community list;
- match BGP extended community list;
- match next hop interface;
- match destination IP address;
- match next hop IP address;
- match route metric;
- match BGP origin code;
- match route type;
- match tag value.

For example, the following command will create a match clause for routes that have a metric of 1500:

```
Switch(config-route-map)# match metric 1500
```

For more details about the match clause commands, consult the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

Note: Multiple match clauses of the same type can be configured for a single rule. When being checked against the match clauses, the route must met only one of the configured match clauses of the that type.

For example, you if configure a permit rule with two match clauses for route metric (one for 1500 and another for 3000), a route must have either a metric of 1500 or 3000 to met the condition of the rule and be redistributed.

Note: If no match clauses are configured for a rule, all routes will meet the conditions.

Apply clauses modify the information of route when the match clauses are met.

To add or remove a apply clause, use the following command:

```
Switch(config-route-map)# [no] apply <condition>
```

When the route matches the conditions of the rule, you can modify the following parameters:

● set the route aggregator attribute;

● set the BGP AS path;

● set the route atomic aggregate attribute;

● delete the BGP community list;

● set the BGP community list;

● set the BGP extended community list;

● set the BGP route dampening attributes;

● set the IP address of the next hop;

● set the BGP local preference attribute;

● set the route metric;

● set the route metric type;

● set the BGP origin code;

● set the BGP originator ID;

● set the tag value;

● set the BGP route weight attribute.

For example, the following command will create a set clause that will modify the originator ID of a BGP route:

```
Switch(config-route-map)# apply originator-id 10.10.125.60
```

For more details about the match clause commands, consult the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

Note: If no apply clauses are configured for a permit rule, the route is redistributed without any modifications.

Route Maps Configuration Example

Following are the basic steps and commands for configuring route maps.

1. Configure a route map rule (for example, a permit rule for route map rmap):

```
Switch(config)# route-map rmap permit 10
Switch(config-route-map)#
```

2. Configure a match clause for the rule (for example, match the first hop interface to ethernet port 1/10):

```
Switch(config-route-map)# match interface ethernet 1/10
```

3. Optionally, you can configure an apply clause for any matching routes (for example, set the routes next hop address):

```
Switch(config-route-map)# apply ip next-hop 120.78.33.78
```

4. Optionally, you can configure additional rules (for example, deny routes with destination IP addresses included in prefix list PrefDeny or have a metric of 200):

```
Switch(config-route-map)# exit

Switch(config)# route-map rmap deny 20
Switch(config-route-map)# match ip address prefix-list PrefDeny
Switch(config-route-map)# match metric 200
Switch(config-route-map)# exit
```

5. Check the route map configuration:

```
Switch(config)# display route-map rmap

route-map rmap, permit, sequence 10
  Match clauses:
    interface: Ethernet1/10
  Set clauses:
    ip next-hop 120.78.33.78

route-map rmap, deny, sequence 20
  Match clauses:
    ip address prefix-list: PrefDeny
    metric: 200
  Set clauses:
```

Part 5: High Availability Fundamentals

Internet traffic consists of myriad services and applications which use the Internet Protocol (IP) for data delivery. However, IP is not optimized for all the various applications. High Availability goes beyond IP and makes intelligent switching decisions to provide redundant network configurations.

This section discusses the following topics:

- *"Basic Redundancy" on page 539*
- *"Virtual Router Redundancy Protocol" on page 543*
- *"Layer 2 Failover" on page 563*

Chapter 25. Basic Redundancy

Lenovo Cloud Network Operating System 10.6 includes various features for providing basic link or device redundancy:
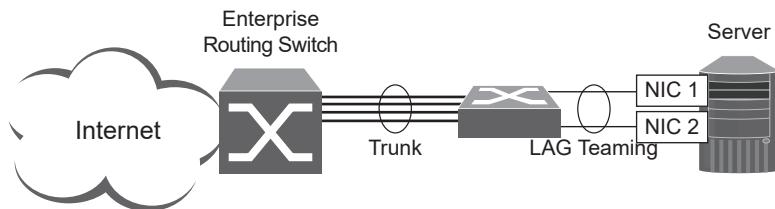
- "Aggregating for Link Redundancy" on page 540
- "Virtual Link Aggregation" on page 541

Aggregating for Link Redundancy

Multiple switch ports can be combined together to form robust, high-bandwidth LAGs to other devices. Since LAGs are comprised of multiple physical links, the LAG is inherently fault tolerant. As long as one connection between the switches is available, the LAG remains active.

In Figure 28, four ports are aggregated together between the switch and the enterprise routing device. Connectivity is maintained as long as one of the links remain active. The ports on the server are in LAG teaming, allowing the secondary NIC to take over in the event that the primary NIC link fails.

Figure 28. Aggregating Ports for Link Redundancy

For more information on aggregation, see Chapter 11, "Ports and Link Aggregation."

Virtual Link Aggregation

Using the VLAG feature, switches can be paired as VLAG peers. The peer switches appear to the connecting device as a single virtual entity for the purpose of establishing a multi-port LAG. The VLAG-capable switches synchronize their logical view of the access layer port structure and internally prevent implicit loops. The VLAG topology also responds more quickly to link failure and does not result in unnecessary MAC flooding.

VLAGs are useful in multi-layer environments for both uplink and downlink redundancy to any regular LAG-capable device. They can also be used in for active-active VRRP connections.

For more information on VLAGs, see Chapter 13, "Virtual Link Aggregation Groups."

Chapter 26. Virtual Router Redundancy Protocol

The switch supports high-availability network topologies through an enhanced implementation of the Virtual Router Redundancy Protocol (VRRP).

The following topics are discussed in this chapter:

- "VRRP Overview" on page 544. This section discusses VRRP operation and Cloud NOS redundancy configurations.

- "Failover Methods" on page 548. This section describes the three modes of high availability.

- "Cloud NOS Extensions to VRRP" on page 549. This section describes VRRP enhancements implemented in CNOS.

- "Configuring the Switch for Tracking" on page 555. This section describes configuring the switch for tracking.

- "Basic VRRP Configuration" on page 556. This section offers a simple example of a basic virtual router configuration consisting of two VRRP routers.

- "Configuring VRRP High-Availability Using Multiple VIRs" on page 558. This section discusses the more useful and easily deployed redundant configurations.

VRRP Overview

Typically, end devices in a LAN are connected to a larger network using a single router (first-hop router). They have the router's IP address configured as their default gateway. While, this method of configuration has its advantages, its main problem is that it creates a single point-of-failure if the first-hop router goes down.

In a high-availability network topology, no device can create a single point-of-failure for the network or force a single point-of-failure to any other part of the network. This means that your network will remain in service despite the failure of any single device. To achieve this usually requires redundancy for all vital network components.

VRRP offers a higher availability default path without the need for configuring dynamic routing or router discovery protocols on end devices.

VRRP enables redundant router configurations within a LAN, providing alternate router paths for a host to eliminate single points-of-failure within a network. Each participating VRRP-capable routing device is configured with the same virtual router IP address and ID number. One of the VRRP routers is elected as the master, based on a number of priority criteria, and assumes control of the shared virtual router IP address. If the master fails, one of the backup virtual routers will take control of the virtual router IP address and actively process traffic addressed to it.

With VRRP, Virtual Interface Routers (VIR) allow two VRRP routers to share an IP address across the routers. VIRs provide a single Destination IP (DIP) address for upstream routers to reach various servers and provide a virtual default Gateway for the servers. Also, end devices are configured to the virtual router's IP address as their default gateway.

Also, while using VRRP, you can configure two VRRP routers to evenly share traffic during normal running conditions. To achieve this, you assign the first VRRP router's virtual IP address as the default gateway for half of the end devices, and the second VRRP router's IP address as the default gateway for the other half of the end devices. In this scenario, both VRRP routers are members of two different virtual routers. Each VRRP router acts both as the master virtual router in one of the two virtual routers and the backup virtual router in the other virtual router.

Cloud NOS uses VRRP version 3 (VRRPv3), which implements support for IPv6 addresses. For more details, see RFC 5798.

Note: VRRP can be enabled only on Layer 3 interfaces, which on the switch are ethernet interfaces configured as routed ports or Switch Virtual Interfaces (SVIs).

By default, all ports are configured as access switch ports. To configure a switch port as a routed port, use the following command (for this example, ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# no bridge-port
```

For more details on routed ports, see "Configuring a Routed Port" on page 408.

VRRP Components

Each physical router running VRRP is known as a *VRRP router*.

Virtual Router

Two or more VRRP routers can be configured to form a *virtual router* (RFC 5798). Each VRRP router may participate in one or more virtual routers. Each virtual router consists of a user-configured *virtual router identifier* (VRID) and an IP address. You can configure up to 255 virtual routers.

Virtual Router MAC Address

The VRID is used to build the *virtual router MAC Address*. The five highest-order octets of the virtual router MAC Address are the standard MAC prefix (00-00-5E-00-01) as defined in RFC 5798. The VRID is used to form the lowest-order octet.

Owners and Renters

Only one of the VRRP routers in a virtual router may be configured as the IP address owner. This router has the virtual router's IP address as its real interface address. This router responds to packets addressed to the virtual router's IP address for ICMP pings, TCP connections and so on.

There is no requirement for any VRRP router to be the IP address owner. Most VRRP installations choose not to implement an IP address owner. For the purposes of this chapter, VRRP routers that are not the IP address owner are called *renters*.

Master and Backup Virtual Router

Within each virtual router, one VRRP router is selected to be the master virtual router. See "Selecting the Master VRRP Router" on page 546 for an explanation of the selection process.

Note: If the IP address owner is available, it will always become the virtual router master.

The master virtual router forwards packets sent to the virtual router. It also responds to Address Resolution Protocol (ARP) requests sent to the virtual router's IP address. Finally, the virtual router master sends out periodic advertisements to let other VRRP routers know it is alive and its priority.

Within a virtual router, the VRRP routers not selected to be the master are known as virtual router backups. If the virtual router master fails, one of the virtual router backups becomes the master and assumes its responsibilities.

Virtual Interface Router

At Layer 3, a Virtual Interface Router (VIR) allows two VRRP routers to share an interface across the routers. VIRs provide a single Destination IP (DIP) address for upstream routers to reach various destination networks and provide a virtual default Gateway.

Note: Every VIR must be assigned to an interface and every interface must be assigned to a VLAN. If no port in a VLAN has link up, the interface of that VLAN is down and if the interface of a VIR is down, that VIR goes into INIT state.

Assigning VRRP Virtual Router ID

When configuring a VRRP virtual router, you must assign it a virtual router identifier (ID). The virtual router ID may be configured as any number between 1 and 255. Use the following command to configure the virtual router ID:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vrrp <1-255>
Switch(config-if-vrrp)#
```

Note: After creating a virtual router with the specified ID, you will enter the VRRP Configuration menu for that virtual router. Also, you can use the same command to configure any existing virtual routers.

Note: This is the configuration method for IPv4 addressing. For IPv6 addressing, see "IPv6 VRRP" on page 553.

VRRP Operation

Only the virtual router master responds to ARP requests. Therefore, the upstream routers only forward packets destined to the master. The master also responds to ICMP ping requests if it is the owner of the pinged IP address or has accept mode enabled. The backup does not forward any traffic, nor does it respond to ARP requests.

If the master is not available, the backup becomes the master and takes over responsibility for packet forwarding and responding to ARP requests.

Selecting the Master VRRP Router

Each VRRP router is configured with a priority between 1–254. A bidding process determines which VRRP router is or becomes the master—the VRRP router with the highest priority.

The master periodically sends advertisements to an IP multicast address. As long as the backups receive these advertisements, they remain in the backup state. If a backup does not receive an advertisement for three advertisement intervals, it initiates a bidding process to determine which VRRP router has the highest priority and takes over as master. In addition to the three advertisement intervals, a manually set hold-off time can further delay the backups from assuming the master status.

If, at any time, a backup determines that it has higher priority than the current master does, it can preempt the master and become the master itself, unless configured not to do so. In preemption, the backup assumes the role of master and begins to send its own advertisements. The current master sees that the backup has higher priority and will stop functioning as the master.

A backup router can stop receiving advertisements for one of two reasons—the master can be down or all communications links between the master and the backup can be down. If the master has failed, it is clearly desirable for the backup (or one of the backups, if there is more than one) to become the master.

Note: If the master is healthy but communication between the master and the backup has failed, there will then be two masters within the virtual router. To prevent this from happening, configure redundant links to be used between the switches that form a virtual router.
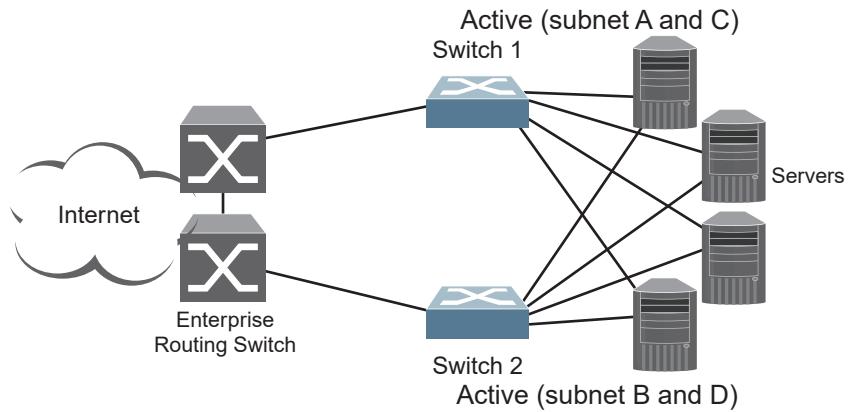
Failover Methods

With service availability becoming a major concern on the Internet, service providers are increasingly deploying Internet traffic control devices, such as application switches, in redundant configurations. Cloud NOS high availability configurations are based on VRRP. The implementation of VRRP includes proprietary extensions.

Active-Active Redundancy

In an active-active configuration, shown in Figure 29, two switches provide redundancy for each other, with both active at the same time. Each switch processes traffic on a different subnet. When a failure occurs, the remaining switch can process traffic on all subnets. For more details and a configuration example, see "Configuring VRRP High-Availability Using Multiple VIRs" on page 558.

Figure 29. Active-Active Redundancy

Cloud NOS Extensions to VRRP

This section describes VRRP enhancements that are implemented in CNOS:

-
-
-
-
-
-
-
-

VRRP Advertisement Interval and Sub-second Failover

The master virtual router periodically sends VRRP advertisements to the backup virtual routers to keep them in their backup state. The master virtual router waits a certain amount of time between each advertisement. This period of time is called advertisement interval.

To configure the advertisement interval (in centiseconds or hundredths of a second), use the following command (for this example, interface ethernet 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vrrp <1-255>
Switch(config-if-vrrp)# advertisement-interval <5-4095>
```

The default value of the advertisement interval is 100 centiseconds or 1 second. To reset the advertisement interval to this value, use the following command:

```
Switch(config-if-vrrp)# no advertisement-interval
```

VRRPv3 supports the sub-second detection of master virtual router failure. To mitigate the potential loss of packets that happens in the time interval between when the master virtual router fails and when the backup virtual router takes its place, the advertisement interval can be configured to lower values than one second.

A backup virtual router detects that the master virtual router is down when three advertisement intervals have passed without the reception of any advertisements from the master. After the backup virtual router detects the failure, it will assume the role of master virtual router. During this time, any traffic that is sent to the downed master virtual router will be lost.

To lessen the volume of lost traffic, you can configure the advertisement interval to sub-second values, thus reducing the time required for the backup virtual router to detect the failure and take the master virtual router's place.

Interface Tracking

VRRP interface tracking allows the tracking of interface states which can modify the priority level of a VRRP router member of a virtual router group. If the tracked interface goes down or if it does not have a configured virtual IP address, the priority level of the virtual router group is lowered by a given value. This allows another VRRP router to become the new group master virtual router.

Only one interface can be tracked at a time for every virtual router group and the tracked interface must be a Layer 3 interface. VRRP does not support Layer 2 interface tracking.

Note: The interface on which you configure VRRP tracking cannot be the interface that is being tracked. For example, ethernet interface 1/1 can be configured to track ethernet interface 1/2, 1/10 or 1/20, but not itself.

You can configure the value that is subtracted from a virtual router group's priority level when the state of the tracked interface changes to down. When the tracked interface returns to an up state, VRRP restores the original priority for the virtual router group.

By default, the tracking of interfaces is disabled.

To configure the subtracted value, use the following command (for this example, interfaces ethernet 1/10 and 1/12 are used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vrrp <1-255>
Switch(config-if-vrrp)# track interface ethernet 1/10 priority <1-253>
```

Note: VRRP preempt mode must be enabled. For more details, see "VRRP Preemption" on page 552.

To disable the tracking of an interface, use the following command:

```
Switch(config-if-vrrp)# no track interface ethernet 1/10 priority <1-253>
```

Switch Back Delay

Other Layer 3 routing protocols, like OSPF, have a slower convergence time than VRRP. When these routing protocols are enabled on the switch, there can be a situation where the route table of the master virtual router is not completely populated when the VRRP router exchanges roles from master to backup virtual router and then vice versa. Remote network routes can be lost during the convergence time of these routing protocols, resulting in discarded traffic.

To avoid such a scenario, a time delay is used. The delay prevents the original master virtual router from becoming the master after it comes back online until a configured amount of time passes. This is called switch back delay.

By default, the switch back delay is disabled.

To configure the switch back delay (in milliseconds), use the following command (for this example, ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vrrp <1-255>
Switch(config-if-vrrp)# switch-back-delay <1-500000>
```

To disable the switch back delay, use the following command:

```
Switch(config-if-vrrp)# no switch-back-delay
```

Backward Compatibility with VRRPv2

The Switch uses VRRPv3, which supports IPv6 addresses, but can be configured to be compatible with VRRPv2 as well.

Note: This is only compatible with IPv4 addresses, since VRRPv2 does not have support for IPv6 addressing.

When VRRPv2 compatibility is enabled on the master virtual router, it sends both VRRPv2 and VRRPv3 advertisements at the configured advertisement interval. For more details about the advertisement interval, see "VRRP Advertisement Interval and Sub-second Failover" on page 549.

To enable or disable VRRPv2 backward compatibility, use the following command (for this example, ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vrrp <1-255>
Switch(config-if-vrrp)# [no] v2-compatible
```

A backup virtual router having VRRPv2 backward compatibility enabled will ignore VRRPv2 advertisements from the master virtual router if it also receives VRRPv3 advertisements.

If a backup virtual router has VRRPv2 backward compatibility enabled and it receives only VRRPv2 advertisements from the master virtual router, it will transform the received advertisement interval into centiseconds.

If the master virtual router has VRRPv2 backward compatibility enabled and the advertisement interval set to a sub-second value, it will ignore this setting and send advertisements to the backup virtual routers every one second.

VRRP Accept Mode

The Switch supports VRRP Accept mode. It makes the master virtual router to accept packets with IP addresses not own by it. Thus, such packets are accepted by the master virtual router if it is the owner of the IP address or accept mode is enabled. The master virtual router will discard any traffic if it is not the owner of the IP address and accept mode is disabled.

If Accept mode is disabled, the master virtual router accepts packets sent to the virtual router only if it is the IP address owner. Any packets sent to the master virtual router with an IP address that it does not own will be discarded.

This restriction causes problems when troubleshooting network connectivity issues. When a host is unreachable, it is common practice to ping the first-hop router to determine if the problem is related to the default gateway of the host. If the host's default gateway is a virtual router that does not respond to pings (or ICMP Echo Requests), this method of troubleshooting is unavailable for use.

Accept mode configures the master virtual router to respond only to ping, traceroute and telnet packets with an IP address not owned by the router. The master virtual router will not respond to any other type of packets. When it responds, it uses the virtual router's IP and MAC addresses.

By default, VRRP accept mode is enabled.

To enable or disable VRRP accept mode, use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vrrp <1-255>
Switch(config-if-vrrp)# [no] accept-mode
```

VRRP Preemption

When VRRP preemption is enabled on a backup virtual router, if the router has a higher priority than the current master virtual router, it will preempt the lower priority master and assume control.

Note: Even when VRRP preemption is disabled, a backup virtual router will always preempt any other master if the router is the owner of the IP address associated with the virtual router.

By default, VRRP preemption is enabled.

To enable or disable VRRP preemption, use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vrrp <1-255>
Switch(config-if-vrrp)# [no] preempt
```

VRRP Priority

Priority is used by VRRP in the master virtual router election process. Each VRRP router participating in the election process is assign a priority level between 0 and 255 and the VRRP router with the highest priority becomes the master virtual process.

You can configure a VRRP router with a priority between 1 and 254. The value of 255 is reserved for the VRRP router that owns the IP address associated with the virtual router. The value of 0 is reserved for the master virtual router to indicate that it has stopped participating in the VRRP. This triggers the backup virtual routers to quickly start a new master election process without waiting for the current master virtual router to timeout.

To configure the priority of a VRRP router, use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vrrp <1-255>
Switch(config-if-vrrp)# priority <1-254>
```

By default, the priority of a VRRP router is 100. To reset the priority to its default value, use the following command:

```
Switch(config-if-vrrp)# no priority
```

Note: If you configure the virtual IP address of the virtual router to have the same IP address as the interface you set up a VRRP router, the previous priority configuration of the VRRP router will be deleted and its new priority will be 255.

Note: If a VRRP router is the owner of the virtual IP address of the virtual router, having a priority of 255, and you configure the virtual IP address to an IP address different than the one owned by the VRRP router, the priority of that VRRP router will be reset to the default.

IPv6 VRRP

VRRPv3 supports both IPv4 and IPv6 addressing. The type of IP addressing used by the VRRP router must be the same as used by the interface on which the VRRP router is configured. For example, if an ethernet interface is configured as an IPv4 address, the VRRP router created on that interface must also be configured with an virtual IPv4 address.

To create an IPv6 VRRP session, use the following command (for this example, ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vrrp <1-255> ipv6
Switch(config-if-vrrp)#
```

Note: If there is a mismatch between the type of the virtual IP address and the type of the IP address used by the interface or by the VRRP session, you will be prompted with an error message.

When using IPv6 addressing, each VRRP session must be configured with a set of two IPv6 addresses: a link-local address as the primary virtual address and a global IPv6 address.

Note: The link-local IPv6 address must be different than the global IPv6 address.

The link-local address must be the first address in the IPv6 address list used in the VRRP packet. If this address is the same as the virtual IP address associated with the virtual router, then the VRRP router is the owner of the virtual IP address and its priority is set to 255. The VRRP router is advertised with priority 255 in VRRP packets regardless of the global IPv6 address.

Note: The same link-local IPv6 address cannot be used in other VRRP sessions on the same interface.

Even if an interface is configured with the same IPv6 address as the virtual global IPv6 address of a VRRP router, the VRRP router is not considered the owner of the IPv6 address. The link-local address takes precedence over the global address.

Note: Only one global IPv6 address and one link-local address are supported for each VRRP IPv6 session.

To configure a global IPv6 address, use the following command:

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# vrrp <1-255> ipv6
Switch(config-if-vrrp)# address <IPv6 address>
```

To delete the global IPv6 address of a VRRP session, use the following command:

```
Switch(config-if-vrrp)# no address <IPv6 address>
```

To configure the link-local IPv6 address of the VRRP session, use the following command:

```
Switch(config-if-vrrp)# address <IPv6 address> primary
```

To delete the link-local IPv6 address of a VRRP session, use the following command:

```
Switch(config-if-vrrp)# no address <IPv6 address> primary
```

To delete all configured IP addresses of a VRRP session, use the following command:

```
Switch(config-if-vrrp)# no address
```

Configuring the Switch for Tracking

Tracking configuration largely depends on user preferences and network environment. Consider the configuration shown in Figure 29 on page 548. Assume the following behavior on the network:

- Upon initialization, switch 1 is the Master Virtual Router and switch 2 is the Backup Virtual Router.

- Switch 1 and 2 are both connected to the each of the four servers on the local network.

- If switch 1 is the master and one of its interfaces goes down, meaning switch 1 is still connected to three out of the four servers, then switch 1 remains the Master Virtual Router.

  This behavior is preferred because running one server down is less disruptive than bringing a new master online and severing all active connections in the process.

- If switch 1 is the master and two or more of its interfaces go down, meaning switch 1 is not connected to at least three of the four servers, then switch 2 becomes the new Master Virtual Router and switch 1 becomes the Backup Virtual Router.

- If switch 2 is the master, it will retain its role as the Master Virtual Router, even when the links between switch 1 and all the servers are restored, as long as switch 2 has at least the same number of actively connected servers as switch 1.

- If switch 2 is the master and the number of actively connected servers is lower than the ones connected to switch 1, it looses its role as the Master Virtual Router. Switch 1 regains its role as the Master Virtual Router and switch 2 becomes the Backup Virtual Router.

You can implement this behavior by configuring the switch for tracking as follows:

1. Set the priority for switch 1 to 101. For more details, see "VRRP Priority" on page 552.

2. Leave the priority for switch 2 at the default value of 100.

3. On both switches, enable tracking based on ethernet interfaces or VLANs. For more details, see "Interface Tracking" on page 550.

   Note: There is no shortcut to setting tracking parameters. The goals must first be set and the outcomes of various configurations and scenarios analyzed to find settings that meet the goals.

Basic VRRP Configuration

The following steps are an example of a basic VRRP configuration:

Configuring Switch 1:

1. Enter Global Configuration command mode:

```
Switch> enable
Switch# configure device
Switch(config)#
```

2. Configure the ethernet port to be part of the virtual router:

```
Switch(config)# interface ethernet 1/1
Switch(config-if)# no bridge-port
```

3. Assign an IP address to the interface:

   a. either an IPv4 address:

```
Switch(config-if)# ip address 192.168.1.10/24
```

   b. or an IPv6 address:

```
Switch(config-if)# ipv6 address 2001:db8:a0b:12f0::10/32
```

4. Create a virtual router with an assigned ID and configure its virtual IP address:

   a. either using IPv4 addressing:

```
Switch(config-if)# vrrp 1
Switch(config-if-vrrp)# address 192.168.1.1
```

   b. or IPv6 addressing:

```
Switch(config-if)# vrrp 1 ipv6
Switch(config-if-vrrp)# address fe80::2001 primary
Switch(config-if-vrrp)# address 2001:db8:a0b:12f0::76e
```

   Note: The virtual IP address of the VRRP router must be same type as the IP address of the interface on which VRRP is configured.

5. Set the priority of the VRRP router and enable VRRP:

```
Switch(config-if-vrrp)# priority 110
Switch(config-if-vrrp)# no shutdown
```

   Note: Switch 1 has a higher priority and will become the master virtual router.

6. Check the VRRP configuration:

```
Switch# display vrrp detail
```

Configuring Switch 2:

1. Enter Global Configuration command mode:

```
Switch> enable
Switch# configure device
Switch(config)#
```

2. Configure the ethernet port to be part of the virtual router:

```
Switch(config)# interface ethernet 1/1
Switch(config-if)# no bridge-port
```

3. Assign an IP address to the interface:

    a. either an IPv4 address:

```
Switch(config-if)# ip address 192.168.1.20/24
```

    b. or an IPv6 address:

```
Switch(config-if)# ipv6 address 2001:db8:a0b:12f0::20/32
```

4. Create a virtual router with an assigned ID and configure its virtual IP address:

    a. either using IPv4 addressing:

```
Switch(config-if)# vrrp 1
Switch(config-if-vrrp)# address 192.168.1.1
```

    b. or IPv6 addressing:

```
Switch(config-if)# vrrp 1 ipv6
Switch(config-if-vrrp)# address fe80::2001 primary
Switch(config-if-vrrp)# address 2001:db8:a0b:12f0::76e
```

Note: The virtual IP address of the VRRP router must be same type as the IP address of the interface on which VRRP is configured.

5. Enable VRRP:

```
Switch(config-if-vrrp)# no shutdown
```
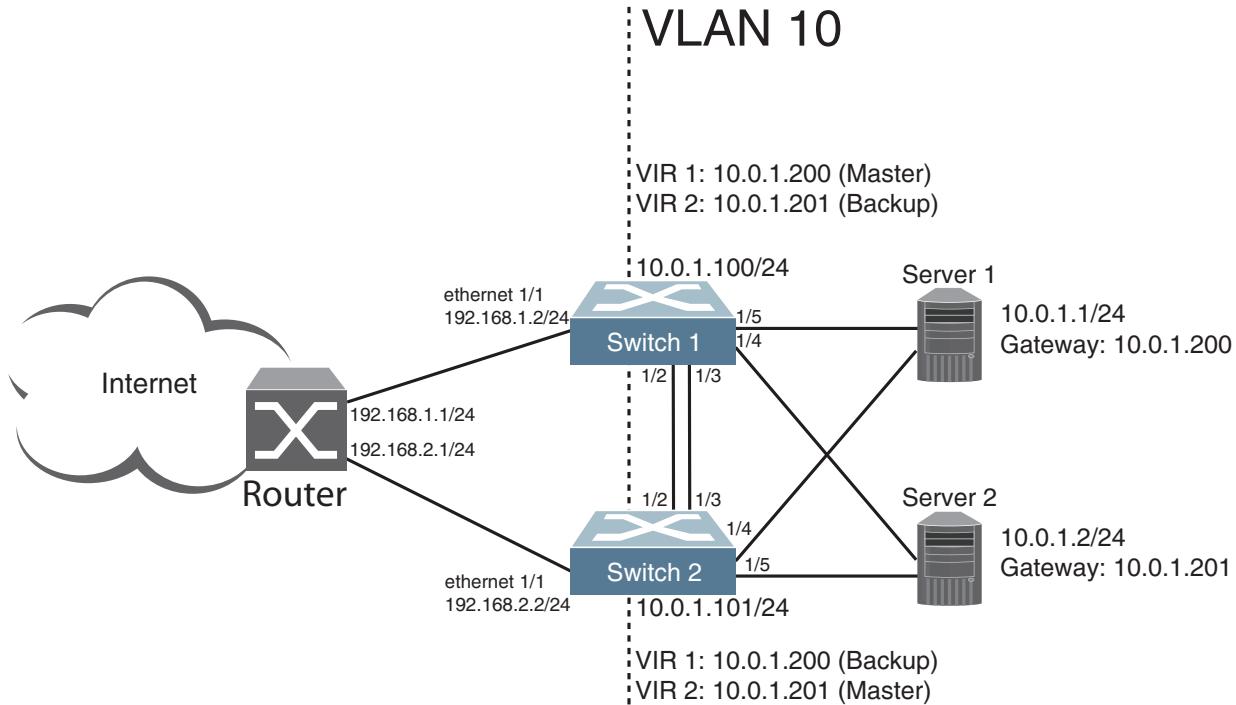
6. Check the VRRP configuration:

```
Switch# display vrrp detail
```

Configuring VRRP High-Availability Using Multiple VIRs

Figure 30 shows an example configuration where two switches are used as VRRP routers in an active-active configuration. In this configuration, both switches respond to packets.

Figure 30. Active-Active Configuration using VRRP

Although this example shows only two switches, there is no limit on the number of switches used in a redundant configuration. It is possible to implement an active-active configuration across all the VRRP-capable switches in a LAN.

In the scenario illustrated in Figure 30, traffic destined for Server 1 (IP address: 10.0.1.1/24) is forwarded through the Router and ingresses on Switch 1 on ethernet port 1/1 (IP address: 192.128.1.2/24). Traffic that is received from Serve 1 is forwarded to the default gateway (192.168.1.1).

If the link between Switch 1 and the Router fails, Switch 2 becomes the Master because it has a higher priority. By configuring VRRP interface tracking, if the link between Switch 1 and the Router fails, the priority of Switch 1 will be lowered, thus allowing Switch 2 to become the new VRRP master.

Switches 1 and 2 communicate through a LAG consisting of ethernet ports 1/2 and 1/3 on each switch. The ports are all members of the same VLAN (VLAN 10).

Switches 1 and 2 have Servers 1 and 2 connected to them through ports 1/4 and 1/5, respectively. These ports are all members of the same VLAN (VLAN 10).

To implement the active-active example, perform the following switch configuration.

Task 1: Configure Switch 1

1. Configure ethernet port 1/1 as a routed port.

```
Switch(config)# interface ethernet 1/1
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 192.168.1.2/24
Switch(config-if)# exit
```

2. Configure the default gateway, which points to the Router.

```
Switch(config)# ip route 0.0.0.0/0 ethernet 1/1 192.168.1.1
```

3. Create VLAN 10 and configure server interfaces (ports 1/4 and 1/5).

```
Switch(config)# vlan 10
Switch(config-vlan)# exit

Switch(config)# interface ethernet 1/4-5
Switch(config-if-range)# bridge-port access vlan 10
Switch(config-if-range)# exit
```

4. Configure two Virtual Interface Routers.

```
Switch(config)# interface vlan 10
Switch(config-if)# ip address 10.0.1.100/24
Switch(config-if)# vrrp 1
Switch(config-if-vrrp)# address 10.0.1.200
Switch(config-if-vrrp)# track interface ethernet 1/1 priority 5
Switch(config-if-vrrp)# priority 101
Switch(config-if-vrrp)# no shutdown
Switch(config-if-vrrp)# exit

Switch(config-if)# vrrp 2
Switch(config-if-vrrp)# address 10.0.1.201
Switch(config-if-vrrp)# no shutdown
Switch(config-if-vrrp)# exit
Switch(config-if)# exit
```

5. Create a LAG and make ethernet ports 1/2 and 1/3 members of that LAG.

```
Switch(config)# interface port-aggregation 1
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 10
Switch(config-if)# exit

Switch(config)# interface ethernet 1/2-3
Switch(config-if-range)# aggregation-group 1 mode active
Switch(config-if-range)# exit
```

6. Verify the VRRP configuration.

```
Switch# display vrrp

  Interface     VR IpVer Pri Time   Pre  State   VR IP addr
-----------------------------------------------------------------
  Vlan10        1   IPV4 101 100  cs  Y   Master  10.0.1.200
  Vlan10        2   IPV4 100 100  cs  Y   Backup  10.0.1.201
```

Task 2: Configure Switch 2

1. Configure ethernet port 1/1 as a routed port.

```
Switch(config)# interface ethernet 1/1
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 192.168.2.2/24
Switch(config-if)# exit
```

2. Configure the default gateway, which points to the Router.

```
Switch(config)# ip route 0.0.0.0/0 ethernet 1/1 192.168.2.1
```

3. Create VLAN 10 and configure server interfaces (ports 1/4 and 1/5).

```
Switch(config)# vlan 10
Switch(config-vlan)# exit

Switch(config)# interface ethernet 1/4-5
Switch(config-if-range)# bridge-port access vlan 10
Switch(config-if-range)# exit
```

4. Configure two Virtual Interface Routers.

```
Switch(config)# interface vlan 10
Switch(config-if)# ip address 10.0.1.101/24
Switch(config-if)# vrrp 1
Switch(config-if-vrrp)# address 10.0.1.200
Switch(config-if-vrrp)# no shutdown
Switch(config-if-vrrp)# exit

Switch(config-if)# vrrp 2
Switch(config-if-vrrp)# address 10.0.1.201
Switch(config-if-vrrp)# track interface ethernet 1/1 priority 5
Switch(config-if-vrrp)# priority 101
Switch(config-if-vrrp)# no shutdown
Switch(config-if-vrrp)# exit
Switch(config-if)# exit
```

5. Create a LAG and make ethernet ports 1/2 and 1/3 members of that LAG.

```
Switch(config)# interface port-aggregation 1
Switch(config-if)# bridge-port mode trunk
Switch(config-if)# bridge-port trunk allowed vlan 10
Switch(config-if)# exit

Switch(config)# interface ethernet 1/2-3
Switch(config-if-range)# aggregation-group 1 mode active
Switch(config-if-range)# exit
```

6. Verify the VRRP configuration.

```
Switch# display vrrp

   Interface     VR  IpVer Pri Time    Pre  State   VR IP addr
-------------------------------------------------------------------
   Vlan10        1   IPV4  100 100  cs  Y   Backup  10.0.1.200
   Vlan10        2   IPV4  101 100  cs  Y   Master  10.0.1.201
```

Chapter 27. Layer 2 Failover

The main purpose of Layer 2 Failover is to support Network Adapter Teaming. With Network Adapter Teaming, all the NICs on each server share the same IP address, and are configured into a team. One NIC is the primary link, and the other is a standby link. For more details, refer to the documentation for your Ethernet adapter.

The following topics are covered in this chapter:

- "Monitoring LAG Links" on page 564
- "Setting the Failover Limit" on page 565
- "Manually Monitoring Port Links" on page 566
- "L2 Failover with Other Features" on page 567
- "Configuration Guidelines" on page 568
- "Configuring Layer 2 Failover" on page 569

Note: Only two links per server can be used for Layer 2 LAG Failover (one primary and one backup). Network Adapter Teaming allows only one backup NIC for each server blade.
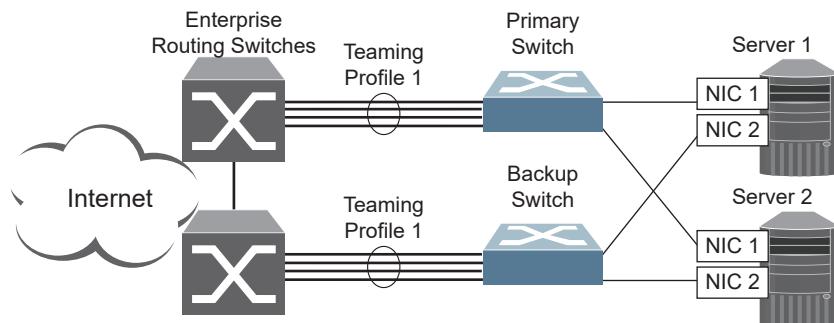
Monitoring LAG Links

Layer 2 Failover can be enabled on any LAG on the switch, including LACP LAGs. LAGs can be added to failover teaming profile groups. Then, if some specified number of monitor links fail, the switch disables all the control ports in the switch. When the control ports are disabled, it causes the NIC team on the affected servers to failover from the primary to the backup NIC. This process is called a failover event.

When the appropriate number of links in a monitor group return to service, the switch enables the control ports. This causes the NIC team on the affected servers to fail back to the primary switch (unless Auto-Fallback is disabled on the NIC team). The backup switch processes traffic until the primary switch's control links come up, which can take up to five seconds.

Figure 31 is an example of Layer 2 Failover. One switch is the primary, and the other is used as a backup. In this example, all ports on the primary switch belong to a single LAG, with Layer 2 Failover enabled, and Failover Limit set to 2. If one or no links in Teaming Profile 1 remain active, the switch temporarily disables all control ports in the teaming profile. This action causes a failover event on Server 1 and Server 2.

Figure 31. Basic Layer 2 Failover

Setting the Failover Limit

The failover limit lets you specify the minimum number of operational links required within each teaming profile before the teaming profile initiates a failover event. For example, if the limit is two, a failover event occurs when the number of operational monitor links in the teaming profile is one or zero. When you set the limit to one, the switch triggers a failover event only when no monitor links in the teaming profile are operational.

Manually Monitoring Port Links

The Manual Monitor (MMON) enables you to configure a set of ports or LAGs to monitor for link failures (a monitor list), and another set of ports or LAGs to disable when number of forwarding monitor links is less than the trigger limit (a control list). When the switch detects a link failure on the monitor list, it automatically disables the items in the control list. When server ports are disabled, the corresponding server's network adapter can detect the disabled link, and trigger a network-adapter failover to another port or LAG on the switch, or another switch.

The switch automatically enables the control list items when the monitor list items return to service.

Note: Control ports will not be re-enabled if they were manually shut down. Control ports will also not be re-enabled if they were shut down or error-disabled by another feature, such as BPDU Guard.

Monitor Port State

A monitor port is considered operational as long as the following conditions are true:

- The port must be in the `Link Up` state.
- If STP is enabled, the port must be in the `Forwarding` state.
- If the port is part of an LACP LAG, the port must be in the `Aggregated` state.

If any of these conditions is false, the monitor port is considered to have failed.

Control Port State

A control port is considered to be disabled only if the teaming profile is in the `triggered` state. The teaming profile is in the `triggered` state only when number of forwarding monitor links is less than limit of the teaming profile.

To view the state of any port, use one of the following commands:

```
Switch# display interface link          (View interface line status)
Switch# display spanning-tree            (View spanning tree status)
Switch# display port-aggregation summary (View port aggregation status)
Switch# display lacp port-aggregation    (View port LACP status)
```

L2 Failover with Other Features

L2 Failover works together with static LAGs, Link Aggregation Control Protocol (LACP), and with Spanning Tree Protocol (STP).

Static LAGs

When you add a portchannel (static LAG) to a monitor port of a teaming profile, any ports in that LAG become members of the trigger. You can also add a portchannel to the control port of a teaming profile, but the control ports of a profile must be different from the monitor ports of the profile.

LACP

Link Aggregation Control Protocol allows the switch to form dynamic LAGs. You can use dynamic LAGs in L2 Failover using the same procedures you use for static LAGs. For more information about LACP and dynamic LAGs, see "Link Aggregation Control Protocol" on page 266.

Spanning Tree Protocol

If Spanning Tree Protocol (STP) is enabled on the monitor ports in a teaming profile, the switch monitors the port STP state rather than the link state. A port failure results when STP is not in a Forwarding state (such as Learning, Discarding, or No Link) in all the VLAN or MSTP instances to which the port belongs. The switch automatically disables the appropriate control ports of this teaming profile.

When the switch determines that ports in the teaming profile are in STP Forwarding state in any one of the VLAN or MSTP instances to which the port belongs, then it automatically enables the appropriate control ports. The switch *fails back* to normal operation.

For example, if a monitor port is a member of VLAN instances 1, 2, and 3, a failover will be triggered only if the port is not in a forwarding state in all the three VLAN instances. When the port state in any of the three VLAN instances changes to forwarding, the control port is enabled and normal switch operation is resumed.

Configuration Guidelines

Follow these guidelines when configuring Layer 2 Failover.

- The `limit` option lets you specify the minimum number of operational monitors within a teaming profile. The profile will trigger control ports into a disabled state that would initiate a failover event only when the number of forwarding monitor ports is lower than this limit number.

- Management ports cannot be used as monitor ports or control ports.

- Member ports of an LACP LAG or a static LAG are not allowed to be either monitor ports or control ports.

- No overlapping port memberships are allowed within or across profiles.
    - Monitor ports or control ports of a profile cannot overlap with monitor ports or control ports of another profile.
    - Control ports of a profile must not overlap with its monitor ports.
    - Monitor interfaces of a profile cannot overlap with different teaming profiles.

Configuring Layer 2 Failover

Use the following procedure to configure Layer 2 failover MMON.

1. Enable teaming globally:

```
Switch(config)# teaming enable
```

2. Enable the teaming profile you plan to use.

```
Switch(config)# teaming profile <profile ID> enable
```

3. Specify the failover limit for your teaming profile.

```
Switch(config)# teaming profile <profile ID> limit <limit>
```

4. Specify the links to monitor.

```
Switch(config)# teaming profile <profile ID> mmon monitor interface
{ethernet <chassis>/<port>|port-aggregation <range>}
```

5. Specify the links to disable when the failover limit is reached.

```
Switch(config)# teaming profile <profile ID> mmon control interface
{ethernet <chassis>/<port>|port-aggregation <range>}
```

6. View the Layer 2 Failover configuration:

```
Switch(config)# display teaming profile
```

7. View the Layer 2 Failover operation information:

```
Switch(config)# display teaming profile information
```

Part 6: Network Management

This section discusses the following topics:

Chapter 28. Link Layer Discovery Protocol

Lenovo Cloud Network Operating System supports Link Layer Discovery Protocol (LLDP). This chapter discusses the use and configuration of LLDP on the switch:

- *"LLDP Overview" on page 574*
- *"Enabling or Disabling LLDP" on page 575*
- *"LLDP Transmit Features" on page 576*
- *"LLDP Receive Features" on page 580*
- *"LLDP Example Configuration" on page 584*

LLDP Overview

Link Layer Discovery Protocol (LLDP) is an IEEE 802.1AB-2005 standard for discovering and managing network devices. LLDP uses Layer 2 (the data link layer), and allows network management applications to extend their awareness of the network by discovering devices that are direct neighbors of already known devices.

With LLDP, the switch can advertise the presence of its ports, their major capabilities, and their current status to adjacent LLDP stations. LLDP transmissions occur on ports at regular intervals or whenever there is a relevant change to their status. The switch can also receive LLDP information advertised from adjacent LLDP-capable network devices.

In addition to discovery of network resources, and notification of network changes, LLDP can help administrators quickly recognize a variety of common network configuration problems, such as unintended VLAN exclusions.

The LLDP transmit function and receive function can be independently configured on a per-port basis. The administrator can allow any given port to transmit only, receive only, or both transmit and receive LLDP information.

The LLDP information to be distributed by the switch ports, and that which has been collected from other LLDP stations, is stored in the switch's Management Information Base (MIB). Network Management Systems (NMS) can use Simple Network Management Protocol (SNMP) to access this MIB information. LLDP-related MIB information is read-only.

Changes, either to the local switch LLDP information or to the remotely received LLDP information, are flagged within the MIB for convenient tracking by SNMP-based management systems.

For LLDP to provide expected benefits, all network devices that support LLDP must be consistent in their LLDP configuration.

Enabling or Disabling LLDP

The switch can be configured to transmit or receive LLDP information on a port-by-port basis.

LLDP is enabled by default on the switch. When LLDP is enabled on the switch, the ports transmit and receive LLDP information.

To enable or disable the transmission of LLDP information, use the following command (for this example ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# [no] lldp transmit
```

To enable or disable the reception of LLDP information, use the following command (for this example ethernet interface 1/12 is used):

```
Switch(config)# interface ethernet 1/12
Switch(config-if)# [no] lldp receive
```

To view the LLDP transmit and receive status, use the following commands:

```
Switch# display lldp interface all
```

LLDP Transmit Features

Numerous LLDP transmit options are available, including scheduled and minimum transmit interval, expiration on remote systems, SNMP trap notification, and the types of information permitted to be shared.

Scheduled Interval

The switch can be configured to transmit LLDP information to neighboring devices once each 5 to 32768 seconds. The scheduled interval is global; the same interval value applies to all LLDP transmit-enabled ports. However, to help balance LLDP transmissions and keep them from being sent simultaneously on all ports, each port maintains its own interval clock, based on its own initialization or reset time. This allows switch-wide LLDP transmissions to be spread out over time, though individual ports comply with the configured interval.

The transmit interval represents the number of seconds between two consecutive LLDP transmissions. To configure the global transmit interval (in seconds), use the following command:

```
Switch(config)# lldp timer <5-32768>
```

The default is 30 seconds. To reset the transmit interval to its default value, use the following command:

```
Switch(config)# no lldp timer
```

Minimum Interval

In addition to sending LLDP information at scheduled intervals, LLDP information is also sent when the switch detects relevant changes to its configuration or status (such as when ports are enabled or disabled). To prevent the switch from sending multiple LLDP packets in rapid succession when port status is in flux, a transmit delay timer can be configured.

The transmit delay timer represents the minimum time permitted between successive LLDP transmissions on a port. Any interval-driven or change-driven updates will be consolidated until the configured transmit delay expires.

To configure the minimum transmit interval (in seconds), use the following command:

```
Switch(config)# lldp transmit-delay <1-8192>
```

where the range is from 1 second to one-quarter of the scheduled transmit interval (**lldp timer** *<value>*), up to a maximum of 8192 seconds.

The default is 2 seconds. To reset the minimum transmit interval to its default value, use the following command:

```
Switch(config)# no lldp transmit-delay
```

Time-to-Live for Transmitted Information

The transmitted LLDP information is held by remote systems for a limited time. A time-to-live parameter allows the switch to determine how long the transmitted data is held before it expires. The hold time is configured as a multiple of the configured transmission interval.

```
Switch(config)# lldp holdtime-multiplier <2-10>
```

The default value is 4, meaning that remote systems will hold the port's LLDP information for four times the 30 second transmit interval/scheduled interval (msgtxint) value, or 120 seconds, before removing it from their MIB.

Note: If the transmission interval has a different value from the default value, the resulting timeout value will be different.

To reset the hold time multiplier to its default value, use the following command:

```
Switch(config)# no lldp holdtime-multiplier
```

Trap Notifications

If SNMP is enabled on the switch (see "Using Simple Network Management Protocol" on page 50), each port can be configured to send SNMP trap notifications whenever an update occurs in the LLDP database. By default, trap notification is disabled for each port. The trap notification state can be changed using the following commands:

```
Switch(config)# interface ethernet 1/1
Switch(config-if)# [no] lldp trap-notification
Switch(config-if)# exit
```

LLDP information is also sent when neighbors to the switch are added, removed, or changed. To prevent the switch from sending multiple trap notifications in rapid succession when port status is in flux, a global trap delay timer can be configured.

The trap delay timer represents the minimum time permitted between successive trap notifications on any port. Any interval-driven or change-driven trap notices from the port will be consolidated until the configured trap delay expires.

To configure the minimum trap notification interval (in seconds), use the following command:

```
Switch(config)# lldp trap-interval <5-3600>
```

The default is 5 seconds. To reset the minimum trap notification interval to its default value, use the following command:

```
Switch(config)# no lldp trap-interval
```

If SNMP trap notification is enabled, the notification messages can also appear in the system log. This is enabled by default. To change whether the SNMP trap notifications for LLDP events appear in the system log, use the following command:

```
Switch(config)# [no] logging level lldp <logging level>
```

where *logging level* is one of the following:

Value	Meaning
0	emergencies only
1	alerts
2	critical
3	errors
4	warnings
5	notifications
6	information
7	debugging

Changing the LLDP Transmit State

When the port is disabled, or when LLDP transmit is turned off for the port (see "LLDP Transmit Features" on page 576), a final LLDP packet is transmitted with a time-to-live value of 0. Neighbors that receive this packet will remove the LLDP information associated with the switch port from their MIB.

In addition, if LLDP is fully disabled on a port and then later re-enabled, the switch will temporarily delay resuming LLDP transmissions on the port to allow the port LLDP information to stabilize.

To globally configure the reinitialization delay interval (in seconds), use the following command:

```
Switch(config)# lldp reinit <1-10>
```

The default is 2 seconds. To reset the reinitialization delay interval to its default value, use the following command:

```
Switch(config)# no lldp reinit
```

Types of Information Transmitted

When LLDP transmission is permitted on the port (see "Enabling or Disabling LLDP" on page 575), the port advertises the following required information in type/length/value (TLV) format:

- Chassis ID

- Port ID

- LLDP Time-to-Live

LLDP transmissions can also be configured to enable or disable inclusion of optional information, using the following command (Interface Ethernet Mode):

```
Switch(config)# interface ethernet 1/1
Switch(config-if)# [no] lldp tlv-select <type>
Switch(config-if)# exit
```

where *type* is one of the following values:

Value	Definition
link-aggregation	Link Aggregation TLV
mac-phy-status	MAC/PHY Configuration/Status TLV
management-address	Management Address TLV
max-frame-size	Maximum frame size TLV
port-description	Port description TLV
port-protocol-vlan	Port and Protocol VLAN ID TLV
port-vlan	Port VLAN ID TLV
power-mdi	Power via MDI TLV
protocol-identity	Protocol identity TLV
system-capabilities	System capabilities TLV
system-description	System description TLV
system-name	System name TLV
vid-management	Version ID Management TLV
vlan-name	VLAN Name TLV

LLDP Receive Features

The following LLDP receive features are supported by CNOS.

Types of Information Received

When the LLDP receive option is enabled on a port (see), the port may receive the following TLV information from LLDP-capable remote systems:

- Chassis ID

- Port ID

- LLDP Time-to-Live

- Link Aggregation

- MAC/PHY Configuration/Status

- Management Address

- Maximum Frame Size

- Port Description

- Port and Protocol VLAN ID

- Port VLAN ID

- Power via MDI

- Protocol Identity

- System Capabilities

- System Description

- System Name

- VID Management

- VLAN Name

The switch stores the collected LLDP information in the MIB. Each remote LLDP-capable device is responsible for transmitting regular LLDP updates. If the received updates contain LLDP information changes (to port state, configuration, LLDP MIB structures, deletion), the switch will set a change flag within the MIB for convenient notification to SNMP-based management systems.

Time-to-Live for Received Information

Each remote device LLDP packet includes an expiration time. If the switch port does not receive an LLDP update from the remote device before the time-to-live timer expires, the switch will consider the remote information to be invalid, and will remove all associated information from the MIB.

Remote devices can also intentionally set their LLDP time-to-live to 0, indicating to the switch that the LLDP information is invalid and must be immediately removed.

Viewing Remote Device Information

LLDP information collected from neighboring systems can be viewed in numerous ways:

- Using a centrally-connected LLDP analysis server
- Using an SNMP agent to examine the switch MIB
- Using commands on the switch

The following command displays LLDP neighbor information:

```
Switch# display lldp neighbors
```

For more information on LLDP commands, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6.*

Note: Received LLDP information can change very quickly. When using display commands, it is possible that flags for some expected events may be too short-lived to be observed in the output.

To view detailed information of all remote devices, use the following command:

```
Switch# display lldp neighbors detail

Chassis ID MAC Address: a897.dcde.530d
Port ID Locally Assigned: Ethernet1/11
Local Port ID: Ethernet1/11
Port Description: Ethernet1/11
System Name: LENOVO G8272
System Description: LENOVO RackSwitch G8272, LENOVO NOS version 10.1.2.0
Time remaining: 106 seconds
System Capabilities: B, R
Enabled Capabilities: B, R
Management Address: 192.168.50.50
VLAN ID: 10

Chassis ID MAC Address: a897.dcde.530f
Port ID Locally Assigned: Ethernet1/13
Local Port ID: Ethernet1/13
Port Description: Ethernet1/13
System Name: LENOVO G8272
System Description: LENOVO RackSwitch G8272, LENOVO NOS version 10.1.2.0
Time remaining: 106 seconds
System Capabilities: B, R
Enabled Capabilities: B, R
Management Address: 192.168.50.50
VLAN ID: 10

Chassis ID MAC Address: a897.dcde.5311
Port ID Locally Assigned: Ethernet1/15
Local Port ID: Ethernet1/15
Port Description: Ethernet1/15
System Name: LENOVO G8272
System Description: LENOVO RackSwitch G8272, LENOVO NOS version 10.1.2.0
Time remaining: 106 seconds
System Capabilities: B, R
Enabled Capabilities: B, R
Management Address: 192.168.50.50
VLAN ID: 10

Total entries displayed: 3
```

Debugging LLDP

LLDP debugging is disabled by default. To enable or disable LLDP debugging, enter:

```
Switch(config)# [no] dbg lldp <type>
```

where *type* is one of the following:

Parameter	Description
all	Enable all debugging flags.
decode	Enable decode debugging.
encode	Enable encode debugging.
error	Enable error debugging.
event	Enable event debugging.
interface all interface ethernet *<chassis number/port number>*	Enable debugging on all interfaces or on the specified interface.
message	Enable Network Security Monitoring (NSM) message debugging.
rx	Enable reception debugging.
trace	Enable trace debugging.
tx	Enable transmission debugging.

The following types of debugging are available:

- all

  Enables all debug flags.

- event

  Messages appear when the following events occur:
  - Timer expiration
  - Admin status changes
  - TLV selection changes
  - Timer configuration changes
  - Transmit initialization
  - Receive initialization

- interface

  Debugging occurs on all physical interfaces or on the specified interface.

- message

  Prints messages between Network Security Monitoring (NSM) and the Linux LLDP process.

- rx

  Prints the packet contents when reception occurs.

- trace

  Prints the process stage of LLDP packets. For example:
  - Encode/Decode management address
  - TLV/TX packet enters delay timer
  - Begin transmission
  - Begin reception

  When an error is received on the LLDP PDU, the packet content is logged to the internal debug buffer.

- tx

  Prints the packet contents when transmission occurs.

- Error

  Prints a message when an error occurs.

To view LLDP debugging information, enter:

```
Switch# display dbg lldp
```

LLDP Example Configuration

The following is an example LLDP configuration.

1. Set the global LLDP timer features.

   a. Configure the transmit interval:

   ```
   Switch(config)# lldp timer 8
   ```

   b. Configure the minimum transmit interval:

   ```
   Switch(config)# lldp transmit-delay 2
   ```

   c. Configure the hold time multiplier:

   ```
   Switch(config)# lldp holdtime-multiplier 4
   ```

   d. Configure the reinitialization interval:

   ```
   Switch(config)# lldp reinit 2
   ```

   e. Configure the minimum trap notification interval:

   ```
   Switch(config)# lldp trap-notification-interval 5
   ```

2. Set LLDP options for each port.

   a. Enter Interface Configuration mode for each ethernet port:

   ```
   Switch(config)# interface ethernet 1/12
   ```

   b. Enable LLDP transmission and reception:

   ```
   Switch(config-if)# lldp receive
   Switch(config-if)# lldp transmit
   ```

   c. Enable SNMP trap notifications:

   ```
   Switch(config-if)# lldp trap-notification
   ```

   d. Configure the port to include port description in its outgoing LLDP messages:

   ```
   Switch(config-if)# lldp tlv-select port-description
   Switch(config-if)# exit
   ```

3. Enable syslog reporting.

   ```
   Switch(config)# logging level lldp 6
   ```

4. View remote device information.

```
Switch# display lldp neighbors

Capability codes:
  (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
  (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID          Local Intf    Hold-time Capability Port ID
MA_49              Ethernet1/14  120       BR         Ethernet1/14
G8052-71           mgmt0         120       BR         3

Total entries displayed: 2
```

```
Switch# display lldp timers

LLDP Timers:

Holdtime in seconds: 120
Reinit-time in seconds: 2
Transmit interval in seconds: 30
Transmit delay in seconds: 2
Trap interval in seconds: 5
```

```
Switch# display lldp interface ethernet 1/12

Interface Name: Ethernet1/12
--------------
Interface Information
 Enable (tx/rx/trap): Y/Y/N   Port Mac address: a8:97:dc:de:02:0e
```

Chapter 29. Service Location Protocol

The Service Location Protocol (SLP) is a service discovery protocol that allows the switch to find services across the local network without the need for an initial configuration. The switch discovers and selects network services without the requiring the hostname or the address of the service provider. You supply the type of service wanted and a set of attributes that describe it. Based on this information, SLP resolves the network address of the service provider.

Each type of service provided on the network has a unique service type string. Service types and their attributes are catalogued by the Internet Assigned Numbers Authority (IANA).

SLP defines specialized components called agents that perform tasks and support services as follows:

- User Agent (UA) - supports service query functions. It requests service information for user applications. The User Agent retrieves service information from the Service Agent. A Host On-Demand client is an example of a User Agent.
- Service Agent (SA) - provides service registration and service advertisement.

The switch behaves as a User Agent, which sends Service Request messages and Attribute Request messages.

When a Service Agent starts up, it locally registers all of its services and constructs a list of attributes for those services. Service Agents must accept both unicast and multicast SLP requests. This is done using the SLP reserved port 427.

Services Agents respond to Service and Attribute Request messages with Service and Attribute Reply messages. They also store the network addresses of up to a maximum of four User Agents. When a new User Agent is discovered, the oldest User Agent is deleted from the Service Agent.

SLP multicast messages use `239.255.255.253` as their destination address.

SLP Agents Communication

The User Agent sends multicast Service Request messages to the Service Agents present on the network on behalf of soliciting applications that run on the switch. For example, if a switch application needs access to a printer, the User Agent will send a Service Request across the network to find a printer for the switch application to use.

The Service Agents send in return unicast Service Reply messages with the location of all the network services that satisfy the requirements specified in the Service Request message.

Services are described by the configuration of attributes associated with a type of service. A User Agent can select an appropriate service by specifying the attributes that it needs in a Service Request message. When Service Reply messages are returned, they contain a Uniform Resource Locator (URL) pointing to the service desired and other information needed by the User Agent, such as server load.

SLP Specific Messages

SLP Agents use the following types of messages to communicate with each other:

- `Service Request` - request information about a specific type of service
- `Service Reply` - contains URLs for the type of service solicited in Service Request message, if there are any network services matching the specified requirements
- `Attribute Request` - request service attributes based on the service URL
- `Attribute Reply` - contains the attributes describing the solicited service

SLP Supported Service Attributes

SLP supports the following attributes:

- `level` - the level of the attribute (type: integer)
- `type` - the type of the device (type: string)
- `data-protocols` - the management protocols supported by the device (type: string)
- `serial-number` - the serial number of the device (type: string)
- `mac-address` - the MAC address of the device (type: string)
- `sysoid` - the System Object ID (type: string)
- `ipv4-enabled` - the status of the IPv4 accessibility of the device over the ethernet interface (type: boolean)
- `ipv4-address` - the IPv4 address of the device (type: string)
- `ipv6-enabled` - the status of the IPv6 accessibility of the device over the ethernet interface (type: boolean)
- `ipv6-address` - the IPv6 address of the device (type: string)
- `deviceName` - the custom name of the device or its hostname, if they are configured on the device (type: string)

SLP Configuration

By default, SLP is enabled on the switch.

To enable or disable SLP, use the following command:

```
Switch(config)# [no] ip slp enable
```

To view SLP information, use the following command:

```
Switch> display ip slp information
 Protocol Version: 2
 SLP State: enabled
 SLP Listening Port: 427
 SLP listening on interface: 11, IP address: 3.0.0.1
 SLP listening on interface: 3, IP address: 10.241.39.122
 SLP listening on interface: 10, IP address: 10.0.0.1
```

To view User Agent information, use the following command:

```
Switch> display ip slp user-agents
List of UAs:
  IP Address: 10.0.0.7 on port Ethernet1/11, updated 00:05:33 seconds ago
```

To view SLP statistics, use the following command:

```
Switch> display ip slp counters
SLP Send Counters       :      unicast      multicast
      SLP Da Adverts    :         0             0
      SLP Service Requests  :     0             0
      SLP Service Replies   :     0             0
      SLP Service Ack       :     0             0
      SLP Attribute Requests :    0             0
      SLP Attribute Replies  :    0             0
      SLP SrvType Requests   :    0             0
      SLP Service Replies   :     0             0
      SLP Srv Registrations  :    0             0
      SLP Srv Deregistrations:    0             0
      SLP SA Adverts    :         0             0
      SLP Unknown       :         0             0

SLP Receive Counters    :      unicast      multicast
      SLP Da Adverts    :         0             0
      SLP Service Requests  :   448           698
      SLP Service Replies   :     0             0
      SLP Service Ack       :     0             0
      SLP Attribute Requests :    0             0
      SLP Attribute Replies  :    0             0
      SLP SrvType Requests   :    0             0
      SLP Service Replies   :     0             0
      SLP Srv Registrations  :    0             0
      SLP Srv Deregistrations:    0             0
      SLP SA Adverts    :         0             0
      SLP Unknown       :         0             0

Scopes mismatch             :         0             0
Wrong destination           :         0             0
Invalid packets             :         0             0
```

Chapter 30. Simple Network Management Protocol

Lenovo Cloud Network Operating System provides Simple Network Management Protocol (SNMP) version 1, version 2, and version 3 support for access through any network management software, such as IBM Director or HP-OpenView.

Note: SNMP read and write functions are enabled by default. For best security practices, if SNMP is not needed for your network, it is recommended that you disable these functions prior to connecting the switch to the network.

The following topics are discussed in this section:

SNMP Versions

The switch can be accessed using SNMP version 1, version 2, or version 3:

- SNMPv1 is the initial implementation of the SNMP protocol. It operates over protocols such as User Datagram Protocol (UDP) and Internet Protocol (IP).

- SNMPv2c revised version 1 and includes improvements in performance, security, confidentiality, and communication.

- SNMPv3 adds cryptographic encryption to version 2. It also uses new textual conventions, concepts, and terminology.

SNMP Version 1 & Version 2

To access the SNMP agent on the switch, the read and write community strings on the SNMP manager must be configured to match those on the switch. By default, there are no default community strings configured on the switch.

The read and write community strings on the switch can be changed using the following commands on the CLI:

```
Switch(config)# snmp-server community <1-32 characters> [ro|rw|group <word>|
|view <view name> version {v1|v2c} {ro|rw}]
```

Note: The option ro provides read-only access. The option rw applies read-write access with the specified community string.

The SNMP manager must be able to reach the management interface or any one of the IP interfaces on the switch.

SNMP Version 3

SNMP version 3 (SNMPv3) is an enhanced version of the Simple Network Management Protocol, approved by the Internet Engineering Steering Group in March, 2002. SNMPv3 contains additional security and authentication features that provide data origin authentication, data integrity checks, timeliness indicators and encryption to protect against threats such as masquerade, modification of information, message stream modification and disclosure.

SNMPv3 allows clients to query the MIBs securely.

Note: Only SNMPv3 is enabled by default. To enable all SMTP versions, enter:

```
Switch(config)# snmp-server version v1v2v3
```

For more information on SNMP MIBs, see "SNMP MIBs" on page 597. For more details about the commands used to configure SNMP on the switch, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

SNMP Protocol Details

SNMP works by sending request and responses between a SNMP manager and a SNMP agent. Following are the option that can be configure to be used by switch when communicating with other devices using SNMP.

SNMP Notifications

Cloud NOS generates SNMP notifications as either traps or informs.

For the SNMP manager to receive the SNMP traps sent out by the SNMP agent on the switch, configure the trap host on the switch with the following commands:

```
Switch(config)# snmp-server enable traps [bfd|link [linkDown|linkUp]]
Switch(config)# snmp-server host <IPv4 or IPv6 address> [traps] version {1|2c|3
{auth|noauth|priv}} {<community string>|<SNMPv3 user name>} [udp-port <1-65535>]
```

In the case of informs, the SNMP manager that receives an inform message will acknowledge the message with an SNMP response PDU. If the sender does not receive a response for an inform message, the inform can be sent again. To configure informs, use the following command:

```
Switch(config)# snmp-server host <IPv4 or IPv6 address> informs version {2c|3
{auth|noauth|priv}} {<community string>|<SNMPv3 user name>} [udp-port <1-65535>]
```

By default, SNMP notifications are sent through all of the Virtual Routing and Forwarding (VRF) instances of the switch. To change the VRF used for SNMP notifications, use the following command to select between the management VRF, the default VRF, or both:

```
Switch(config)# snmp-server trap vrf {all|default|management}
```

Note: Only SNMPv3 is enabled by default. To enable all SMTP versions, enter:

```
Switch(config)# snmp-server version v1v2v3
```

SNMP Device Contact and Location

You can assign the device contact information and device location by using the following commands:

```
Switch(config)# snmp-server contact <contact name>
Switch(config)# snmp-server location <location name>
```

One-Time Authentication for SNMP over TCP

You can enable a one-time authentication for SNMP over a TCP session by using the following command:

```
Switch(config)# snmp-server tcp-session
```

Default Configuration

Up to 16 SNMP users can be configured on the switch. To modify an SNMP user, enter the following commands:

```
Switch(config)# snmp-server user <5-32 characters>
```

Users can be configured to use the authentication/privacy options. The switch supports:

- two authentication algorithms:
  - o MD5

```
Switch(config)# snmp-server user <username> auth md5 <password>
```

  - o SHA

```
Switch(config)# snmp-server user <username> auth sha <password>
```

- two privacy protocols:
  - o DES (CBC-DES Symmetric Encryption Protocol)

```
Switch(config)# snmp-server user <username> auth {md5|sha} <password> priv
des <privilege password>
```

  - o AES (AES-128 Advanced Encryption Standard)

```
Switch(config)# snmp-server user <username> auth {md5|sha} <password> priv
aes <privilege password>
```

Configuration Examples

Following are examples of the SNMP operations supported by Lenovo CNOS.

Basic SNMP Configuration Example

1. Enable SNMP on the switch.

```
Switch(config)# snmp-server enable snmp
```

2. Enable the switch to use all versions of SNMP.

```
Switch(config)# snmp-server version v1v2v3
```

3. (Optional) Set an SNMP view with the name all, OID Tree OID-TREE and type included.

```
Switch(config)# snmp-server view all OID-TREE included
```

4. Configure the community parameters.

```
Switch(config)# snmp-server community public group network-operator
Switch(config)# snmp-server community private group network-admin
```

5. View the SNMP configuration.

```
Switch(config)# display running-config snmp
```

User Configuration Example

1. Enable SNMP feature.

```
Switch(config)# snmp-server enable snmp
```

2. Set an SNMP view with the name all, OID Tree OID-TREE and type included.

```
Switch(config)# snmp-server view all OID-TREE included
```

3. Configure an SNMP user with authentication password password1 and privilege password password2.

```
Switch(config)# snmp-server user summer network-admin auth sha password1
priv des password2
```

4. View the SNMP user configuration.

```
Switch(config)# display snmp user
```

Configuring SNMP Trap Hosts

1. Enable SNMP traps.

```
Switch(config)# snmp-server enable traps
```

2. Send SNMPv3 traps messages to an SNMP recipient:

```
Switch(config)# snmp-server host 1.0.0.2 traps version 3 priv test
udp-port 162
```

3. View the SNMP trap configuration.

```
Switch(config)# display snmp trap
```

SNMP MIBs

The CNOS SNMP agent supports SNMP version 3. Security is provided through SNMP community strings. By default, there are no default community strings configured on the switch. The community string can be configured and modified only through the Industry Standard Command Line Interface (ISCLI). Detailed SNMP MIBs and trap definitions of the CNOS SNMP agent are contained in the CNOS enterprise MIB document.

The CNOS SNMP agent supports the following standard MIBs:

- rfc1213.mib
- rfc1573.mib
- rfc1643.mib
- rfc2011.mib
- rfc2012.mib
- rfc2013.mib
- rfc2037.mib
- rfc2233.mib
- rfc2790.mib
- rfc2863.mib
- rfc3411.mib
- rfc3412.mib
- rfc3413.mib
- rfc3414.mib
- rfc3415.mib
- rfc3418.mib
- rfc4363.mib
- lag-mib.mib
- lldp-mib.mib
- rfc4022.mib
- rfc4113.mib
- rfc4133.mib
- rfc4188.mib
- rfc4363.mib
- rfc4273.mib
- rfc4293.mib
- rfc4750.mib

The CNOS SNMP agent supports the following generic traps as defined in:

- RFC 1215:
  - linkDown
  - linkUp
  - egpNeighborLoss
- RFC 4188:
  - newRoot
  - topologyChange
- RFC 4273:
  - bgpEstablishedNotification
  - bgpBackwardTransNotification
- RFC 4750:
  - ospfVirtIfStateChange
  - ospfNbrStateChange
  - ospfVirtNbrStateChange
  - ospfIfConfigError
  - ospfVirtIfConfigError
  - ospfIfAuthFailure
  - ospfVirtIfAuthFailure
  - ospfIfRxBadPacket
  - ospfVirtIfRxBadPacket
  - ospfTxRetransmit
  - ospfVirtIfTxRetransmit
  - ospfOriginateLsa
  - ospfMaxAgeLsa
  - ospfLsdbOverflow
  - ospfLsdbApproachingOverflow
  - ospfIfStateChange
  - ospfNssaTranslatorStatusChange
  - ospfRestartStatusChange
  - ospfNbrRestartHelperStatusChange
  - ospfVirtNbrRestartHelperStatusChange

Chapter 31. Telemetry

Telemetry is an automated communications process in which measurements and other data are collected at remote points and transmitted to receiving equipment for monitoring. This data can then be used for various purposes.

This chapter discusses the following topics:

Network Telemetry Overview

Network telemetry is used by organizations to monitor their network devices, such as switches and routers, and provide this data to software controllers, which can analyze the data. In Lenovo Cloud Network Operating System, telemetry is used for pro-active congestion monitoring.

Telemetry lets you monitor networking devices continuously to detect potential congestion problems, ideally before they happen. This type of monitoring operation must address the following scenarios:

● Long-term congestion

Packets are dropped by the switch, such as by ports being used close to their line rate, or flows are back pressured for a long period of time due to the lack of buffer space.
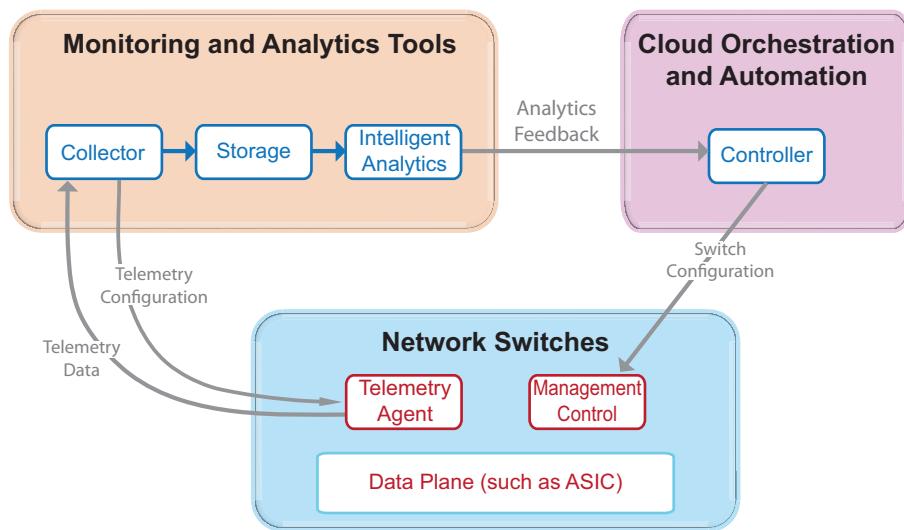
● Microbursts

Rapid bursts of packets at line rate can temporarily overflow switch buffers and cause packet loss or backpressure. These microbursts do not last long enough to be detected by traditional switch counters, such as SNMP or port statistics. In cloud environments, increased complexity and reduced visibility make detecting microbursts even more difficult.

CNOS Telemetry Architecture

In a complete cloud and data center ecosystem, the ultimate goal is to make sure applications can run in an efficient and reliable fashion. To accomplish this, applications use computing, storage, and network resources as accessories to this goal. Having end-to-end visible information about application performance becomes a critical aspect on modern networks. Telemetry provides real-time monitoring and reporting of how virtual and physical networks are being used. Figure 32 shows the key components of a typical telemetry solution.

Figure 32. Telemetry Ecosystem

In this figure, the **Collector** efficiently collets, normalizes, and transforms data. This data will be used to create different views and help solve various telemetry use cases.

This data is sent to a time-series resource indexing and metric **Storage** service, which provides a scalable means of storing short-term and long-term data.

Intelligent Analytics trigger actions based on defined rules against sampled or event data collected from the network.

The analytics are sent to the **Cloud Orchestration and Automation Controller**. The controller sends the best switch configuration to the Management Control agent on the switch.

The **Telemetry Configuration** tells the **Telemetry Agent** which subset of all supported data types must be sent to the **Collector**.

The **Telemetry Agent** sends the **Telemetry Data** back to the **Collector**, which sends the **Telemetry Configuration** back to the **Telemetry Agent** on the switch.

CNOS has a telemetry agent that is validated using the open source Ganglia monitoring application, which is used in this document as an example. Any application that supports the REST architecture and is capable of exchanging JSON messages over HTTP or HTTPS can be used to interact with the CNOS telemetry agent. The CNOS telemetry agent is built into CNOS and runs on the switch, whereas the telemetry controllers that interact with it run on external systems.

Any external application that uses a standard REST client can interact with the CNOS telemetry agent using the CNOS REST API. For more information about the REST functions supported by the CNOS telemetry agent, see the *CNOS REST API Programming Guide*.

The Ganglia Analytics Application

Note: The Ganglia analytics application is used in this document as an example of an analytics package. Any application that supports the REST architecture and is capable of exchanging JSON messages over HTTP or HTTPS can be used to interact with the CNOS telemetry agent.

Ganglia is an open source application that monitors and collects massive quantities of system metrics in near real time for large installations, such as computer networks where there are hundreds or thousands of nodes to be monitored. Its architecture is based on the separation between polling, storage, and data presentation operations.

The Ganglia software assumes monitored nodes can be organized meaningfully into groups. Ganglia refers to a group of hosts as a cluster, and it requires that at least one cluster of nodes exist. A cluster must have at least one node. The main purpose of grouping nodes into clusters is to achieve scalability by distributing the processing amongst the nodes instead of placing the burden on a centralized collection server.

The Ganglia software consists of the following three daemons and a tool:

- `gmond` (daemon)
- `gmetad` (daemon)
- `gweb` (daemon)
- `gmetric` (tool)

Each daemon is self-contained, needing only its own configuration file to operate. Each daemon will start and can run in the absence of the other two, but you need all three daemons to have a useful installation. Architecturally, the three daemons are cooperative. The Ganglia components interact with the CNOS telemetry agent (telemetryd).

The telemetry agent is logically equivalent to an embedded database that provides message broker services. It uses the abstraction layer to configure and retrieve telemetry information from the Application-Specific Integrated Circuit (ASIC).

The Ganglia Agent

The Ganglia Monitoring Daemon, `gmond`, is responsible for interacting with the monitored node to retrieve telemetry measurements, which are called metrics in Ganglia terminology. These metrics are the actual telemetry information, such as buffers, interfaces, and CPU statistics. In CNOS, `gmond` runs on an external server and interacts with the telemetry agent running on each CNOS switch.

The Central Data Aggregator

The Ganglia Meta Daemon, `gmetad`, polls separate `gmond` processes and stores the metric data to disk for later use. The result is a single meta-cluster data view in Round Robin Database (RRD) and XML formats.

The `gmetad` process stores older values in the RRD files and updates the metric data files, which consist of static allocations of values for various chunks of time.

The Data Visualization Front End

The Ganglia Web user interface, `gweb`, is a PHP program that serves as a front end for data visualization. It needs no preconfiguration, and it allows you to create your own graphs to explore and analyze data in a customized fashion. The `gweb` component is usually installed on the same physical hardware as `gmetad` because it needs access to the RRD databases created by `gmetad`.

The Ganglia Metric Tool

The Ganglia Metric tool, `gmetric`, lets you generate custom reports of the metrics from your own scripts in any language. These metrics are historic "streams" of data containing a sequence of "name, value" pairs with a time stamp associated with each pair, for example, (`cpu_usage, 90`%) at 10 AM today. The full historic sequence can provide data for a range such as the last minute, last five minutes, last ten minutes, last two hours, or last two weeks.

Using Ganglia with CNOS

The Lenovo Ganglia plugin helps the telemetry agent to integrate with Ganglia. It provides pull and push mode support to collect telemetry BST statistics, translates data to Ganglia metrics, and provides a visualization tool.

The Lenovo Ganglia Plugin works with Ganglia version 7.3.2. It does not add any scalability restrictions and it employs the distributed nature of Ganglia.

The CNOS Ganglia interface uses a Python script running on a server to communicate with the CNOS telemetry daemon on the switches using unicast HTTPS messages. The plugin sends HTTPS requests to the CNOS telemetry daemon, gets the JSON response from the switches, extracts the values, generates multiple metrics, and sends them out to `gmetad` with `gmetric`. This mode is used for both telemetry configuration and data collection.

You can download the Lenovo Ganglia plugin from:

https://github.com/lenovo/networking-telemetry

To install Ganglia to work with CNOS:

1. Make sure your system is running Ubuntu Linux 16.04 or the latest stable release.

2. Make sure the Ubuntu Advanced Packaging Tool (APT) system source is correctly configured:

```
% sudo apt-get update
```

This is necessary because during the installation, some dependency packages will be downloaded and installed automatically.

3. Install Ganglia:

Install with everything in one package:

```
% tar zxvf lenovo-ganglia-1.0.04.tar.gz
% cd lenovo-ganglia-1.0.04
% sudo ./install.sh install
```

Install with the Ganglia source code in separate packages:

```
% tar zxvf lenovo-ganglia-1.0.04.tar.gz
% cd lenovo-ganglia-1.0.04/ganglia-src
% wget
https://sourceforge.net/projects/ganglia/files/ganglia-web/3.7.2/ganglia-
web-3.7.2.tar.gz/download
% wget
https://sourceforge.net/projects/ganglia/files/ganglia%20monitoring%20cor
e/3.7.2/ganglia-3.7.2.tar.gz/download
% cd ..
% sudo ./install.sh install
```

4. Start the Ganglia Monitoring and Ganglia Meta Daemons:

```
% sudo service gmond start
% sudo service gmetad start
```

5. Make sure Ganglia is installed correctly by visiting:

   http://*<Ganglia Host IP Address>*/ganglia

   The default summary page, which includes the overall load, memory, and CPU status, will display.

6. Manually switch the IP list and interface list:

```
% lenovo-gangalia-1.0.04/pull/conf.py
```

7. Manually set the login username and password:

```
% lenovo-gangalia-1.0.04/pull/pull.py
```

8. Start the script to keep collection drop counters:

```
% cd lenovo-gangalia-1.0.04/pull/
% python pull.py
```

Types of Data Supplied by the CNOS Telemetry Agent

The following types of data are supplied by the CNOS telemetry agent.

Buffer Statistics

The following types of buffer statistics counters are available.

Congestion Drop Counters

Congestion drop counters keep track of packet loss caused by the lack of buffers to process the packets. The telemetry agent provides the following reports:

- top-drops

  Ports suffering maximum congestion in the switch and the associated drop counters.

- top-port-queue-drops

  Top port-queue level drop-counters in the switch. Each queue maps to a class of service.

- port-drops

  Per-port total drop counters.

- port-queue-drops

  Port-queue level drop-counters.

You can retrieve congestion drop counters from the agent periodically (periodic push) or immediately (on demand pull).

Buffer Utilization Counters

Buffer utilization counters are associated with different buffer types in the switch. Buffer utilization statistics are presented under the following corresponding categories, called *realms*:

- device, port, port + priority group, service pool, port + service pool

- ingress, egress

- unicast, multicast

You can retrieve buffer utilization counters periodically (periodic push), immediately (on demand pull), or via spontaneous asynchronous reports (threshold-driven push) from the agent.

Buffer Statistics Names

Buffer statistics names use the following conventions:

Prefix	Traffic Type
uc-	unicast traffic (for example, `uc-share-buffercount`)
mc-	multicast traffic (for example, `um-share-threshold`)
um-	both unicast and multicast traffic (for example, `mc-queue-entries`)

Table 42 provides a complete list of realms, indexes, thresholds, and statistics related to buffer utilization.

Table 42. *Buffer Utilization Realms, Indexes, Statistics, and Thresh holds*

Realm	Index # 1	Index # 2	Statistics	Thresholds
`device`			`data`	`threshold`
`ingress-port-priority-group`	*interface* (such as `Ethernet1/7`)	`priority-group`	`um-share-buffer-count`	`um-share-threshold`
`ingress-port-service-pool`	*interface* (such as `Ethernet1/7`)	`service-pool`	`um-share-buffer-count`	`um-share-threshold`
`ingress-service-pool`	`service-pool`		`um-share-buffer-count`	`um-share-threshold`
`egress-port-service-pool`	*interface* (such as `Ethernet1/7`)		`uc-share-buffer-count,` `um-share-buffer-count,` `mc-share-buffer-count,`	`uc-share-threshold,` `um-share-threshold`
`egress-service-pool`	`service-pool`		`um-share-buffercount,` `mc-share-buffer-count`	`um-share-threshold`
`egress-rqe-queue`	`queue`		`rqe-buffer-count`	`rqe-threshold`
`egress-cpu-queue`	`queue`		`cpu-buffer-count`	`cpu-threshold`
`egress-uc-queue`	`queue`		`uc-buffer-count`	`uc-threshold`
`egress-mc-queue`	`queue`		`mc-buffer-count`	`mc-threshold`

Note: A threshold value must be greater than zero.

Realm Parameters and Indexes

Realm parameters are necessary to access buffer statistics. These parameters act as indexes for the given buffer.

For example, to access a buffer in the `egress-uc-queue` realm, the realm parameter is the corresponding queue ID. Similarly, to access a buffer in the ingress-port-priority-group realm, the parameters are the interface name and the priority-group ID. The `device` realm has no indexes. No realm has more than two indexes.

Each buffer, given its parameters, may have one or more statistic. For example, the `ingress-port-service-pool` realm buffer, for a given port and service pool ID, has a single statistic called `um-share-buffer-count`. The `egress-port-service-pool buffer`, for a given port and service pool ID, offers three statistics:

- `uc-share-buffer-count`
- `um-share-buffer-count`
- `mc-share-buffer-count`

Table 43 and Table 44 show the ranges for some of the parameters used for indexes, thresholds, and statistics.

Table 43. *Telemetry Index Parameter Maximums*

Parameter	Description	Resources per ASIC		
		NE1032 NE1032T NE1072T	G8272 G8296 G8332	NE10032 NE2572
`queue`	Unicast Queues	16,384	2,960	2,600
`queue`	Multicast Queues	720	1,040	2,600
`queue-group`	Unicast Queue Groups	4,096	128	128
`service-pool`	Common Service Pools	1	1	1
`egress-cpu-queue`	ASIC CPU COS Queues	40	40	40
`egress-rqe-queue`	RQE	11	11	11

Table 44. *Telemetry Index Parameter Ranges*

Parameter	Description	Resources per ASIC		
		NE1032 NE1032T NE1072T	G8272 G8296 G8332	NE10032 NE2572
`service-pool`	Service Pool ID - Non-CEE Mode (default)	0	0	0
	Service Pool ID - CEE Mode	0-1	0-1	0
`priority-group`	Priority Group ID - Non-CEE Mode (default)	7	7	7
	Priority Group ID - CEE Mode	0-7	0-7	0-7

Note: Unless otherwise stated, when a number is used to identify a particular parameter, it is zero-indexed. For example, if a platform supports 720 multicast queues, multicast queue identifiers will range from 0 to 719.

Setting Up the CNOS Telemetry Agent

The setup commands for the CNOS telemetry agent must be done on the CLI. Once this basic configuration is been performed on each switch, the entire operation of the telemetry feature is mostly driven by the REST API because one of the main goals of telemetry is to have a centralized controller automatically discovering and interacting with multiple switches.

Note: For more details about the telemetry REST functions, see the *Lenovo Network REST API Programming Guide For Lenovo Cloud Network Operating System*.

Enable the Telemetry Agent

The CNOS telemetry agent is enabled by default. To enable or disable the CNOS telemetry agent, enter:

```
Switch(config)# [no] feature telemetry
```

Configure the Telemetry Controller

To configure the telemetry controller, enter:

```
Switch(config)# telemetry controller ip <IP address> port <port>
[vrf {default|management}]
```

where:

Argument	Definition
IP address	Client IP address
port	Port number to be used between 1 and 65535
default	(Optional) Have the agent and client associate the default VRF with the client's IP address
management	(Optional) Have the agent and client associate the management VRF with the client's IP address

To configure the telemetry controller to use the HTTPS protocol, enter:

```
Switch(config)# telemetry controller https
```

To configure the telemetry controller user, enter:

```
Switch(config)# telemetry controller username <username> password>
[encrypted] <password>
```

where:

Argument	Definition
username	Telemetry controller username
`encrypted`	(Optional) Encrypt the password
password	Telemetry controller user password.

To remove a previously configured controller, enter:

```
Switch(config)# no telemetry controller
```

Set Up the Telemetry Heartbeat

The telemetry heartbeat is automatically enabled by default at 5 second intervals. To change this, enter:

```
Switch(config)# telemetry heartbeat enabled interval <interval>
```

where *interval* is the new heartbeat interval, in seconds.

To disable the transmission of heartbeats, enter:

```
Switch(config)# telemetry heartbeat disabled
```

Configuring Telemetry Agent Parameters

To get the telemetry agent to provide telemetry data to the controller or collector, you must specify which data you want and how this data will be transferred. This section discusses the types of data you can collect and how to obtain it.

Congestion Drop Counters

To retrieve congestion drop counters, follow these steps:

1. Enable the buffer statistics utilization feature using the REST (PUT) URI:

   http://<*agent-IP-address*>**:**<*agent-port*>**/nos/api/cfg/telemetry/bst/feature**

   where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol.
bst-enable	Set to 1 to enable BST, 0 to disable it. Enabling BST allows the switch to track buffer utilization statistics.
send-async-reports	Set to 1 to enable the transmission of periodic asynchronous reports, 0 to disable this feature.
collection-interval	The collection interval, in seconds. This defines how frequently periodic reports will be sent to the configured controller.
trigger-rate-limit	The trigger rate limit, which defines the maximum number of threshold-driven triggered reports that the agent is allowed to send to the controller per trigger-rate-limit-interval; an integer from 1-5.
trigger-rate-limit-interval	The trigger rate limit interval, in seconds; an integer from 10-60.

Element	Description
send-snapshot-on-trigger	Set to 1 to enable sending a complete snapshot of all buffer statistics counters when a trigger happens, 0 to disable this feature.
async-full-report	Set to 1 to enable the async full report feature, 0 to disable it. When this feature is enabled, the agent sends full reports containing data related to all counters. When the feature is disabled, the agent sends incremental reports containing only the counters that have changed since the last report.

In the following example, setting the bst-enable field to 1 enables the buffer statistics and tracking feature on the switch. All other fields must be set to zero for this use case except the trigger-rate-limit, which has a range of 1-5, and the trigger-rate-limit-interval, which has a range from 10-60.

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: PUT
Header: Name: Content-type, Value: application/json
Request Body:
{
    "bst-enable" : 1,
    "send-async-reports" : 0,
    "collection-interval" : 0,
    "trigger-rate-limit" : 1,
    "trigger-rate-limit-interval" : 10,
    "send-snapshot-on-trigger" : 0,
    "async-full-report" : 0
}
```

2. Verify the buffer statistics utilization configuration via the CLI or via REST.

   ● CLI:

   ```
   Switch> display telemetry bst-feature
   ```

   ● REST (GET):

   ```
   http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/feature
   ```

   where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol

CLI example:

```
Switch> display telemetry bst-feature
BST Enabled             : enabled
Send Async Reports      : disabled
Collection Interval     : 0 seconds
Stats in Percentage     : enabled
Stat Units in Cells     : disabled
Trigger Rate Limit      : 1
Trigger Rate Limit Interval : 10 seconds
Send Snapshot on Trigger : disabled
Async Full Reports      : disabled
```

REST example:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: GET
Response Body:
{"stat-units-in-cells": 0, "stats-in-percentage": 1,
"collection-interval": 0, "send-async-reports": 0,
"send-snapshot-on-trigger": 0, "trigger-rate-limit": 1,
"async-full-report": 0, "trigger-rate-limit-interval": 10,
"bst-enable": 1}
```

3. Retrieve the ports with the top congestion drop counters on the switch:

- REST (POST):

```
http://<agent-IP-address>:<agent-port>/nos/api/info/telemetry/bst/congestion-dr
op-counters
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
req-id	The request ID; an integer

Element	Description
request-type	Request type; one of the following: • top-drops: Show ports with maximum congestion on the switch and their drop-counters • top-port-queue-drops: Show top port-queue level drop-counters on the switch • port-queue-drops: Show per port-queue level drop-counters on the switch • port-drops: Show per-port total drop counters on the switch

request-params	Request parameters dependent upon the request-type. For all congestion ports or port queues, the following parameters are valid.
	
	For top-drops, the following parameters are valid:
	
	• count: Number of ports required in the report. The ports are sorted with the port suffering maximum congestion at the top; an integer.
	• queue-type: Filters the report on the queue type; one of the following strings:
	  – ucast: Unicast queues
	  – mcast: Multicast queues
	  – all: All supported queues
	• interface-list: Comma-separated list of ports for the congestion drop counter report; an array. A value of all requests all the ports.
	• queue-list: An array of queue numbers to be considered for the drop report.
	• collection-interval: (Optional) The period in which the counters are collected from ASIC; An integer from 1-60. Default value: 0.

• CLI:

```
Switch> display telemetry bst-congestion top-drops [count <count>]
```

where *count* (optional) is the number of ports to include in the output. The default value is 64.

The following example retrieves the top three ports experiencing maximum congestion and their associated drop counters.

REST example:

```
URL: http://10.240.177.153:8090/nos/api/info/telemetry/bst/congestion-
drop-counters
Method: POST
Header: Name: Content-type, Value: application/json
Request Body:
{
    "req-id" : 2,
    "request-type" : "top-drops",
    "request-params": { "count":4 }
}

Response Body:
{
    "time-stamp": "2017-11-20 - 12:56:13 ",
    "report-type": "top-drops",
    "congestion-ctr": [
        {
            "interface": "Ethernet1/38",
            "ctr": "3212203451"
        },
        {
            "interface": "Ethernet1/40",
            "ctr": "2855833510"
        },
        {
            "interface": "Ethernet1/47",
            "ctr": "2855767786"
        },
        {
            "interface": "Ethernet1/18",
            "ctr": "2855598983"
        }
    ]
}
```

CLI example:

```
Switch>#  display telemetry bst-congestion top-drops count 4
Timestamp                             : 2017-11-20 - 13:00:00
Request identifier                    : 10001
Top ports with congestion drops       : 4
------------------------------------
interface                 counter
------------------------------------
Ethernet1/38            3212203451
Ethernet1/40            2855833510
Ethernet1/47            2855767786
Ethernet1/18            2855598983
```

4. Retrieve the port, queue pairs with top congestion drop counters on the switch:

- REST (POST):

```
http://<agent-IP-address>:<agent-port>/nos/api/info/telemetry/bst/congestion-dr
op-counters
```

where:

Argument	Definition
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
req-id	The request ID; an integer
request-type	One of the following: ● top-drops: Show ports with maximum congestion on the switch and their drop-counters ● top-port-queue-drops: Show top port-queue level drop-counters on the switch ● port-queue-drops: Show per port-queue level drop-counters on the switch ● port-drops: Show per-port total drop counters on the switch

request-params	Request parameters; one of the following strings::
	
	● qcount: Number of ports required in the report. The ports are sorted with the port suffering maximum congestion at the top; an integer.
	● queue-type: Filters the report on the queue type; one of the following strings:
	  – ucast: Unicast queues
	  – mcast: Multicast queues
	  – all: All supported queues
	● interface-list: Comma-separated list of ports for the congestion drop counter report; an array. A value of all requests all the ports.
	● queue-list: An array of queue numbers to be considered for the drop report.
	● collection-interval: (Optional) The period in which the counters are collected from ASIC; An integer from 1-60. Default value: 0.

- CLI:

```
Switch> display telemetry bst-congestion top-port-queue-drops [count
<count>]
```

where:

Parameter	Description
count	(Optional) The number of port, queue pairs to include in the output. The default value is 64.
queue-type	(Optional) Queue type to be included. One of: ● all: Unicast and multicast traffic queues (default) ● mcast: Multicast traffic queues ● ucast: Unicast traffic queues

The following example retrieves the top three port, queue pairs experiencing maximum congestion and their associated drop counters for unicast traffic:

```
URL: http://10.240.177.153:8090/nos/api/info/telemetry/bst/congestion-
drop-counters
Method: POST
Header: Name: Content-type, Value: application/json
Request Body:
Request body:
{
"req-id" : 2,
"request-type" : "top-drops",
"request-params":
  {
      "count":4
    }
}
Response Body:
{
    "time-stamp": "2017-11-20 - 12:56:13 ",
    "report-type": "top-drops",
    "congestion-ctr": [
        {
            "interface": "Ethernet1/38",
            "ctr": "484033646"
        },
        {
            "interface": "Ethernet1/40",
            "ctr": "429830595"
        },
        {
            "interface": "Ethernet1/47",
            "ctr": "429817996"
        },
        {
            "interface": "Ethernet1/18",
            "ctr": "429794738"
        }
    ]
}
```

5. Retrieve the per-port total congestion drop counters for a specified set of interfaces:

- REST (POST):

```
http://<agent-IP-address>:<agent-port>/nos/api/info/telemetry/bst/congestion-
drop-counters
```

where:

Argument	Definition
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
req-id	The request ID; an integer
request-type	One of the following: ● top-drops: Show ports with maximum congestion on the switch and their drop-counters ● top-port-queue-drops: Show top port-queue level drop-counters on the switch ● port-queue-drops: Show per port-queue level drop-counters on the switch ● port-drops: Show per-port total drop counters on the switch
request-params	Request parameters; one of the following strings:: ● qcount: Number of ports required in the report. The ports are sorted with the port suffering maximum congestion at the top; an integer. ● queue-type: Filters the report on the queue type; one of the following strings: – ucast: Unicast queues – mcast: Multicast queues – all: All supported queues ● interface-list: Comma-separated list of ports for the congestion drop counter report; an array. A value of all requests all the ports. ● queue-list: An array of queue numbers to be considered for the drop report. ● collection-interval: (Optional) The period in which the counters are collected from ASIC; An integer from 1-60. Default value: 0.

- CLI:

```
Switch> display telemetry bst-congestion port-drops [interface
ethernet <slot>-<chassis>]
```

where *slot-chassis* is the slot and chassis of a specific ethernet interface.

The following example retrieves the per-port drop counters.

REST example:

```
URL: http://10.240.177.153:8090/nos/api/info/telemetry/bst/congestion-
drop-counters
Method: POST
Header: Name: Content-type, Value: application/json
Request Body:
Request body:
{
 "req-id" : 10,
 "request-type" : "port-drops",
 "request-params":
{
     "interface-list": ["Ethernet1/9","Ethernet1/10"]
}

Response Body:
{
    "time-stamp": "2017-11-20 - 13:01:47 ",
    "report-type": "port-drops",
    "congestion-ctr": [
        {
            "interface": "Ethernet1/9",
            "ctr": "0"
        },
        {
            "interface": "Ethernet1/10",
            "ctr": "0"
        },
    ]
}
```

CLI example:

```
Switch> display telemetry bst-congestion port-drops interface ethernet
1/9-10
0-46
Timestamp                                  : 2017-11-20 - 13:02:36
Request identifier                         : 10001
Ports reported                             : 2
------------------------------------
interface                 counter
------------------------------------
Ethernet1/9               5092169130
Ethernet1/10                       0
```

Note that not setting the field `collection-interval` instructs the telemetry
agent to provide the requested drop counter information immediately. The
`interface-list` provides an explicit set of interfaces for which we want to
obtain the counters. In the Response Body of this example, the number of packets
dropped on interface Ethernet1/1 was 1778, whereas there were no packet drops on
the other two interfaces given.

6. Retrieve the total congestion drop counters per port, queue pair for a specified set of interfaces and queues:

- REST (POST):

```
http://<agent-IP-address>:<agent-port>/nos/api/info/telemetry/bst/congestion-dr
op-counters
```

where:

Argument	Definition
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
req-id	The request ID; an integer
request-type	One of the following: • top-drops: Show ports with maximum congestion on the switch and their drop-counters • top-port-queue-drops: Show top port-queue level drop-counters on the switch • port-queue-drops: Show per port-queue level drop-counters on the switch • port-drops: Show per-port total drop counters on the switch

request-params	Request parameters; one of the following strings::
	
	• qcount: Number of ports required in the report. The ports are sorted with the port suffering maximum congestion at the top; an integer.
	• queue-type: Filters the report on the queue type; one of the following strings:
	    – ucast: Unicast queues
	    – mcast: Multicast queues
	    – all: All supported queues
	• interface-list: Comma-separated list of ports for the congestion drop counter report; an array. A value of all requests all the ports.
	• queue-list: An array of queue numbers to be considered for the drop report.
	• collection-interval: (Optional) The period in which the counters are collected from ASIC; An integer from 1-60. Default value: 0.

● CLI:

```
Switch> display telemetry bst-congestion port-queue-drops [interface
ethernet <slot>-<chassis>] [queue <queue number>] [queue-type <queue type>]
```

where:

Parameter	Description
count	(Optional) is the number of ports to include in the output. The default value is 64.
queue	(Optional) Queue number; an integer from 0-7. Default value: all queues.
queue-type	(Optional) Queue type to be included. One of: ● all: Unicast and multicast traffic queues (default) ● mcast: Multicast traffic queues ● ucast: Unicast traffic queues

The following example retrieves the per-port drop counters.

REST example:

```
URL: http://10.240.177.153:8090/nos/api/info/telemetry/bst/congestion-
drop-counters
Method: POST
Header: Name: Content-type, Value: application/json
Request Body:
{
"req-id" : 2,
"request-type" : "port-queue-drops",
"request-params" :
    {
        "interface-list" : ["Ethernet1/38","Ethernet1/39"],
        "queue-type" : "all",
        "queue-list" : [0,1]
    }
}

Response Body:

    "time-stamp": "2017-11-20 - 13:04:08 ",
    "report-type": "port-queue-drops",
    "congestion-ctr": [
        {
            "interface": "Ethernet1/38",
            "queue-type": "ucast",
            "queue-drop-ctr": [
                [0,"0"],
                [1,"0"]
            ]
        },
        {
            "interface": "Ethernet1/38",
            "queue-type": "mcast",
            "queue-drop-ctr": [
                [0,"6985575610"],
                [1,"0"]
            ]
        },
        {
            "interface": "Ethernet1/39",
            "queue-type": "ucast",
            "queue-drop-ctr": [
                [0,"0"],
                [1,"0"]
            ]
        },
        {
            "interface": "Ethernet1/39",
            "queue-type": "mcast",
            "queue-drop-ctr": [
                [0,"5491324486"],
                [1,"0"]
            ]
        },
    ]
}
```

Note that not setting the field `collection-interval` instructs the telemetry agent to provide the requested drop counter information immediately. The `interface-list` provides an explicit set of interfaces for which we want to obtain the counters. The `queue-type` specifies `all`, which includes unicast and multicast queues.

In the Response Body of this example, the number of packets dropped on interface Ethernet1/38 was 6985575610, the number of packets dropped on interface Ethernet 1/39 was 5491324486, all of them were multicast packets, and they were dropped in queue 0.

CLI example:

```
Switch> display telemetry bst-congestion port-queue-drops interface
ethernet 1/38-39 queue 0
Timestamp                              : 2017-11-20 - 13:05:14
Request identifier                     : 10001
(port, queue) entries reported         : 4
-----------------------------------------------------------
interface       queue     queue-type            counter
-----------------------------------------------------------
Ethernet1/38    0         ucast                        0
Ethernet1/38    0         mcast                6985575610
Ethernet1/39    0         ucast                        0
Ethernet1/39    0         mcast                5491324486
```

In the output of this example, the number of packets dropped on interface Ethernet1/38 was 6985575610, the number of packets dropped on interface Ethernet 1/39 was 5491324486, all of them were multicast packets, and they were dropped in queue 0.

BST Buffer Counters

To retrieve BST buffer counters, follow these steps:

1. Enable the buffer statistics utilization feature using the REST (PUT) URI:

   http://_<agent-IP-address>_**:**_<agent-port>_**/nos/api/cfg/telemetry/bst/feature**

   where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol.
`bst-enable`	Set to 1 to enable BST, 0 to disable it. Enabling BST allows the switch to track buffer utilization statistics.
`send-async-reports`	Set to 1 to enable the transmission of periodic asynchronous reports, 0 to disable this feature.

Element	Description
`collection-interval`	The collection interval, in seconds. This defines how frequently periodic reports will be sent to the configured controller.
`trigger-rate-limit`	The trigger rate limit, which defines the maximum number of threshold-driven triggered reports that the agent is allowed to send to the controller per `trigger-rate-limit-interval`; an integer from 1-5.
`trigger-rate-limit-interval`	The trigger rate limit interval, in seconds; an integer from 10-60.
`send-snapshot-on-trigger`	Set to 1 to enable sending a complete snapshot of all buffer statistics counters when a trigger happens, 0 to disable this feature.
`async-full-report`	Set to 1 to enable the async full report feature, 0 to disable it. When this feature is enabled, the agent sends full reports containing data related to all counters. When the feature is disabled, the agent sends incremental reports containing only the counters that have changed since the last report.

In the following example, setting the `bst-enable` field to 1 enables the buffer statistics and tracking feature on the switch. All other fields must be set to zero for this use case except the trigger-rate-limit, which has a range of 1-5, and the trigger-rate-limit-interval, which has a range from 10-60.

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: PUT
Header: Name: Content-type, Value: application/json
Request Body:
{
    "bst-enable" : 1,
    "send-async-reports" : 0,
    "collection-interval" : 0,
    "trigger-rate-limit" : 1,
    "trigger-rate-limit-interval" : 10,
    "send-snapshot-on-trigger" : 0,
    "async-full-report" : 0
}
```

2. Verify the buffer statistics utilization configuration via the CLI or via REST.

- CLI:

```
Switch> display telemetry bst-feature
```

- REST (GET):

```
http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/feature
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol

CLI example:

```
Switch> display telemetry bst-feature
BST Enabled                 : enabled
Send Async Reports          : disabled
Collection Interval         : 0 seconds
Stats in Percentage         : enabled
Stat Units in Cells         : disabled
Trigger Rate Limit          : 1
Trigger Rate Limit Interval : 10 seconds
Send Snapshot on Trigger    : disabled
Async Full Reports          : disabled
```

REST example:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: GET
Response Body:
{"stat-units-in-cells": 0, "stats-in-percentage": 1,
"collection-interval": 0, "send-async-reports": 0,
"send-snapshot-on-trigger": 0, "trigger-rate-limit": 1,
"async-full-report": 0, "trigger-rate-limit-interval": 10, "bst-enable":
1}
```

3. Configure buffer statistics tracking using the REST (PUT) URI:

```
http://<agent-IP-address>:<agent-port>/nos/api/info/telemetry/bst/tracking
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
`track-peak-stats`	Track peak statistics; 1 to enable, 0 to disable.
`track-ingress-port-priority-group`	Track the ingress port priority group; 1 to enable, 0 to disable.
`track-ingress-port-service-pool`	Track the ingress port service pool; 1 to enable, 0 to disable.
`track-ingress-service-pool`	Track the ingress service pool; 1 to enable, 0 to disable.
`track-egress-port-service-pool`	Track the egress port service pool; 1 to enable, 0 to disable.
`track-egress-service-pool`	Track the egress service pool; 1 to enable, 0 to disable.
`track-egress-uc-queue`	Track the egress unicast queue; 1 to enable, 0 to disable.
`track-egress-mc-queue`	Track the egress multicast queue; 1 to enable, 0 to disable.
`track-egress-cpu-queue`	Track the egress CPU queue; 1 to enable, 0 to disable.
`track-egress-rqe-queue`	Track the egress RQE queue; 1 to enable, 0 to disable.
`track-device`	Track the device; 1 to enable, 0 to disable.

4. Verify the buffer statistics tracking configuration via the CLI or via REST.

- CLI:

```
Switch> display telemetry bst-tracking
```

- REST (GET):

```
http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/tracking
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol

CLI example:

```
Switch> display telemetry bst-tracking
Track Peak Stats                 : disabled
Track Device                     : enabled
Track Ingress Port Priority Group : enabled
Track Ingress Port Service Pool  : enabled
Track Ingress Service Pool       : enabled
Track Egress CPU Queue           : enabled
Track Egress MC Queue            : enabled
Track Egress Port Service Pool   : enabled
Track Egress RQE Queue           : enabled
Track Egress Service Pool        : enabled
Track Egress UC Queue            : enabled
```

REST example:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: GET
Response Body:
{"track-egress-port-service-pool": 1, "track-egress-rqe-queue": 1,
"track-ingress-port-service-pool": 1, "track-ingress-service-pool": 1,
"track-peak-stats": 1, "track-ingress-port-priority-group": 1,
"track-egress-service-pool": 1, "track-egress-uc-queue" : 1,
"track-egress-mc-queue" : 1, "track-egress-cpu-queue" : 1,
"track-device": 1}
```

5. Retrieve the buffer statistics report on demand.

- REST (POST):

```
http://<agent-IP-address>:<agent-port>/nos/api/info/telemetry/bst/report
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
`include-ingress-port-priority-group`	Retrieve the ingress port priority group; 1 to enable, 0 to disable.
`include-ingress-port-service-pool`	Retrieve the ingress port service pool; 1 to enable, 0 to disable.
`include-ingress-service-pool`	Retrieve the ingress service pool; 1 to enable, 0 to disable.
`include-egress-port-service-pool`	Retrieve the egress port service pool; 1 to enable, 0 to disable.
`include-egress-service-pool`	Retrieve the egress service pool; 1 to enable, 0 to disable.
`include-egress-rqe-queue`	Retrieve the egress RQE queue; 1 to enable, 0 to disable.
`include-device`	Retrieve the device; 1 to enable, 0 to disable.
`include-egress-uc-queue`	Retrieve the egress unicast queue; 1 to enable, 0 to disable.
`include-egress-mc-queue`	Retrieve the egress multicast queue; 1 to enable, 0 to disable.
`include-egress-cpu-queue`	Retrieve the egress CPU queue; 1 to enable, 0 to disable.

- CLI:

```
Switch> display telemetry bst-report <realm>
```

where *realm* is one of the following:

Realm	Description
`device`	Show BST statistics report for device realm
`egress-cpu-queue`	Show BST statistics report for the egress-cpu-queue realm
`egress-mc-queue`	Show BST statistics report for the egress-mc-queue realm
`egress-port-service-pool`	Show BST statistics report for the egress-port-service-pool realm
`egress-rqe-queue`	Show BST statistics report for the egress-rqe-queue realm
`egress-service-pool`	Show BST statistics report for the egress-service-pool realm
`egress-uc-queue`	Show BST statistics report for the egress-uc-queue realm
`ingress-port-priority-group`	Show BST statistics report for the ingress-port-priority-group realm
`ingress-port-service-pool`	Show BST statistics report for the ingress-port-service-pool realm
`ingress-service-pool`	Show BST statistics report for the ingress-service-pool realm

In the following example, the include-*xxx* fields in the request specify the counters we want to retrieve.

```
URL: http://10.240.177.153:8090/nos/api/info/telemetry/bst/report
Method: POST
Header: Name: Content-type, Value: application/json
Request Body:
{
    "include-ingress-port-priority-group" : 1,
    "include-ingress-port-service-pool" : 1,
    "include-ingress-service-pool" : 1,
    "include-egress-port-service-pool" : 1,
    "include-egress-service-pool" : 1,
    "include-egress-rqe-queue" : 1,
    "include-device" : 1
}

Response Body:
{"report": [{"realm": "device", "data": "37"}, {"realm":
"ingress-service-pool", "data": [[0, "47"]]},
{"realm": "ingress-port-service-pool", "data": [{"interface":
"Ethernet1/1", "data": [0, "42"]}, {"interface": "Ethernet1/47", "data":
[0, "43"]}]},
{"realm": "ingress-port-priority-group", "data": [{"interface":
"Ethernet1/1", "data": [7, "47", "0"]}, {"interface": "Ethernet1/47",
"data": [7, "47", "0"]}]},
{"realm": "egress-port-service-pool", "data": [{"interface":
"Ethernet1/48", "data": [0, "43", "0", "0", "0"]}]},
{"realm": "egress-service-pool", "data": [[0, "48", "0", "0"]]},
{"realm": "egress-rqe-queue", "data": [[0, "0"], [1, "0"], [2, "0"], [3,
"0"], [4, "0"], [5, "0"], [6, "0"], [7, "0"], [8, "0"], [9, "0"], [10,
"0"]]}],
"time-stamp": "2017-05-04 - 14:46:24 "}
```

6. Retrieve the buffer statistics threshold report on demand:

- REST (POST)

```
http://<agent-IP-address>:<agent-port>/nos/api/info/telemetry/bst/threshold
```

- CLI:

```
Switch> display telemetry bst-thresholds <realm>
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol

Element	Description
realm	One of the following: ● `include-ingress-port-priority-group` ● `include-ingress-port-service-pool` ● `include-ingress-service-pool` ● `include-egress-port-service-pool` ● `include-egress-service-pool` ● `include-egress-rqe-queue` ● `include-device` ● `include-egress-cpu-queue` ● `include-egress-uc-queue` ● `include-egress-mc-queue` If no realm is used, all realms are reported.
`include-ingress-port-priority-group`	Retrieve the ingress port priority group; 1 to enable, 0 to disable.
`include-ingress-port-service-pool`	Retrieve the ingress port service pool; 1 to enable, 0 to disable.
`include-ingress-service-pool`	Retrieve the ingress service pool; 1 to enable, 0 to disable.
`include-egress-port-service-pool`	Retrieve the egress port service pool; 1 to enable, 0 to disable.
`include-egress-service-pool`	Retrieve the egress service pool; 1 to enable, 0 to disable.
`include-egress-rqe-queue`	Retrieve the egress RQE queue; 1 to enable, 0 to disable.
`include-device`	Retrieve the device; 1 to enable, 0 to disable.
`include-egress-uc-queue`	Retrieve the egress unicast queue; 1 to enable, 0 to disable.
`include-egress-mc-queue`	Retrieve the egress multicast queue; 1 to enable, 0 to disable.
`include-egress-cpu-queue`	Retrieve the egress CPU queue; 1 to enable, 0 to disable.

In the following example, the response containing the buffer statistics thresholds on demand has been shortened because the actual response is several pages long.

```
URL: http://10.240.177.153:8090/nos/api/info/telemetry/bst/threshold
Method: POST
Header: Name: Content-type, Value: application/json
Request Body:
{
    "include-ingress-port-priority-group" : 1,
    "include-ingress-port-service-pool" : 1,
    "include-ingress-service-pool" : 1,
    "include-egress-port-service-pool" : 1,
    "include-egress-service-pool" : 1,
    "include-egress-rqe-queue" : 1,
    "include-device" : 1
}

Response Body:
{"report": [{"realm": "device", "data": "100"}, {"realm":
"ingress-service-pool", "data": [[0, "100"]]}, {"realm":
"ingress-port-service-pool", "data": [{"interface": "Ethernet1/1",
"data": [0, "100"]}, {"interface": "Ethernet1/2", "data": [0, "100"]},
...
{"interface": "Ethernet1/32", "data": [0, "100", "100", "0", "0"]}]},
{"realm": "egress-service-pool", "data": [[0, "100", "100", "0"]]},
{"realm": "egress-rqe-queue", "data": [[0, "100"], [1, "50"], [2, "100"],
[3, "100"], [4, "100"], [5, "100"], [6, "100"], [7, "100"], [8, "100"],
[9, "100"], [10, "100"]]}], "time-stamp": "2017-05-04 - 16:45:52 "}
```

Detect Congestion After it Happens

To detect congestion after it has happened, follow these steps:

1. Configure an external controller using the following CLI command:

```
Switch> telemetry controller ip <IP address> port <port>
[vrf {default|management}]
```

where:

Argument	Definition
IP address	Client IP address
port	Port number to be used between 1 and 65535
default	(Optional) Have the agent and client associate the default VRF with the client's IP address
management	(Optional) Have the agent and client associate the management VRF with the client's IP address

For example, to allow the telemetry agent to send heartbeat messages to the controller at 10.240.177.235 via port 80 to the VRF management port, enabling to the controller to automatically discover the properties of the agent running on CNOS, enter:

```
Switch> telemetry controller ip 10.240.177.235 port 80 vrf management
```

2. Verify the controller configuration using the CLI command:

```
Switch> display telemetry information
```

For example:

```
Switch> display telemetry information
Telemetry admin status  : enabled
Telemetry oper status   : up
Heartbeat status        : disabled
Controllers configured  : 1
Controller 0:
    IP address          : 10.240.177.235
    TCP port            : 80
    VRF                 : management
    Protocol            : HTTP
```

3. Enable the buffer statistics utilization feature using the following REST (PUT) URI:

```
http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/feature
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
bst-enable	Set to 1 to enable the BST feature, 0 to disable this feature
send-async-reports	Set to 1 to send async reports, 0 to disable this feature
trigger-rate-limit	Trigger rate limit; an integer from 1-5
trigger-rate-limit-interval	Trigger rate limit interval, in seconds; an integer from 10-60
send-snapshot-on-trigger	Set to 1 to send a snapshot when a trigger happens, 0 to disable this feature
async-full-report	Set to 1 to send a full async report, 0 to disable this feature.

For example, to enable this feature on host 10.240.17.153 on port 8090, use:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: PUT
Header: Name: Content-type, Value: application/json
Request Body:
{
    "bst-enable" : 1,
    "send-async-reports" : 0,
    "collection-interval" : 0,
    "trigger-rate-limit" : 1,
    "trigger-rate-limit-interval" : 10,
    "send-snapshot-on-trigger" : 0,
    "async-full-report" : 0
}
```

4. Verify the buffer statistics utilization configuration via the CLI or via REST.

- CLI:

```
Switch> display telemetry bst-feature
```

CLI example:

```
Switch> display telemetry bst-feature
BST Enabled                 : enabled
Send Async Reports          : disabled
Collection Interval         : 0 seconds
Stats in Percentage         : enabled
Stat Units in Cells         : disabled
Trigger Rate Limit          : 1
Trigger Rate Limit Interval : 10 seconds
Send Snapshot on Trigger    : disabled
Async Full Reports          : disabled
```

- REST (GET):

```
http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/feature
```

where:

Element	Description
`agent-IP-address`	The IP address of the telemetry agent
`agent-port`	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol

REST example:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: GET
Response Body:
{"stat-units-in-cells": 0, "stats-in-percentage": 1,
"collection-interval": 0, "send-async-reports": 0,
"send-snapshot-on-trigger": 0, "trigger-rate-limit": 1,
"async-full-report": 0, "trigger-rate-limit-interval": 10,
"bst-enable": 1}
```

5. Configure the periodic retrieval of congestion drop counters for ports with maximum drop counters using the following REST (POST) URI:

```
http://<agent-IP-address>:<agent-port>/nos/api/info/telemetry/bst/congestion-drop-
counters
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
`req-id`	The request ID; an integer

Element	Description
request-type	Request type; one of the following: ● `top-drops`: Show ports with maximum congestion on the switch and their drop-counters ● `top-port-queue-drops`: Show top port-queue level drop-counters on the switch ● `port-queue-drops`: Show per port-queue level drop-counters on the switch ● `port-drops`: Show per-port total drop counters on the switch
request-params	Request parameters dependent upon the `request-type`. For all congestion ports or port queues, the following parameters are valid. For `top-drops`, the following parameters are valid: ● `count`: Number of ports required in the report. The ports are sorted with the port suffering maximum congestion at the top; an integer. ● `queue-type`: Filters the report on the queue type; one of the following strings: 　– `ucast`: Unicast queues 　– `mcast`: Multicast queues 　– `all`: All supported queues ● `interface-list`: Comma-separated list of ports for the congestion drop counter report; an array. A value of all requests all the ports. ● `queue-list`: An array of queue numbers to be considered for the drop report. ● `collection-interval`: (Optional) The period in which the counters are collected from ASIC; An integer from 1-60. Default value: 0.

The following example requests reports to be sent every 300 seconds (five minutes) from the telemetry agent including up to eight ports experiencing maximum congestion and their associated drop counters to the controller whose IP address and port number have been previously configured:

```
URL: http://10.240.177.153:8090/nos/api/info/telemetry/bst/congestion-dro
p-counters
Method: POST
Header: Name: Content-type, Value: application/json
Body:
{
    "req-id": 1,
    "request-type": "top-drops",
    "request-params": {
        "count": 8
    },
    "collection-interval": 300
}
```

6. Configure the periodic retrieval of congestion drop counters for the queues (per port) with maximum drop counters using the same REST (POST) URI but with a `request-type` of `top-port-queue-drops`.

REST example

```
URL: http://10.240.177.153:8090/nos/api/info/telemetry/bst/congestion-dro
p-counters
Method: POST
Header: Name: Content-type, Value: application/json
Body:
{
"req-id" : 1,
"request-type" : "top-port-queue-drops",
"request-params":
 {
      "count":2 ,
      "queue-type": "all"  <<< can be ucast/mcast/ucast
 }
}

Response body:
{
    "time-stamp": "2017-11-20 - 12:58:58 ",
    "report-type": "top-port-queue-drops",
    "congestion-ctr": [
        {
            "interface": "Ethernet1/38",
            "queue-type": "mcast",
            "queue-drop-ctr": [
                [
                    0,
                    "2472802159"
                ]
            ]
        },
        {
            "interface": "Ethernet1/47",
            "queue-type": "mcast",
            "queue-drop-ctr": [
                [
                    0,
                    "2198113613"
                ]
            ]
        }
    ]
}
```

CLI example:

```
Switch> display telemetry bst-congestion top-drops count 2
Timestamp                             : 2017-11-20 - 13:00:00
Request identifier                    : 10001
Top ports with congestion drops       : 2
------------------------------------
interface                  counter
------------------------------------
Ethernet1/38          2472802159
Ethernet1/47          2198113613
```

7. Configure the Ganglia analytics application and collector.

For information on how to configure Ganglia, see:

http://ganglia.info/?tag=documentation

Predicting Congestion Before it Happens

Follow these steps to use the telemetry agent to predict where and when congestion might occur.

1. Configure an external controller using the following CLI command:

```
Switch> telemetry controller ip <IP address> port <port>
[vrf {default|management}]
```

where:

Argument	Definition
IP address	Client IP address
port	Port number to be used between 1 and 65535
default	(Optional) Have the agent and client associate the default VRF with the client's IP address
management	(Optional) Have the agent and client associate the management VRF with the client's IP address

For example, to allow the telemetry agent to send heartbeat messages to the controller at 10.240.177.235 via port 80 to the VRF management port, enabling to the controller to automatically discover the properties of the agent running on CNOS, enter:

```
Switch> telemetry controller ip 10.240.177.235 port 80 vrf management
```

2. Verify the controller configuration using the CLI command:

```
Switch> display telemetry information
```

For example:

```
Switch> display telemetry information
Telemetry admin status   : enabled
Telemetry oper status    : up
Heartbeat status         : disabled
Controllers configured   : 1
Controller 0:
    IP address           : 10.240.177.235
    TCP port             : 80
    VRF                  : management
    Protocol             : HTTP
```

3. Enable the buffer statistics utilization feature using the following REST (PUT) URI:

```
http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/feature
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
`bst-enable`	Set to 1 to enable the BST feature, 0 to disable this feature
`send-async-reports`	Set to 1 to send async reports, 0 to disable this feature
`trigger-rate-limit`	Trigger rate limit; an integer from 1-5
`trigger-rate-limit-interval`	Trigger rate limit interval, in seconds; an integer from 10-60
`send-snapshot-on-trigger`	Set to 1 to send a snapshot when a trigger happens, 0 to disable this feature
`async-full-report`	Set to 1 to send a full async report, 0 to disable this feature.

For example, to enable this feature on host 10.240.17.153 on port 8090, use:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: PUT
Header: Name: Content-type, Value: application/json
Request Body:
{
    "bst-enable" : 1,
    "send-async-reports" : 0,
    "collection-interval" : 0,
    "trigger-rate-limit" : 1,
    "trigger-rate-limit-interval" : 10,
    "send-snapshot-on-trigger" : 0,
    "async-full-report" : 0
}
```

4. Verify the buffer statistics tracking configuration via the CLI or via REST.

● CLI:

```
Switch> display telemetry bst-tracking
```

CLI example:

```
Switch> display telemetry bst-tracking
Track Peak Stats                 : disabled
Track Device                     : enabled
Track Ingress Port Priority Group : enabled
Track Ingress Port Service Pool  : enabled
Track Ingress Service Pool       : enabled
Track Egress CPU Queue           : enabled
Track Egress MC Queue            : enabled
Track Egress Port Service Pool   : enabled
Track Egress RQE Queue           : enabled
Track Egress Service Pool        : enabled
Track Egress UC Queue            : enabled
```

● REST (GET):

```
http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/tracking
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol

REST example:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: GET
Response Body:
{"track-egress-port-service-pool": 1, "track-egress-rqe-queue": 1,
"track-ingress-port-service-pool": 1, "track-ingress-service-pool": 1,
"track-peak-stats": 1, "track-ingress-port-priority-group": 1,
"track-egress-service-pool": 1, "track-device": 1}
```

5. Configure buffer statistics tracking using the following REST (PUT) URI:

```
REST: http://<agent-ip-address>:<agent-port>/nos/api/cfg/telemetry/bst/tracking
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
`track-peak-stats`	Set to 1 to peak statistics tracking, 0 to disable this feature
`track-ingress-port-priority-group`	Set to 1 to enable ingress port priority group tracking, 0 to disable this feature
`track-ingress-port-service-pool`	Set to 1 to enable ingress port service pool tracking, 0 to disable this feature
`track-ingress-service-pool`	Set to 1 to enable ingress service pool tracking, 0 to disable this feature
`track-egress-port-service-pool`	Set to 1 to enable egress port service pool tracking, 0 to disable this feature
`track-egress-service-pool`	Set to 1 to enable egress service pool tracking, 0 to disable this feature
`track-egress-rqe-queue`	Set to 1 to enable egress RQE queue tracking, 0 to disable this feature
`track-device`	Set to 1 to enable tracking of this device, 0 to disable this feature
`track-egress-uc-queue`	Set to 1 to enable egress unicast queue, 0 to disable.
`track-egress-mc-queue`	Set to 1 to enable egress multicast queue , 0 to disable.
`itrack-egress-cpu-queue`	Set to 1 to enable egress CPU queue, 0 to disable.

The following example uses the element "track-peak-stats" : 1 to tell the underlying switching ASIC to track the peak statistics so we do not miss microbursts. The other elements track all types of buffer statistic counters:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/tracking
Method: PUT
Header: Name: Content-type, Value: application/json
Request Body:
{
   "track-peak-stats" : 1,
   "track-ingress-port-priority-group" : 1,
   "track-ingress-port-service-pool" : 1,
   "track-ingress-service-pool" : 1,
   "track-egress-port-service-pool" : 1,
   "track-egress-service-pool" : 1,
   "track-egress-rqe-queue" : 1,
   "track-device" : 1
   "track-egress-uc-queue" :1,
   "track-egress-mc-queue" :1,
   "track-egress-cpu-queue" :1
}
```

6. Verify the buffer statistics tracking configuration.

- CLI:

```
Switch> display telemetry bst-tracking
```

CLI example:

```
Switch> display telemetry bst-tracking
Track Peak Stats                 : disabled
Track Device                     : enabled
Track Ingress Port Priority Group : enabled
Track Ingress Port Service Pool  : enabled
Track Ingress Service Pool       : enabled
Track Egress CPU Queue           : enabled
Track Egress MC Queue            : enabled
Track Egress Port Service Pool   : enabled
Track Egress RQE Queue           : enabled
Track Egress Service Pool        : enabled
Track Egress UC Queue            : enabled
```

- REST (GET):

```
http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/tracking
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol

REST example:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: GET
Response Body:
{"track-egress-port-service-pool": 1, "track-egress-rqe-queue": 1,
"track-ingress-port-service-pool": 1, "track-ingress-service-pool": 1,
"track-peak-stats": 1, "track-ingress-port-priority-group": 1,
"track-egress-service-pool": 1, "track-device": 1}
```

7. Configure buffer statistics thresholds using the following REST (PUT) URI:

```
REST: http://<agent-ip-address>:<agent-port>/nos/api/cfg/telemetry/bst/threshold
```

with a JSON request body of:

```
Request Body:
{
    "realm": "<realm>",
    "<index1>": <index1 value>,
    "<index1>": <index1 value>,
    "<threshold>": <threshold value>,
    "<threshold1>": <threshold value>,
    "<threshold2>": <threshold value>,
    ...
    "<thresholdN>": <threshold value>,
}
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
realm	Any of the realms shown in Table 42.

Element	Description
index1	(Optional) The corresponding **Index1** keyword shown in Table 42. **Note:** All realms except device have at least one Index value.
index1 value	A valid value for the *index1* keyword.
index2	(Optional) The corresponding Index2 value shown in Table 42.
threshold, threshold1, threshold2,... thresholdN	The corresponding Threshold keyword shown in Table 42. There can be multiple thresholds
threshold value	A valid value for the *threshold* keyword.

The following examples configure a threshold for every realm and counter supported by the telemetry agent. In this example, the threshold is set to 50%. This means if the buffer utilization goes above 50%, the agent must send an asychronous report to the configured controller. This helps identify potential congestion issues before they occur. When the buffer utilization for a particular realm and buffer type reaches 100%, the switch will start dropping packets.

Example 1:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/threshold
Method: PUT
Header: Name: Content-type, Value: application/json
Request Body:
{
    "realm": "device",
    "threshold": 50
}
```

Example 2:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/threshold
Method: PUT
Header: Name: Content-type, Value: application/json
{
    "realm": "ingress-port-priority-group",
    "interface": "Ethernet1/1",
    "priority-group": 7,
    "um-share-threshold": 50
}
```

Example 3:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/threshold
Method: PUT
Header: Name: Content-type, Value: application/json
{
    "realm": "ingress-port-service-pool",
    "interface": "Ethernet1/1",
    "priority-group": 7,
    "um-share-threshold": 50
}
```

Example 4:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/threshold
Method: PUT
Header: Name: Content-type, Value: application/json
{
    "realm": "ingress-service-pool",
    "service-pool": 0,
    "um-share-threshold": 50
}
```

Example 5:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/threshold
Method: PUT
Header: Name: Content-type, Value: application/json
{
    "realm": "egress-port-service-pool",
    "interface": "Ethernet1/1",
    "service-pool": 0,
    "uc-share-threshold": 50,
    "um-share-threshold": 50,
    "mc-share-threshold": 50
}
```

Example 6:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/threshold
Method: PUT
Header: Name: Content-type, Value: application/json
{
    "realm": "egress-service-pool",
    "service-pool": 0,
    "um-share-threshold": 50,
    "mc-share-threshold": 50
}
```

Example 7:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/threshold
Method: PUT
Header: Name: Content-type, Value: application/json
{
    "realm": "egress-rqe-queue",
    "queue": 1,
    "rqe-threshold": 50
}
```

Example 8:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/threshold
Method: PUT
Header: Name : Content-type, Value: application/json
 {
    "realm": "egress-cpu-queue",
    "queue": 0,
    "cpu-threshold": 10
 }
```

Example 9:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/threshold
Method: PUT
Header: Name: Content-type, Value: application/json
{
    "realm": "egress-uc-queue",
    "queue": 50,
    "uc-threshold": 10
}
```

Example 9:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/threshold
Method: PUT
Header: Name: Content-type, Value: application/json
{
    "realm": "egress-mc-queue",
    "queue": 1000,
    "uc-threshold": 10
}
```

A few notes about these examples:

- The priority-group is 7 because this is the only value supported while the switch runs in non-CEE mode (default mode).

- The service-pool is 0 because this is the only value supported while the switch runs in non-CEE mode (default mode).

- The queue is 1 because this value falls within the valid range for the egress-rqe-queue realm.

- To make this example useful, we have to configure the thresholds for the entire valid range of parameters because we do not know, in advance, which interfaces, service pools, queues, and so forth may experience high utilization.

8. Configure the Ganglia analytics application and collector.

   For information on how to configure Ganglia, see:

   http://ganglia.info/?tag=documentation

Capacity Planning Based on Trend Analysis

To plan for future network capacity based on long-term, detailed trend analysis, follow these steps.

1. Configure an external controller using the following CLI command:

```
Switch> telemetry controller ip <IP address> port <port>
[vrf {default|management}]
```

where:

Argument	Definition
IP address	Client IP address
port	Port number to be used between 1 and 65535
default	(Optional) Have the agent and client associate the default VRF with the client's IP address
management	(Optional) Have the agent and client associate the management VRF with the client's IP address

For example, to allow the telemetry agent to send heartbeat messages to the controller at 10.240.177.235 via port 80 to the VRF management port, enabling to the controller to automatically discover the properties of the agent running on CNOS, enter:

```
Switch> telemetry controller ip 10.240.177.235 port 80 vrf management
```

2. Verify the controller configuration using the CLI command:

```
Switch> display telemetry information
```

For example:

```
Switch> display telemetry information
Telemetry admin status  : enabled
Telemetry oper status   : up
Heartbeat status        : disabled
Controllers configured  : 1
Controller 0:
    IP address          : 10.240.177.235
    TCP port            : 80
    VRF                 : management
    Protocol            : HTTP
```

3. Enable the buffer statistics utilization feature using the following REST (PUT) URI:

```
http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/feature
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
bst-enable	Set to 1 to enable the BST feature, 0 to disable this feature
send-async-reports	Set to 1 to send async reports, 0 to disable this feature
trigger-rate-limit	Trigger rate limit; an integer from 1-5
trigger-rate-limit-interval	Trigger rate limit interval, in seconds; an integer from 10-60
send-snapshot-on-trigger	Set to 1 to send a snapshot when a trigger happens, 0 to disable this feature
async-full-report	Set to 1 to send a full async report, 0 to disable this feature.

In this example, enabling bst_enable enables buffer statistics and tracking on the switch, and enabling send-async-reports tells the telemetry agent to send period asynchoronous reports. Setting the value of collection-interval to 60 means these reports will be sent every 60 seconds.

Disabling async-full-report means the reports will contain only information about about the counters with values that changed since the last report was sent. All other fields must be set to zero for this use case, except for trigger-rate-limit and trigger-rate-limit-interval.

4. Verify the buffer statistics tracking configuration.

- CLI:

```
Switch> display telemetry bst-tracking
```

CLI example:

```
Switch> display telemetry bst-feature
BST Enabled                 : enabled
Send Async Reports          : enabled
Collection Interval         : 60 seconds
Stats in Percentage         : enabled
Stat Units in Cells         : disabled
Trigger Rate Limit          : 1
Trigger Rate Limit Interval : 10 seconds
Send Snapshot on Trigger    : disabled
Async Full Reports          : disabled
```

- REST (GET):

```
http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/tracking
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol

REST example:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: GET
Response Body:
{"stat-units-in-cells": 0, "stats-in-percentage": 1,
"collection-interval": 60, "send-async-reports": 1,
"send-snapshot-on-trigger": 0, "trigger-rate-limit": 1,
"async-full-report": 0, "trigger-rate-limit-interval": 10,
"bst-enable": 1}
```

5. Configure buffer statistics tracking using the following REST (PUT) URI:

```
REST: http://<agent-ip-address>:<agent-port>/nos/api/cfg/telemetry/bst/tracking
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol
track-peak-stats	Set to 1 to peak statistics tracking, 0 to disable this feature
track-ingress-port-priority-group	Set to 1 to enable ingress port priority group tracking, 0 to disable this feature
track-ingress-port-service-pool	Set to 1 to enable ingress port service pool tracking, 0 to disable this feature
track-ingress-service-pool	Set to 1 to enable ingress service pool tracking, 0 to disable this feature
track-egress-port-service-pool	Set to 1 to enable egress port service pool tracking, 0 to disable this feature
track-egress-service-pool	Set to 1 to enable egress service pool tracking, 0 to disable this feature
track-egress-rqe-queue	Set to 1 to enable egress RQE queue tracking, 0 to disable this feature
track-device	Set to 1 to enable tracking of this device, 0 to disable this feature
track-egress-uc-queue	Set to 1 to enable egress unicast queue tracking, 0 to disable tracking
track-egress-mc-queue	Set to 1 to enable egress multicast queue tracking, 0 to disable tracking
track-egress-cpu-queue	Set to 1 to enable egress CPU queue tracking, 0 to disable tracking

The following example uses the element "track-peak-stats" : 1 to tell the underlying switching ASIC to track the peak statistics so we do not miss microbursts. The other elements track all types of buffer statistic counters:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/tracking
Method: PUT
Header: Name: Content-type, Value: application/json
Request Body:
{
   "track-peak-stats" : 1,
   "track-ingress-port-priority-group" : 1,
   "track-ingress-port-service-pool" : 1,
   "track-ingress-service-pool" : 1,
   "track-egress-port-service-pool" : 1,
   "track-egress-service-pool" : 1,
   "track-egress-rqe-queue" : 1,
   "track-device" : 1,
   "track-egress-uc-queue" : 1,
   "track-egress-mc-queue" : 1,
   "track-egress-cpu-queue " : 1
}
```

6. Verify the buffer statistics tracking configuration.

- CLI:

```
Switch> display telemetry bst-tracking
```

CLI example:

```
Switch> display telemetry bst-tracking
Track Peak Stats                 : disabled
Track Device                     : enabled
Track Ingress Port Priority Group : enabled
Track Ingress Port Service Pool   : enabled
Track Ingress Service Pool       : enabled
Track Egress CPU Queue           : enabled
Track Egress MC Queue            : enabled
Track Egress Port Service Pool    : enabled
Track Egress RQE Queue           : enabled
Track Egress Service Pool        : enabled
Track Egress UC Queue            : enabled
```

- REST (GET):

```
http://<agent-IP-address>:<agent-port>/nos/api/cfg/telemetry/bst/tracking
```

where:

Element	Description
agent-IP-address	The IP address of the telemetry agent
agent-port	The port used by the telemetry agent. The agent-port is 8090 when the CNOS REST server operates using the HTTP protocol and 443 when the CNOS REST server operates using the HTTPS protocol

REST example:

```
URL: http://10.240.177.153:8090/nos/api/cfg/telemetry/bst/feature
Method: GET
Response Body:
{"track-egress-port-service-pool": 1, "track-egress-rqe-queue": 1,
"track-ingress-port-service-pool": 1, "track-ingress-service-pool": 1,
"track-peak-stats": 1, "track-ingress-port-priority-group": 1,
"track-egress-service-pool": 1, "track-device": 1}
```

7. Configure the Ganglia analytics application and collector.

For information on how to configure Ganglia, see:

http://ganglia.info/?tag=documentation

Part 7: Hyperconverged Infrastructure

A Hyperconverged Infrastructure provides an integrated compute, storage, and networking system that is easy to manage from end to end. Consolidating the management of these systems to a single tool greatly simplifies the overall system management and reduces the number of resources needed. In many cases, the virtual network domain is managed by Server Administrators, who may have limited network expertise.

This section discusses the following topics:

Chapter 32. Network Virtualization Gateway

Note: Network Virtualization Gateway is available on the following switches:

- Lenovo RackSwitch G8272
- Lenovo RackSwitch G8296
- Lenovo RackSwitch G8332

The following topics are covered in this chapter:

- "NSX Integration Concepts" on page 656
- "VXLAN" on page 661
- "Lenovo VXLAN Gateway" on page 663
- "VXLAN Gateway Standalone Topologies" on page 670
- "High Availability Support" on page 672
- "VXLAN Gateway Configuration Example" on page 674

NSX Integration Concepts

NSX is a VMware virtualized network platform that offers the operational model of a virtual machine over a network. Virtual networks function in a similar way to virtual machines for computers. VMware NSX builds virtual networks inside software, providing a full set of networking services, such as logical switching, routing, firewall, load balancing, VPN, quality of service (QoS), and monitoring.

Virtual networks are managed independently of the underlying type of network hardware. VMware NSX reproduces the entire network model in software, allowing any network topology (simple or complex) to be created and provisioned in a few seconds. Virtual networks can then be deployed over any existing network hardware without generating disruptions in functionality.

VMware NSX uses the Virtual Extensible LAN (VXLAN) protocol to provide network virtualization for cloud computing. VXLAN offers the same Ethernet Layer 2 services as the VLAN protocol, but with increased flexibility and scalability. The VXLAN protocol uses an overlay mechanism to tunnel virtualized network traffic over existing Layer 3 or Layer 2 networks. These logical networks must be programmed and managed throughout the network including virtual and physical servers, networking equipment, and storage devices.

The integration of VMware NSX and Lenovo switches offers the following benefits to deploying network virtualization within software defined cloud networking:

- Virtual and physical workloads can be connected upon demand on a common logical segment regardless of hypervisor, IP subnet, or physical location

- Holistic views of the virtual and physical topology increase operational efficiency

- Network virtualization with VMware NSX does not require IP multicast to learn or forward broadcast, unknown unicast, or multicast packets

- A single point of management and control (NSX API) can be used for configuring logical networks across hypervisors and physical access switches

Figure 33. VMware NSX Standalone Architecture
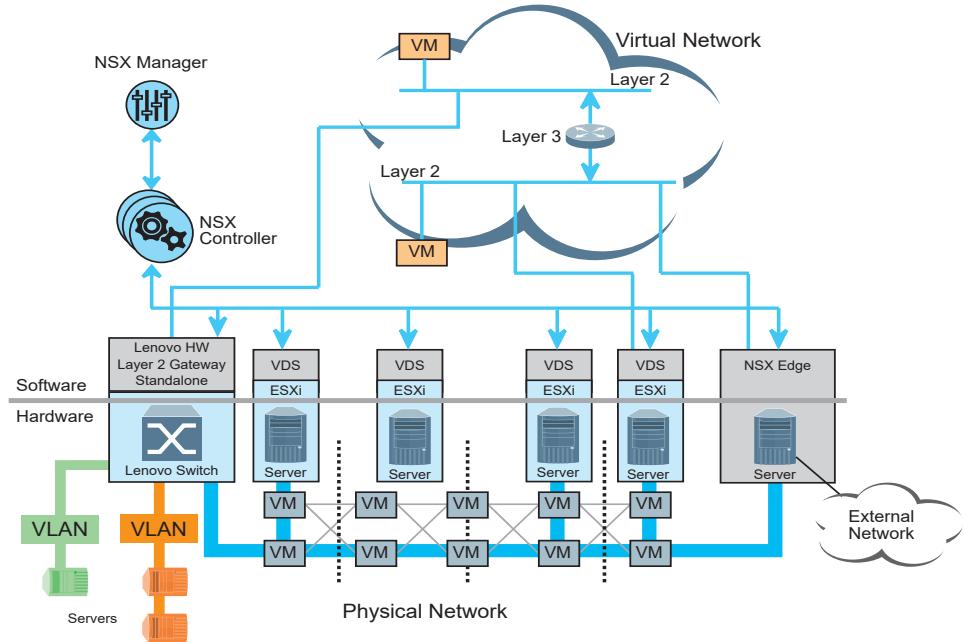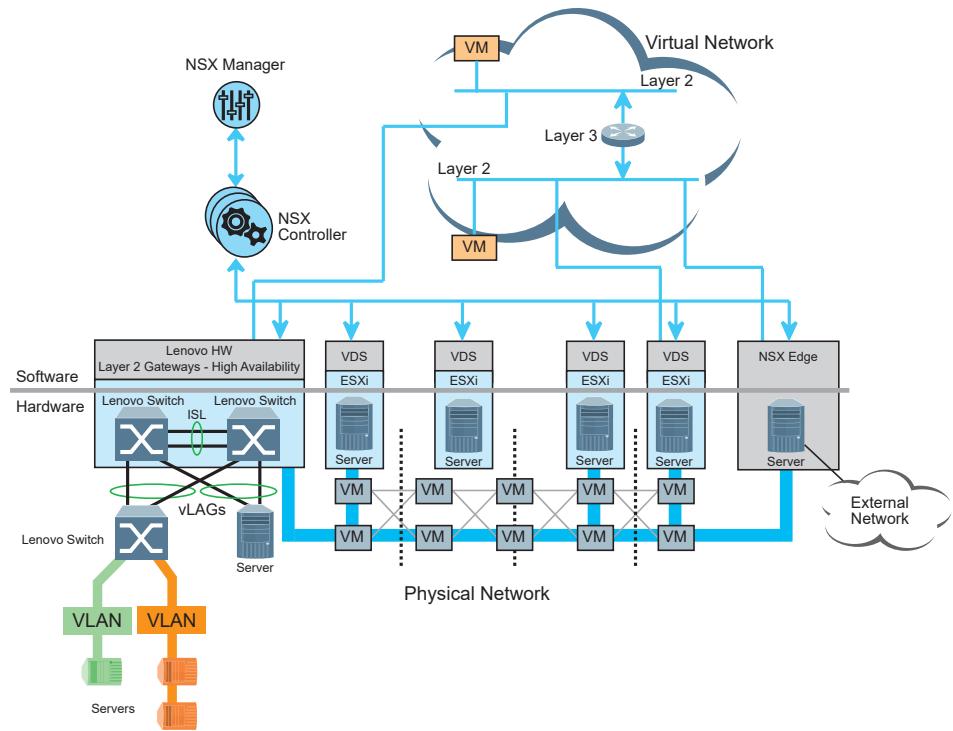
Figure 34. VMware NSX High Availability Architecture

VMware NSX Components

VMware NSX consists of the following components:

- NSX Manager
- NSX Controller
- NSX Edge
- NSX Virtual Switch (vSwitch)

NSX Manager

The NSX Manager is the centralized management component of VMware NSX. It provides a management interface for VMware NSX, through which virtual networks can then be configured.

The NSX Manager is capable of deploying the NSX Controller and NSX Edge, configuring the controller cluster, installing VXLAN, and more.

NSX Controller

The NSX Controller is a distributed state management system that controls virtual networks and overlay transport tunnels. It consists of a cluster of several virtual machines.

It is the central control point for all logical switches and it maintains information about all virtual machines, logical switches, hosts, and VXLANs.

NSX Edge

NSX Edge isolates a virtual network by providing network edge security and gateway services, such as dynamic routing, perimeter firewall, DHCP, NAT, and load balancing.

NSX vSwitch

A NSX vSwitch is the software that creates an abstraction between the servers hypervisors and the physical network. It allows the distribution of virtual workloads on any available infrastructure in the data center, regardless of the underlying physical network infrastructure.

NSX Tunneling

VMware NSX establishes logical connections called tunnels between specific physical devices or virtual machines on the virtual network. This allow the advantage of not needing to change the physical topology of a network. The network separation is achieved through the use of software, thus making it a logical separation and not a physical one.

For example, in a large cloud data center the physical network needs to be restructured to meet the needs of multiple clients. Rather than physically altering the network, VMware NSX provides the option of creating any virtual network topology and deploying it over the existing physical network.

To achieve this logical separation, VMware NSX creates tunnels between the physical or virtual devices needed in a specific topology. A tunnel is created using the VXLAN protocol and it originates and terminates in a VXLAN Tunnel End Point (VTEP).

A VTEP can be created on a VMware vSphere Hypervisor or on a physical switch. Once a tunnel is set up, it gives the impression that the devices (physical or virtual) connected to its VTEPs are communicating across a Layer 2 domain. The underlying Layer 3 infrastructure is invisible to the devices communicating through the tunnel.

A VTEP has an IP address. A tunnel can only have its endpoints (VTEPs) in different subnets. If two virtual machines running on different host devices communicate directly, then unicast traffic is transmitted between the two VTEPs without network flooding.

In some cases of Layer 2 broadcast, unknown unicast, and multicast traffic (BUM traffic) that originates on a virtual machine, packets may need to be sent to all other virtual machines that belong to the same virtual network. The virtual network can span multiple VMware vSphere Hypervisors. BUM traffic that originates on a virtual machine hosted on a single hypervisor may need to be replicated to remote hypervisors which host other virtual machines that are connected to the same virtual network.

When a virtual machine sends traffic to other virtual machines that have VTEPs located in the same subnet, it creates separate copies of the BUM packets and sends them directly to each virtual machine.

If BUM traffic is destined for virtual machines with VTEPs located in a different subnet, the originating virtual machine assigns one VTEP on each subnet as a replicator. The originating virtual machine does not send copies of the BUM packets to each virtual machine that are in the same subnet as the replicator. Instead it sends a single copy of each BUM packet to every replicator. When a replicator receives BUM traffic, it takes the role of creating separate copies of the BUM packets and sending them to each virtual machine on its subnet.

To view the replicators to which the Lenovo VXLAN Gateway is connected, enter:

```
Switch> display nwv vxlan tunnels

Tunnel count: 261

Tunnel IP Address     Tunnel Type     Status
-----------------     -----------     ------
100.1.34.56           Local           UP
192.168.2.246         SN(Active)      UP
192.168.2.241         SN(Backup)      UP
192.168.2.242         SN(Backup)      UP
192.168.2.244         SN(Backup)      UP
192.168.2.245         SN(Backup)      UP
192.168.2.247         SN(Backup)      UP
192.168.2.248         SN(Backup)      UP
192.168.2.250         SN(Backup)      UP
172.16.1.2            Remote          UP
192.168.1.1           Remote          UP
192.168.1.2           Remote          UP
192.168.1.3           Remote          UP
192.168.1.4           Remote          UP
...
```

Note: The replicators are displayed as being tunnel type SN (Service Node).

If there are multiple replicators configured, then only one of them is set as active, while the rest are set as backups. The Lenovo VXLAN Gateway sends the BUM traffic that was initiated from the connected servers only to the active replicator. If the active replicator fails, its status changes to down and one of the backup replicators takes its role. In the current implementation only the active replicator is used for BUM traffic replication.

Lenovo CNOS uses Bidirectional Forwarding Detection (BFD) over the VXLAN tunnel to speed up failure detection and enable backup replicators to forward VXLAN traffic.

VXLAN
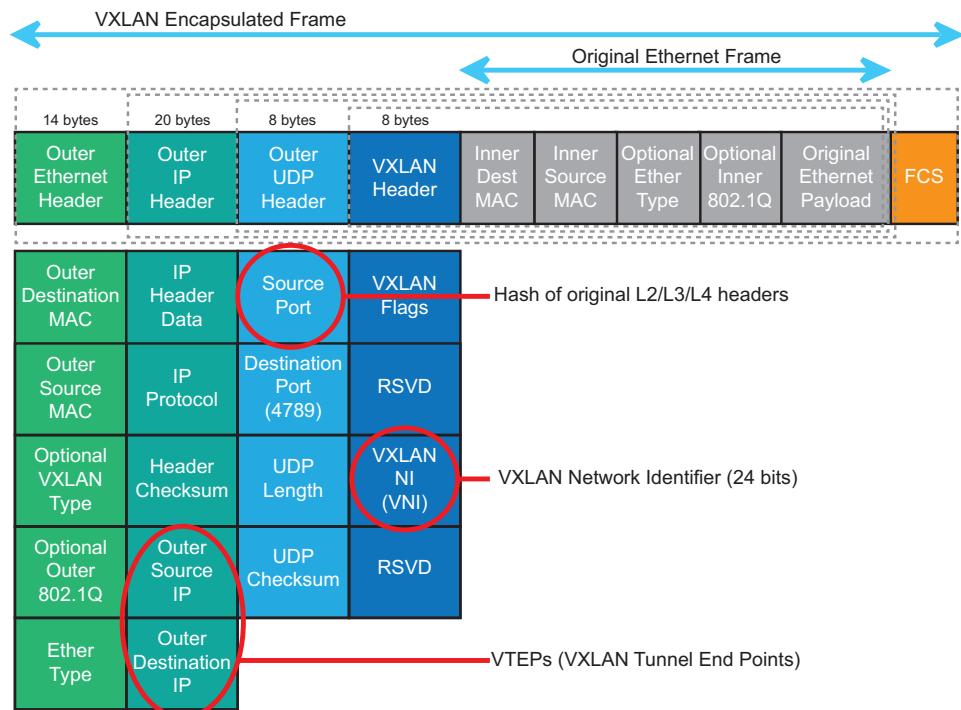
VXLAN is an extension to the VLAN protocol, designed to provide increased scalability in virtual networks. VXLAN is an Ethernet Layer 2 overlay protocol over a Layer 3 network. It uses an encapsulation method similar to VLAN that wraps MAC-based Ethernet Layer 2 frames with Layer 4 UDP packets, using destination UDP port 4789.

In typical physical networks, the number of VLANs is limited to 4094. VXLAN increases scalability up to 16 million logical networks and allows for Layer 2 adjacency across IP networks. This is achieved by adding a 24 bit segment ID to the VXLAN frame. The segment ID differentiates between individual logical networks, allowing millions of isolated Layer 2 VXLAN networks to coexist over the same Layer 3 infrastructure. Similar to VLANs, only virtual machines on the same VXLAN can exchange information with one another.

The virtualization of computing enables the mobility of virtual machines across different physical servers that exist in separate Layer 2 domains. This is done by tunneling virtual traffic over Layer 3 networks. Tunneling allows the dynamic distribution of resources within or across data centers without the limitations of Layer 2 boundaries or the necessity of creating large geographical Layer 2 domains.

VXLAN is an overlay technology that encapsulates Ethernet frames generated by physical or virtual workloads that are connected to the same logical Layer 2 segment. This segment is commonly called a Logical Switch (LS). The VXLAN frame format is shown in Figure 35.

Figure 35. VXLAN Frame Format

VXLAN uses Layer 2 over Layer 3 encapsulation. The Ethernet frame generated by a workload is wrapped within external VXLAN, UDP, IP, and Ethernet headers to ensure its transportation across the network infrastructure that connects the VXLAN endpoints together.

The first step in encapsulation is wrapping the Ethernet frame within a VXLAN header. This header uses a 24 bit VXLAN Network Identifier (VNI) that scales beyond the 4094 limitation of VLANs, allowing up to 16 million logical networks. Similar to VLANs, the VXLAN header is associated to an IP subnet. Internal IP subnet communication is achieved only between devices connected to the same virtual network or logical switch.

The second step is to encapsulate the VXLAN header within a UDP header. The Layer 2, Layer 3, and Layer 4 headers of the original Ethernet frame are hashed to determine the source port for the external UDP header. This ensures the load balancing of VXLAN traffic across equal cost paths available within the transport network infrastructure.

After adding the UDP header, the packet is encapsulated with an external IP header. The source and destination IP addresses of this header are used to uniquely identify the ESXi hosts originating and terminating the VXLAN frame encapsulation. These ESXi hosts are called VXLAN Tunnel End Points (VTEPs).

The final step is to wrap the packet in an external Ethernet header. This header uses the MAC address of VTEP associated with the original Ethernet frame as its source MAC address and the MAC address of the next-hop routing device as its destination MAC address.

Note: VXLAN encapsulation increases the size of the IP packet by wrapping the internal Ethernet frame in an external UDP header. The overall Maximum Transmission Unit (MTU) for all interfaces members of the physical infrastructure that will carry the VXLAN frame needs to be increased to a minimum of 1,600 bytes.
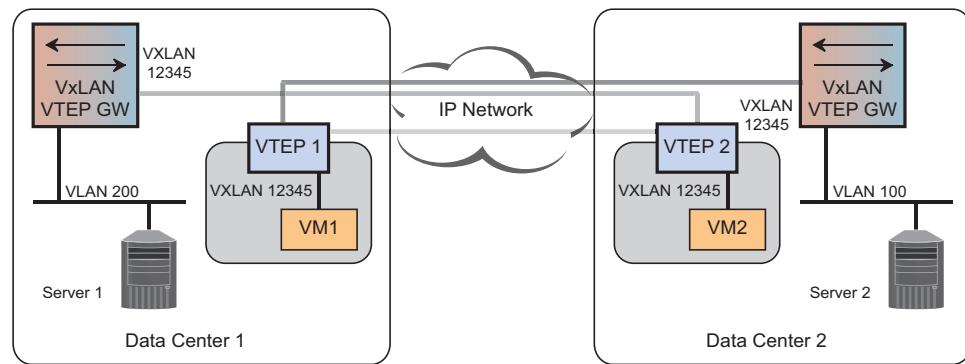
Lenovo VXLAN Gateway

In a typical cloud data center, physical servers and virtual machines need to share the same Layer 2 domain. A physical server is unable to communicate with a virtual machine using the VXLAN protocol.

A VXLAN Gateway is required to enable the communication between physical and virtual devices using the VXLAN protocol. The Gateway enables this by translating the VXLAN packet into a traditional VLAN.

The Lenovo VXLAN Gateway allows physical servers to consistently connect to virtual machines within a cloud infrastructure of a VMware NSX for vSphere environment.

Figure 36. VXLAN Gateway Services

The Lenovo VXLAN Gateway provides the following:

- Configuration and monitoring using ISCLI

- Packet counters for virtual ports and networks associated with the VXLAN Gateway

- Open vSwitch Database (OVSDB) Protocol for orchestration from Software-Defined Networking (SDN) Controller Node

- Full support for Bidirectional Forwarding Detection (BFD) to ensure SDN Replication Cluster availability

- Line rate packet forwarding for both VXLAN and non-VXLAN packets

To enable the VXLAN Gateway on the switch and configure HSC to run in VXLAN Tunnel Endpoint (VTEP) mode, enter:

```
Switch(config)# hsc mode vtep
```

To disable the VXLAN Gateway, enter:

```
Switch(config)# no hsc mode
```

To configure the VTEP, enter:

```
Switch(config)# hsc vtep
Switch(conf-vtep)#
```

Note: After issuing this command, you will enter VTEP configuration command mode.

To be part of the virtual network, the switch must inform the NSX Manager which of its interfaces will be used as a VTEP. To configure the IP address of the local VTEP, enter:

```
Switch(conf-vtep)# tunnel ip <IPv4 address>
```

To delete the VTEP IP configuration, enter:

```
Switch(conf-vtep)# no tunnel
```

To display the tunnels created between the local switch VTEP and other VTEPs across the virtual network, enter:

```
Switch> display nwv vxlan tunnels

Tunnel count: 261

Tunnel IP Address    Tunnel Type    Status
-----------------    -----------    ------
100.1.34.56          Local          UP
192.168.2.246        SN(Active)     UP
192.168.2.241        SN(Backup)     UP
192.168.2.242        SN(Backup)     UP
192.168.2.244        SN(Backup)     UP
192.168.2.245        SN(Backup)     UP
192.168.2.247        SN(Backup)     UP
192.168.2.248        SN(Backup)     UP
192.168.2.250        SN(Backup)     UP
172.16.1.2           Remote         UP
192.168.1.1          Remote         UP
192.168.1.2          Remote         UP
192.168.1.3          Remote         UP
192.168.1.4          Remote         UP
...
```

For the switch to operate as a VXLAN Gateway, it must synchronize network information with the NSX Controller. To configure the switch to use VMware NSX as the controller provider, enter:

```
Switch(config)# hsc vtep
Switch(conf-vtep)# controller provider nsx
```

To configure the IP address of the NSX Controller, enter:

```
Switch(conf-vtep)# controller ip <IPv4 address>
```

To configure the TCP port of the NSX Controller, enter:

```
Switch(conf-vtep)# controller port <1-65535>
```

To configure the Virtual Routing and Forwarding (VRF) instance used by the NSX Controller, enter:

```
Switch(conf-vtep)# controller vrf {default|management}
```

To delete the NSX Controller configuration from the switch, enter:

```
Switch(conf-vtep)# no controller
```

To verify the connection to the NSX Controller, enter:

```
Switch> display hsc ovsdb connection

Idx  Type          Peer            State   Inact. ms  Backoff ms  Latest Method
---  ------------  --------------  ------  ---------  ----------  -----------------
1    SSL (Active)  11.1.0.202:6640 ACTIVE  30000      8000        transact (comment)
2    SSL (Active)  11.1.0.203:6640 ACTIVE  30000      8000        monitor
3    SSL (Active)  11.1.0.204:6640 ACTIVE  30000      8000        transact (select)
```

To display information about the VXLAN Gateway, enter:

```
Switch> display hsc vtep

VTEP Information:
Status:            Enabled
HA Mode:           vLAG
Device Name:       HSC_BE2800
Tunnel IP:         100.1.34.56
BFD Status:        Enabled

Controller Connections:
Idx  Type                 Peer                      Inact.  Backoff ms
---  ------------------   -----------------------   ------  ---------
1    SSL (NSX Controller) 11.1.0.202:6640           30000   8000
2    SSL (NSX Controller) 11.1.0.203:6640           30000   8000
3    SSL (NSX Controller) 11.1.0.204:6640           30000   8000

VTEP Connections:
Id  Type        Peer                          Vrf         State
--  ----------  ----------------------------  ----------  -------
1   RESTful API Local                         default     running
2   RESTful API https://10.241.43.71:443      management  running

Physical Port Count: 1
Name                     Mappings  VLAN  VNID
-----------------------  --------  ----  ---------
vlag-instance-1          1000      1     5001
                                   2     5002
                                   3     5003
                                   4     5004
                                   17    5017
                                   18    5018
                                   ...
```

To display virtual network information, enter:

```
Switch> display hsc vtep virtual-network

Virtual Network Count: 1000
VNI        Name
----------  --------------------------
5001       03b264c5-9540-3666-a34a-c75d828439bc
5002       415585bd-389b-3965-9223-807d77a96791
5003       240ac937-1ec2-371a-a998-47c3ae2e6384
5004       3202111c-f90e-3c81-aa47-2aaceb72b0df
5017       0af78794-5872-396b-82c9-f73ead2565c8
5018       71463aaa-cf04-3fa2-8e7d-fa4558607545
5019       53fdae58-e861-376b-982b-0cd6beade809
...
```

Software Architecture Overview

The software architecture of the Lenovo VXLAN Gateway consists of three processes, each having multiple functional blocks:

● Network Virtualization Daemon (NWVD) - implements the VTEP Manager and the VXLAN core

● OVSDB Management Protocol - implements the Open vSwitch Database and the SSL client

● Hardware Switch Controller (HSC) - handles the interaction with the VMware NSX Controller through OVSDB

NWVD - Network Virtualization Daemon

NWVD communicates with other processes that run locally on the switch and it interacts with multiple CNOS functional blocks to collect information and handle events.

The VTEP Manager is a generic layer for VXLAN configuration, that receives the VXLAN network configuration: VTEPs, virtual networks, VTEP to virtual network mapping and local (switch interfaces) to virtual network mapping. It translates this information to messages for VXLAN to be used in data path calculations.

The VXLAN module receives network information from the VTEP manager (Local Tunnel End Points (LTEP) IP addresses, VTEPs, virtual networks, and more), it gathers all required information from other CNOS modules and sends messages to the Hardware Specific Layer (HSL) process which does the ASIC programming. The VXLAN module also handles events like link up/link down events, VLAN membership changes, route and ARP info changes, and MAC learning notifications from HSL.

The VXLAN module communicates with other CNOS processes to set up the hardware configuration and to receive Forwarding Database (FDB) updates, VLAN updates (create, delete, add port, remove port), ARP entries removal notifications, and route change notifications.

OVSDBD – Open Virtual Switch Database Daemon

The OVSDB module handles the communication with the NSX Controller using the OVSDB management protocol. It includes a handler for protocol messages, a connection manager, and a database manager.

The OVSDB process implements a SSL client which is used to interact with the Network Virtualization Protocol (NVP) controller. The client – server SSL connection uses a private key infrastructure for connection security. This involves a server certificate and a client certificate and key.

Similar with other protocols, OVSDB generates two certificates. By default the names are `ovsdb_mgmt` for the management Virtual Routing and Forwarding (VRF) instance and `ovsdb_default` for the default VRF instance.

To display the OVSDB certificates used by the switch, run the following command:

```
Switch> display ovsdb certificate

ovsdb pki ovsdb_mgmt vrf management
ovsdb pki ovsdb_default vrf default
```

To display all available certificates stored on the switch, enter:

```
Switch> display pki

PKI Profile Name: ovsdb_default
CSR: non-existed
Host certificate: existed
CA: 0
In used:Yes

PKI Profile Name: ovsdb_mgmt
CSR: non-existed
Host certificate: existed
CA: 0
In used:Yes

PKI Profile Name: rest_default
CSR: non-existed
Host certificate: existed
CA: 0
In used:Yes

PKI Profile Name: rest_mgmt
CSR: non-existed
Host certificate: existed
CA: 0
In used:Yes
```

To display a specific certificate, enter:

```
Switch> display pki <certificate name> host-certificate base64
```

You can generate new certificates using the names of the default certificates. For example, you can create a new certificate using the ovsdb_mgmt certificate:

```
Switch(config)# pki ovsdb_mgmt
Switch(config-pki)# host-cert generate
```

For example, to display the ovsdb_mgmt certificate, enter:

```
Switch> display pki ovsdb_mgmt host-certificate base64

-----BEGIN CERTIFICATE-----
MIIEAjCCAuqgAwIBAgIJAIItdlYjRJKMMA0GCSqGSIb3DQEBCwUAMIGYMQswCQYD
VQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcm5pYTEUMBIGA1UEBwwLU2FudGEgQ2xh
cmExLTArBgNVBAoMJExlbm92byBOZXR3b3JrIE9wZXJhdGluZyBTeXN0ZW0gQ05P
UzEdMBsGA1UECwwUTmV0d29yayBFbmdpbmVlcmluZ2kxEDAOBgNVBAMMBzAuMC4w
LjAwHhcNMTcwNTI4MTIyNzU1WhcNMTgwNTI4MTIyNzU1WjCBmDELMAkGA1UEBhMC
VVMxEzARBgNVBAgMCkNhbGlmb3JuaWExFDASBgNVBAcMC1NhbnRhIENsYXJhMS0w
KwYDVQQKDCRMZW5vdm8gTmV0d29yayBPcGVyYXRpbmcgU3lzdGVtIENOT1MxHTAb
BgNVBAsMFE5ldHdvcmsgRW5naW5lZXJpbmdpMRAwDgYDVQQDDAcwLjAuMC4wMIIB
IjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwKwtmynRgjLWyjUb5IBNmGQ+
8kY2hvs3syQx+wVQYYt+XYi+/D89o68ITZxqTOezfUQJE2SXnTT73lWEJLfvd9F3
ttKFMk3nQijefQpLhI/J3SkZB2B8pnxi5RtSBxXGnNd2+0+nMyeuIC0Dnm8itu93
cd2Ks8iRncALDNOhoQyJTVWxgdEG/rIUFLveeSlKG0SMy25hvtRc44D639+c9eGp
RbRUHFIlpn0AlnMjnSf1rtKeQN9NBUtJNYzEW894J4HePPBUKPltLH+OzqKTT3Zj
yOHPi+hzt2Aey6EaOXyj0q4WbSudiL6zIvqB2N05+yWeJK707TznoZ6++Y8aFQID
AQABo00wSzAdBgNVHQ4EFgQUpf9+p9Hb5MSMDZV3HhQmhfVvvqEwHwYDVR0jBBgw
FoAUpf9+p9Hb5MSMDZV3HhQmhfVvvqEwCQYDVR0TBAIwADANBgkqhkiG9w0BAQsF
AAOCAQEAUJTJCdaVKBH8ZwpjfSRdGRClJWoTole5Qa4DbhhSJR4VbH4QhxuHuaI+
UmrKqve7G7XkaXxf078vB4Z+mQkvIYpWJEm99ZC43swVVKkKCxGm0yikzAmVGnzn
/LJJR4w1EbOXm+OAecG4/QQuvul7KkYLqvjQCSOt4rUUFDAUJSJV5zPQ2c2xqySo
HF6skFpEc1SginCVrzy76AJiEusPEXV8wmi1jFQOkhIIYwdscUJP6MAGnU7Jybji
O3Ar6L4gopIYpACEajGUcahXhT+svxQg+nsB4TgYP3KICjVRRYqxk5AbfLEmEc2j
DKz5wE0oeYxoO0gV7nZbWqtloXGblA==
-----END CERTIFICATE-----
```

You can use other certificate names as long as they meet the following conditions:

- a maximum length of 16 characters
- the name contains "ovsdb"
- the name contains "default" if the certificate is used on the default VRF instance, or "mgmt" if the certificate is used on the management VRF instance

For example, you can generate a certificate called ovsdb_11_mgmt, which can be used on the management VRF instance. Before using the new certificate, you must first generate the certificate using the following command:

```
Switch(config)# pki ovsdb_11_mgmt
Switch(config-pki)# host-cert generate
```

After generation, the new certificate must be assigned to the OVSDB protocol:

```
Switch(config)# ovsdb pki ovsdb_11_mgmt vrf management
```

Once the switch connects to controller, it will use the new certificate. If the new certificate is not assigned to the OVSDB protocol, the switch is not going to use the certificate.

HSC - Hardware Switch Controller

The HSC process (HSCD) is a software module that runs on a switch, which implements the HSC Manager, the OVSDB client, and the RESTful API client. HSCD communicates with other processes within the same switch using inter-process communication (IPC). It also communicates vLAG VTEP switches using RESTful API over SSL.

HSC acts like the interface to the NSX Controller. When creating a VXLAN Gateway, the management IP address of the HSC or the SSL certificate needs to be added to the NSX Controller. The OVSDB server and the HSC agent need to run simultaneously on the HSC.

HSC retrieves the configuration of each vLAG switch, such as hostname, switch description, MAC address and so on, and transmits to each vLAG switch the VTEP IP address and vLAG port list.

When HSC receives the configuration of the vLAG switches, it registers the information with the OVSDB server and monitors the OVSDB tables. The NSX Controller transmits its configuration to OVSDB, which sends an update message to the HSC. The configuration is converted to RESTful API messages and sent to the vLAG switches. The Network Virtualization Daemon (NWVD) on the primary vLAG switch parses the RESTful API messages and call the VXLAN APIs to program the data path.

When configuring the two vLAG switches to act as VXLAN Gateways, there are no master/backup roles. Each switch behaves as standalone and replies with its own configuration to the HSC, which maintains duplicated information.

Because each vLAG switch behaves as standalone, the network is not affected by failover. If one of the two vLAG switches becomes unreachable, due to failure or connection problems, the HSC re-programs the switch when it is reachable once more.

While there is no master role for the vLAG switches when running as VXLAN Gateways, Bidirectional Forwarding Detection (BFD) runs only on one of the switches, because both devices share the same VTEP IP address. HSC decides which switch runs BFD.

Every 30 seconds, HSC gathers BFD and status information from the vLAG switches and sends this information to the NSX Controller as an OVSDB message over SSL. The OVSDB message is sent only if there are value changes.

Lenovo VXLAN uses vLAG to implement VMware NSX High Availability (HA). To facilitate the communication between VXLAN and legacy servers, VXLAN is designed as an added layer on top of vLAG. HA is achieved by means of ECMP running locally on each switch.
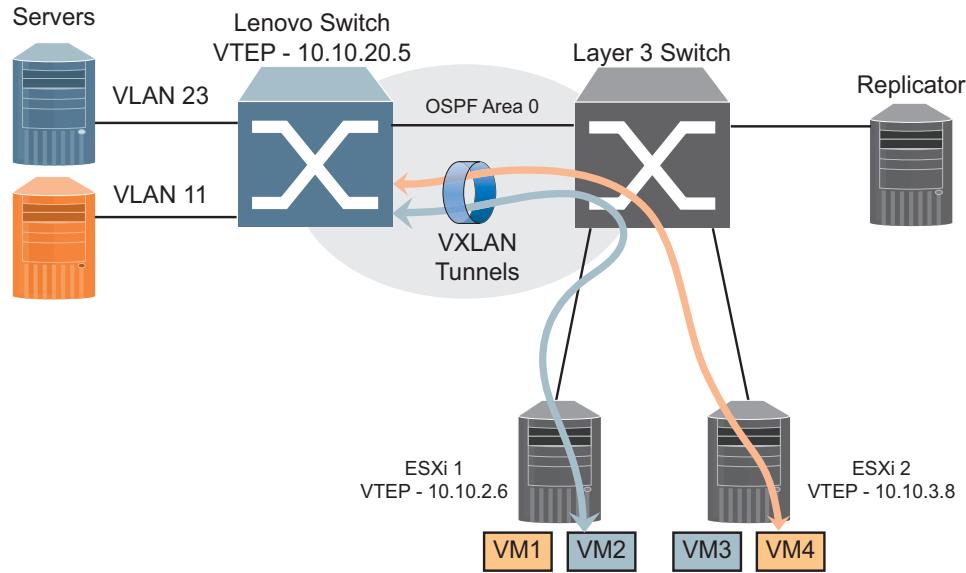
VXLAN Gateway Standalone Topologies

The following deployment topologies are supported by the Lenovo VXLAN Gateway:

VXLAN Tunnels over Layer 3 Routed Network

This topology shows the servers directly connected to the switch ports in access mode while connectivity between the VTEPs is Layer 3 routed.
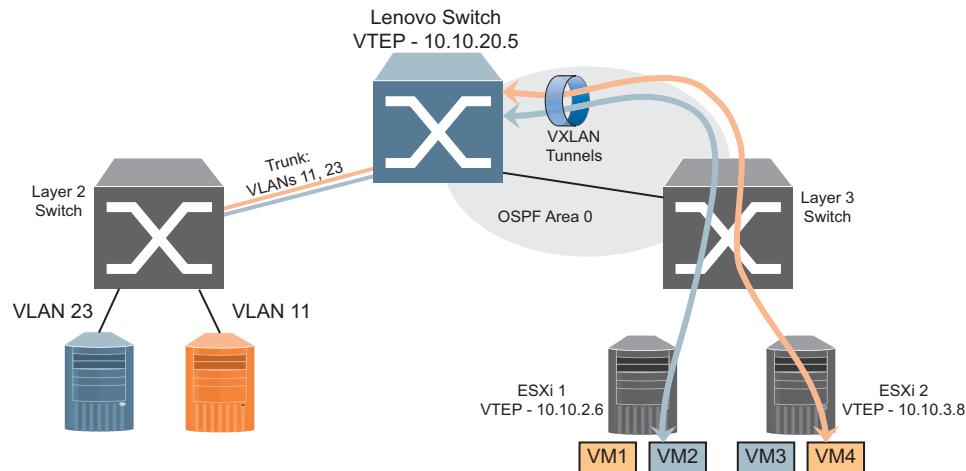
Figure 37. VXLAN Tunneling over Layer 3 Routed Network

Physical Servers on Layer 2 Switches

This topology allows connecting physical servers using VLAN trunks into the switch VXLAN gateway switch.
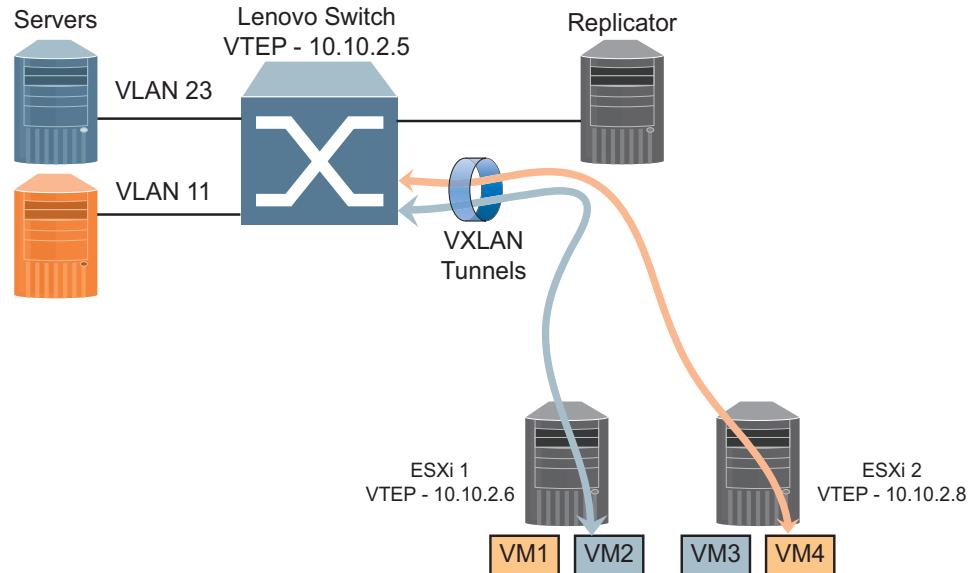
Figure 38. VXLAN Tunneling over VXLAN Gateway Switch

Directly Attached VXLAN Tunnel with a Layer 2 Network (Not Supported)

This topology is not supported when all nodes including ESXi and physical servers are connected directly to the switch VXLAN Gateway switch in a Layer 2 configuration.
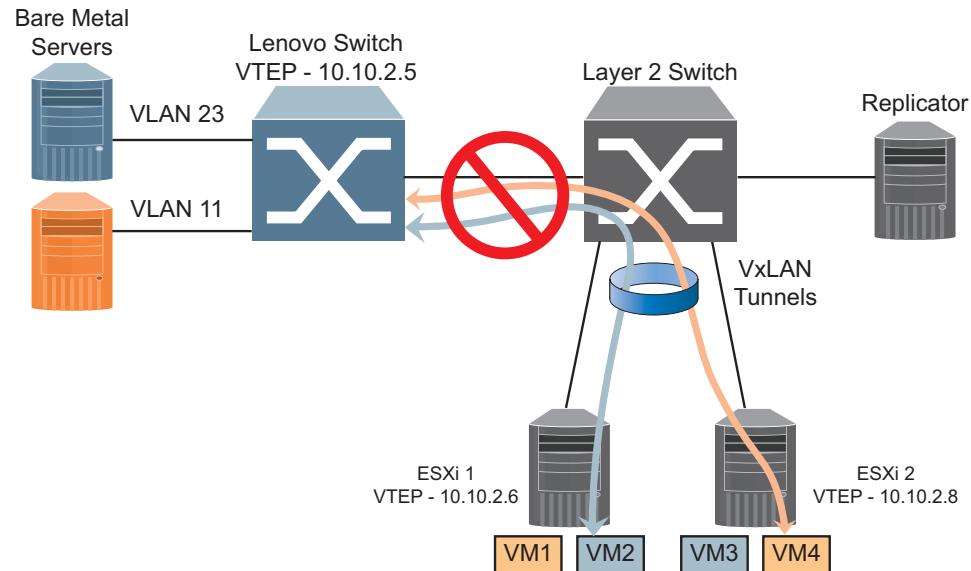
Figure 39. VXLAN Tunneling over Layer 2 Network

VXLAN Tunnels through a Layer 2 Network (Not Supported)

This topology is not supported when multiple VTEPs are connected to the Lenovo VXLAN gateway switch using a Layer 2 switch.

Figure 40. VXLAN Tunneling over Layer 2 Switches

The switch has a hardware restriction that does not allow this kind of topology. It supports only a single next-hop per network port. It cannot initiate tunnels from a single network port to multiple remote Tunnel End Points (TEPs) across a Layer 2 network. Instead, it can initiate tunnels from a single network port to multiple remote TEPs across a Layer 3 network.

High Availability Support

The Lenovo VXLAN Gateway supports Virtual Link Aggregation Group (vLAG) and Equal Cost Multiple Paths (ECMP) to provide an active-active, fully redundant high availability solution.

The VXLAN solution is configured and administered by the NSX Manager through the Management Network. Traffic that is transmitted across this network is used by VMware NSX to manage each device that is part of the VXLAN solution, such as virtual machines and switches.

Data traffic that is not used in the management process is transmitted across the Transport Network. Such traffic includes, for example, the transfer of a file from a legacy server to a virtual machine. The file is divided into smaller packets that are sent to the virtual machine through the Transport Network using FTP.

The Management and Transport Networks can be physically separated or combined. This means that the two networks can have distinct physical links that connect all the devices in the VXLAN solution as shown in Figure 41, or that they share the same physical links as shown in Figure 42.

Figure 41. High Availability Topology Solution with Separated Management and Transport Networks
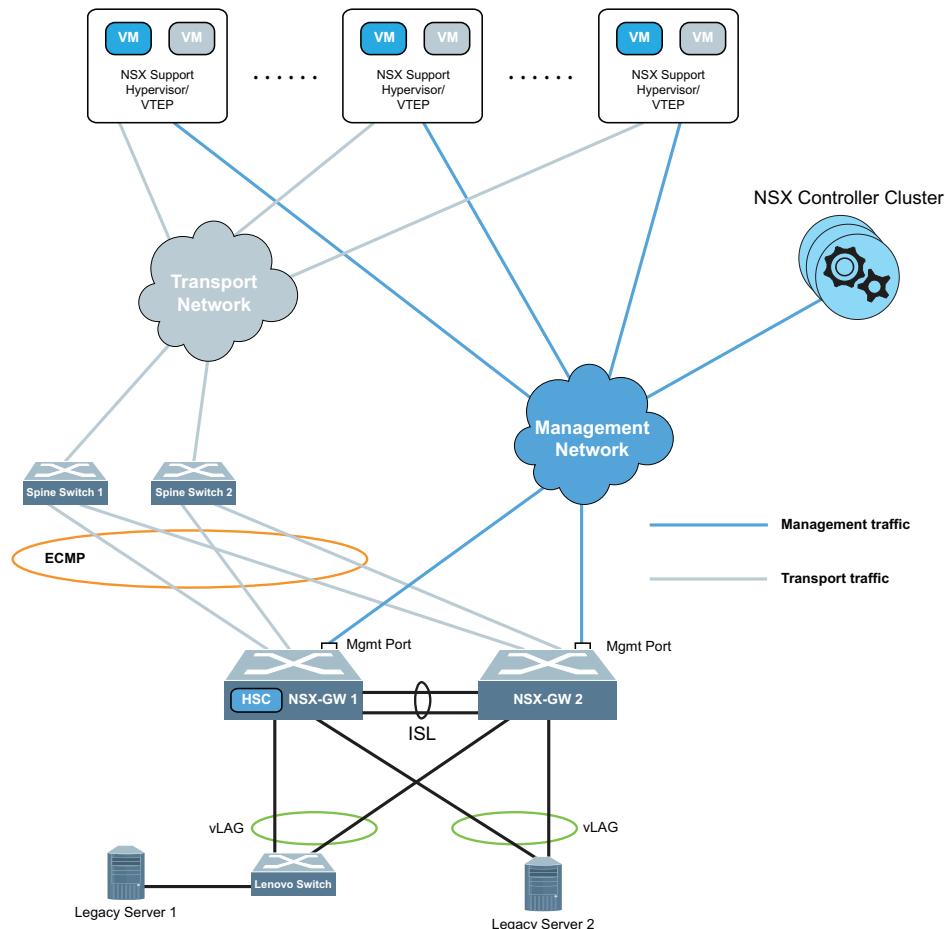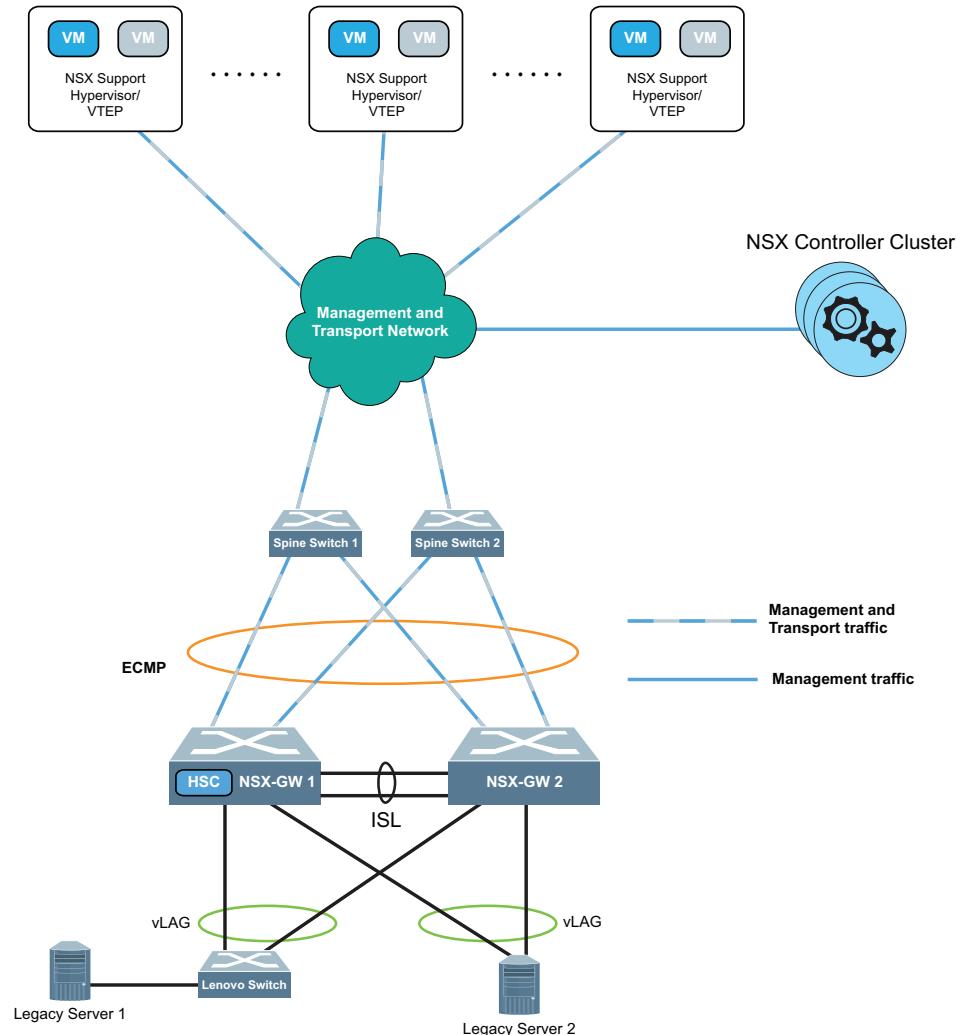
Figure 42. High Availability Topology Solution with Combined Management and Transport Networks

Note: The NSX Manager uses only the Management Network to configure and monitor the VXLAN solution. It does not have a link to the Transport Network. As shown in the two previous figures, regardless if the Management and Transport Networks are separated or combined, the NSX Controller Cluster is connected to the rest of the VXLAN solution only through a management link (colored in blue), and not a transport link (colored in grey).

The connection is established between the NSX Controller and a dedicated piece of software called the Hardware Switch Controller (HSC). The HSC can be embedded in the physical switches or can run as a separate standalone process. The HSC can control one or several physical gateways.
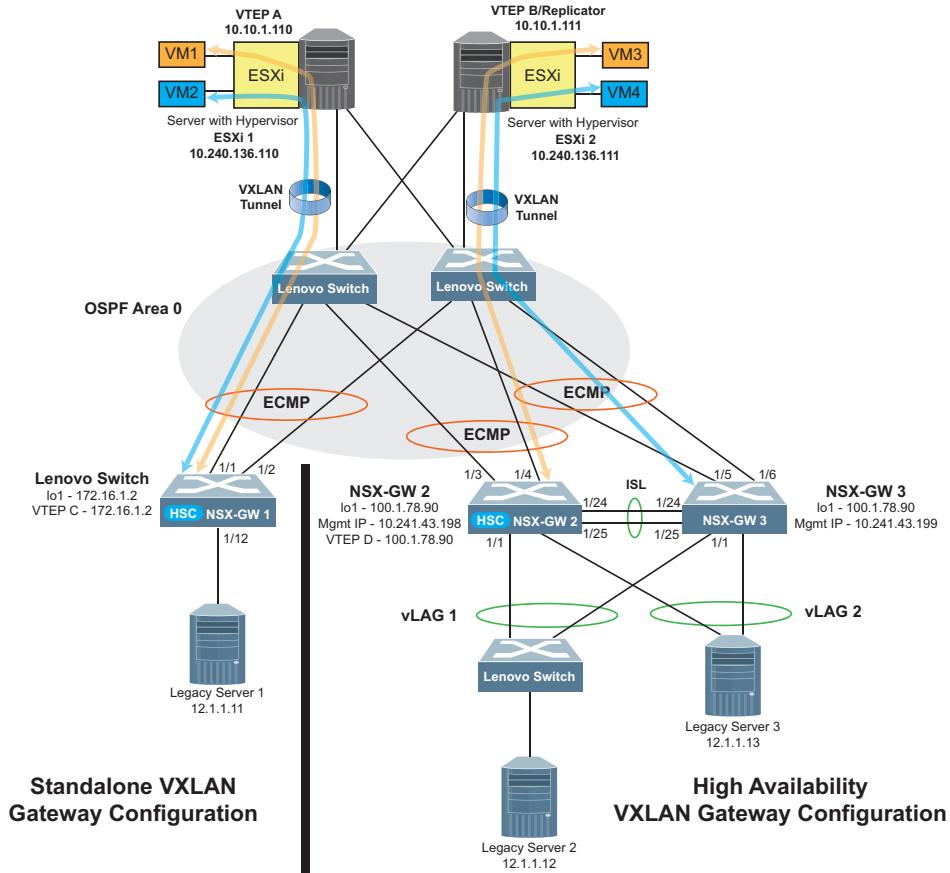
During operation, HSC registers with the NSX Controller and uses the OVSDB protocol to synchronize topology information, MAC to VXLAN Endpoints, and VXLAN ID bindings with the NSX Controller. HSC appropriately programs the Lenovo vLAG switch pairs as the VMware NSX physical gateway via RESTful API. This physical gateway integration allows for the nearly instantaneous synchronization of states between physical and virtual VXLAN Tunnel Endpoints during any network change or workload modification event.

VXLAN Gateway Configuration Example

The following example shows the steps required to configure a VXLAN Gateway for VMware NSX. The figure presents two topology scenarios:

- NSX-GW 1 is a standalone switch acting as VTEP C
- NSX-GW 2 and NSX-GW 3 form the vLAG topology which acts as VTEP D

Figure 43. VXLAN Gateway Example

Standalone VXLAN Gateway Configuration Example

Use the following steps to configure NSX-GW 1 as a standalone VXLAN Gateway as shown in the lower left side of Figure 43.

Given the following:

● VTEP C IP address: 172.16.1.2

● HSC is implemented on NSX-GW 1

1. Configure the DNS and NTP server addresses to get the correct time and avoid certificate failure check due to certificate expiration.

    a. Enable DNS on the switch and configure DNS server addresses:

    ```
    Switch(config)# ip domain-lookup
    Switch(config)# ip name-server <DNS server address> vrf management
    ```

    b. Enable NTP and configure the NTP server address:

    ```
    Switch(config)# ntp enable
    Switch(config)# ntp server <NTP server address> mgt-port
    ```

2. Configure routed interfaces for ECMP:

    ```
    Switch(config)# interface Ethernet 1/1
    Switch(config-if)# no bridge-port
    Switch(config-if)# ip address 172.45.17.2/24
    Switch(config-if)# exit

    Switch(config)# interface Ethernet 1/2
    Switch(config-if)# no bridge-port
    Switch(config-if)# ip address 172.45.18.2/24
    Switch(config-if)# exit
    ```

3. Assign the hardware VTEP an IP address by using either a loopback interface or a routed interface. For this example, loopback interface 1 is used:

    ```
    Switch(config)# interface loopback 1
    Switch(config-if)# ip address 172.16.1.2 255.255.255.255
    Switch(config-if)# exit
    ```

4. Configure the NSX Controller IP address and Controller port on the switch:

```
Switch(config)# hsc vtep
Switch(config-vtep)# controller ip 11.1.0.202
Switch(config-vtep)# controller port 6640
Switch(config-vtep)# controller vrf default
Switch(config-vtep)# exit
Switch(config)#
```

To verify that the connection to the NSX Controller is active, enter:

```
Switch(config)# display hsc ovsdb connection

Idx  Type          Peer            State   Inact. ms  Backoff ms  Latest Method
---  ------------  --------------  ------  ---------  ----------  ------------------
1    SSL (Active)  11.1.0.202:6640  ACTIVE  30000      8000        transact (comment)
2    SSL (Active)  11.1.0.203:6640  ACTIVE  30000      8000
3    SSL (Active)  11.1.0.204:6640  ACTIVE  30000      8000        monitor
```

5. Configure the local VXLAN TEP IP address:

```
Switch(config)# hsc vtep
Switch(conf-vtep)# tunnel ip 172.16.1.2
```

Note: The VTEP IP address is the IP address of the loopback interface configured at Step 3.

6. Configure VXLAN on the switch ports that will physically participate in the virtual network:

```
Switch(config-vtep)# vtep 1 vxlan-ports ethernet 1/12
```

7. Configure the username and password of the VTEP instance:

```
Switch(config-vtep)# vtep 1 username <username> password <password>
Switch(config-vtep)# exit
Switch(config)#
```

8. Enable the VXLAN Gateway:

```
Switch(config)# hsc mode vtep
```

9. The switch is now configured as a VXLAN Gateway. The next step is to add it to NSX Manager as a Hardware Device:

   a. Generate and obtain the PKI certificate from the switch:

```
Switch(config)# pki ovsdb_mgmt
Switch(config-pki)# host-cert generate
Switch(config-pki)# exit

Switch(config)# display pki ovsdb_mgmt host-certificate base64

-----BEGIN CERTIFICATE-----
MIIDezCCAmOgAwIBAwIBADANBgkqhkiG9w0BAQsFADCBgDELMAkGA1UEBhMCVVMx
FDASBgNVBAcMC1NhbnRhIENsYXJhMSswKQYDVQQKDCJMZW5vdm8gTmV0d29ya2lu
ZyBPcGVyYXRpbmcgU3lzdGVtMRwwGgYDVQQLDBNOZXR3b3JrIEVuZ2luZWVyaW5n
MRAwDgYDVQQDDAcwLjAuMC4wMB4XDTE3MDUwNTA4NTgwN1oXDTI3MDUwMzA4NTgw
N1owgYAxCzAJBgNVBAYTAlVTMRQwEgYDVQQHDAtTYW50YSBDbGFyYTErMCkGA1UE
CgwiTGVub3ZvIE5ldHdvcmtpbmcgT3BlcmF0aW5nIFN5c3RlbTEcMBoGA1UECwwT
TmV0d29yayBFbmdpbmVlcmluZzEQMA4GA1UEAwwHMC4wLjAuMDCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAK+xpPdfJWzU4B4OYHw9E1ImDwisfz8uCaZu
bG93H44+gklSCDAshk6Mj+z97K232Sb+wYYRfV22NZ6vRY02g+oTjr0MYSTn1TCy
eT4E3RJZb66R2+TLS0HO9KCA17Sv+d74rwVDkQI8zO5t1ZfDK2UJDrZiZx4UPLXj
5ncAM+6zYr0z5BhSH+mpVDHyQcAdsTQgkYu81XRLNZjjndGxlfblU3D6rTaAg26y
d9AUXerKem4aAevT4VG817/lVa9SqTLc0bnkGZCNSM2TbDHsuNo9y35koTw0mBOv
FWZGjOeBJUNt7MXcmDkZvm/KhzmGoXQCGHZyeXAvv+H7O9mAXVkCAwEAATANBgkq
hkiG9w0BAQsFAAOCAQEAQbO52RPzXr4qj75epyZExJvPUygv2JGhr8qlhkrvbqch
6o/dyE0wgiFYN1rzpyMJ/DzOS//+fjB3EyMeo23m0lPdafPKShMHIHb7znKkROgz
3dEd8ozyQzLcxP0L6Lb+MTgImTBDlFxuPgXA05L6unVEY6fYv4/aIDv+7n9vhVzk
uExUprCIy4OjCfaKfplcjf4iY7KWTSJkoBjiackM1SoVQLJcNpE2CzDwqaPXQIZS
wMthE7O7QwjB/FsQt6b2xb7SBpzwiBjqq/FdbMV0bogRJRNESK4LYHdqbPbNrjaw
QlzW4WdDyOhOGcw2VWkytQcvEzwv10uTkzZSPmNuSg==
-----END CERTIFICATE-----
```

   b. Using the VMware vSphere Web Client add the switch as a Hardware Device. Attach the physical switch ports configured at step 5 to Logical Switches as required for network connectivity. For more details on how to add the Hardware Gateway, see the VMware NSX Administration Guide.

High Availability VXLAN Gateway Configuration Example

To configure a high availability solution as shown in the lower right side of Figure 43, you need to configure the HSC and a vLAG between the two VXLAN Gateways (NSX-GW 2 and NSX-GW 3).

In this example, consider the following:

- the management IP address of NSX-GW 2: 10.241.43.198

- the management IP address of NSX-GW 3: 10.241.43.199

- VTEP D IP address: 100.1.78.90

- the NSX Controller is configured on NSX-GW 2 with IP address 11.1.0.202

Basic Switch Configuration

On both VXLAN Gateway switches (NSX-GW 2 and NSX-GW 3), configure the DNS and NTP server addresses to get the correct time and avoid certificate failure check due to certificate expiration.

1. Enable DNS on the switch and configure DNS server addresses:

```
Switch(config)# ip domain-lookup
Switch(config)# ip name-server <DNS server address> vrf management
```

2. Enable NTP and configure the NTP server address:

```
Switch(config)# ntp enable
Switch(config)# ntp server <NTP server address> mgt-port
```

vLAG Configuration

Configure a vLAG between the VXLAN Gateways (NSX-GW 2 and NSX-GW 3). On both switches, ethernet ports 1/24 and 1/25 are used as the Inter-Switch Link (ISL) and ethernet port 1/1 forms the vLAG between them.

Note: The following steps need to be configured on both switches.

1. Create a static or an LACP LAG on ethernet port 1/1:

- Static LAG:

```
Switch(config)# interface ethernet 1/1
Switch(config-if)# aggregation-group 2 mode on
Switch(config-if)# exit
Switch(config)#
```

- LACP LAG:

```
Switch(config)# interface ethernet 1/1
Switch(config-if)# aggregation-group 2 mode active
Switch(config-if)# exit
Switch(config)#
```

2. Create a static LAG on ethernet ports 1/24 and 1/25:

```
Switch(config)# interface ethernet 1/24-25
Switch(config-if-range)# aggregation-group 1 mode on
Switch(config-if-range)# exit
Switch(config)#
```

3. Configure the vLAG tier ID:

```
Switch(config)# vlag tier-id 1
```

4. Configure NSX-GW 2 as the primary vLAG switch:

```
Switch(config)# vlag priority 1
```

5. Configure NSX-GW 3 as the secondary vLAG switch:

```
Switch(config)# vlag priority 2
```

6. Configure the vLAG ISL on the LAG which includes ethernet port 1/24:

```
Switch(config)# vlag isl port-aggregation 1
```

7. Configure the health check IP address of the peer switch, using the management IP address of the peer switch:

   a. Configure NSX-GW 2 with the management IP address of NSX-GW 3:

```
Switch(config)# vlag hlthchk peer-ip 10.241.43.199 vrf management
```

   b. Configure NSX-GW 3 with the management IP address of NSX-GW 2:

```
Switch(config)# vlag hlthchk peer-ip 10.241.43.198 vrf management
```

8. Globally enable the vLAG:

```
Switch(config)# vlag enable
```

9. Configure the vLAG instance and enable it:

```
Switch(config)# vlag instance 1 port-aggregation 2
Switch(config)# vlag instance 1 enable
```

ECMP takes advantage of the available links and balances the VXLAN encapsulated traffic egressing the VXLAN Gateway. You need to configure the VTEP IP address on a loopback interface for both vLAG switches. Also, routed interfaces or SVI interfaces can be configured for ECMP.

Note: When the VXLAN tunnel is an SVI, we recommend you keep only VXLAN-transport network facing ports as VLAN members. This is required for the VXLAN software module to detect that the tunnel interface is down when all VLAN ports are down (have no route to the VXLAN transport network).

10. Configure routed interfaces for ECMP:

a. NSX-GW 2:

```
Switch(config)# interface Ethernet 1/3
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 100.1.7.2/24
Switch(config-if)# exit

Switch(config)# interface Ethernet 1/4
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 100.1.8.2/24
Switch(config-if)# exit
```

b. NSX-GW 3:

```
Switch(config)# interface Ethernet 1/5
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 100.1.9.2/24
Switch(config-if)# exit

Switch(config)# interface Ethernet 1/6
Switch(config-if)# no bridge-port
Switch(config-if)# ip address 100.1.10.2/24
Switch(config-if)# exit
```

11. On both vLAG switches, configure a loopback interface which is used as the Tunnel IP address. This configuration must be the same on both vLAG peers:

```
Switch(config)# interface loopback 1
Switch(config-if)# ip address 100.1.78.90/32
```

HSC Configuration

Note: HSC needs to run only on a single switch that is part of the vLAG. It is recommended that you configure HSC on the primary vLAG switch.

1. Enter VTEP configuration mode:

```
Switch(config)# hsc vtep
```

2. Enable vLAG High Availability (HA) mode:

```
Switch(config-vtep)# ha mode vlag
```

3. Configure the NSX Controller:

```
Switch(config-vtep)# controller provider nsx
Switch(config-vtep)# controller ip 11.1.0.202
Switch(config-vtep)# controller port 6640
Switch(config-vtep)# controller vrf default
```

4. Configure the VTEP IP address:

```
Switch(config-vtep)# tunnel ip 100.1.78.90
```

Note: The VTEP IP address is the IP address of the loopback interface configured during vLAG Configuration at Step 11.

5. Configure the local VTEP instance:

```
Switch(config-vtep)# vtep 1 vxlan-ports vlag-instance 1
```

6. Configure the username and password of the local VTEP instance:

```
Switch(config-vtep)# vtep 1 username <username> password <password>
```

7. Configure the VTEP instance corresponding to the vLAG peer:

```
Switch(config-vtep)# vtep 2 ip 10.241.43.199
Switch(config-vtep)# vtep 2 vrf management
Switch(config-vtep)# vtep 2 vxlan-ports vlag-instance 1
```

8. Configure the username and password for the VTEP instance of the vLAG peer:

```
Switch(config-vtep)# vtep 2 username <username> password <password>
```

9. Exit VTEP configuration mode and configure HSC to run in VXLAN Tunnel Endpoint (VTEP) mode:

```
Switch(config-vtep)# exit

Switch(config)# hsc mode vtep
```

Chapter 33. Network Policy Agent

Lenovo Cloud Network Operating System provides a network policy agent that works with Nutanix's Virtual Domain Module (VDM). The VDM is responsible for interaction with the external Virtual domain manager.

This chapter describes how to use the Lenovo VDM plugin to obtain updates on events in a Nutanix cluster. These updates are then used to provide the switch with a view of the virtual network, and to automatically configure updated VLAN to port mappings for event changes.

The following topics are discussed in this section:
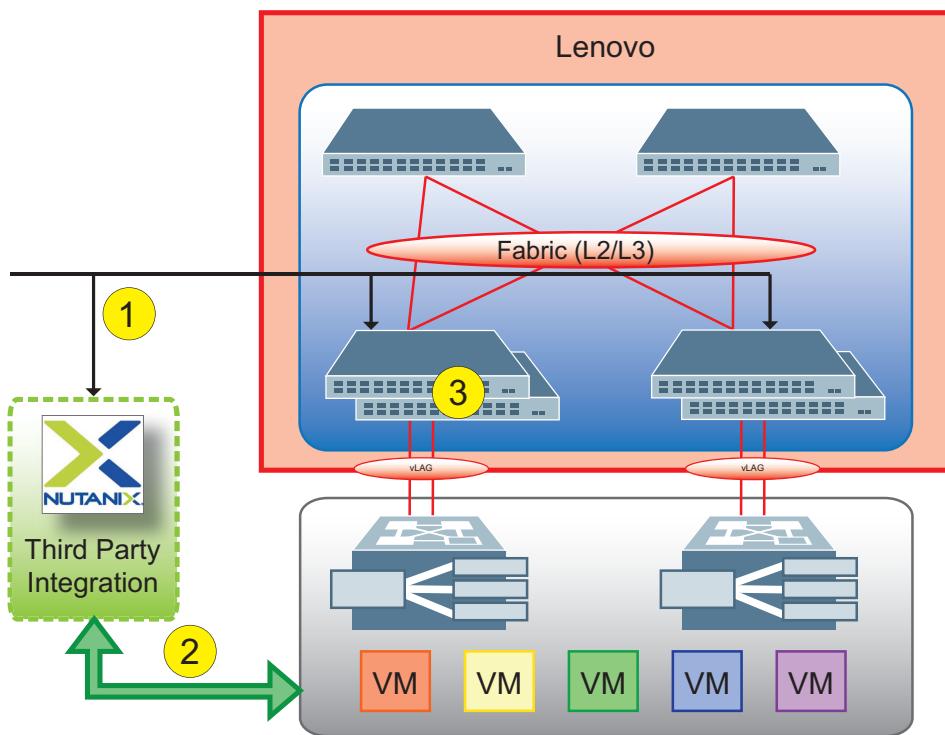
Overview

In traditional IT, the infrastructure administrator relies on the network administrator to provision, operate, and troubleshoot network-related issues, and the boundary of the network typically starts upwards from the server NIC. However, the dynamics are different in a Hyperconverged Infrastructure (HCI) solution.

HCI Point of Delivery (POD) architecture typically includes at least two pairs of Top-of-Rack (ToR) switches. The infrastructure administrator typically takes up the role of managing the entire solution component. It is therefore necessary to reduce the complexity of network language associated with network configuration of sn HCI POD. The network still exists, but it needs to disappear as much as possible and integrate seamlessly from a workload provisioning and operational point of view. Figure 44 shows VM workload visibility and positioning.

Figure 44. VM Workload Visibility and Provisioning

The steps in this figure are:

1. Integration with the Virtual Management Module (VMM) for notifications and topology discovery. The following notifications happen:

   - VM and vNIC Create, Read, Update, and Delete (CRUD)

   - Virtual network CRUD

2. The VMM controls and manages the virtual network infrastructure management and VM workload and overlay provisioning.

3. The workload visibility includes VM information and overlay network information, and topology discovery associates VMs to the physical switch interface. Dynamic network provisioning occurs:

- The physical network (L2 VLAN) is provisioned based on virtual network and VM associations.

- The Network Policy Agent dynamically handles VM migrations and moves L2 VLAN configurations.

The Lenovo CNOS Network Policy Agent gives increased visibility of the virtual infrastructure, including workload (VM) information and virtual network information. It also provides auto-discovery of any changes to the virtual network topology.

The CNOS Network Policy Agent also provides automatic VM-aware provisioning. The physical network is automatically configured based on the auto-discovered topology. The agent provides ongoing, dynamic updates to the physical network configuration in response to new VMs, updated VMs, and deleted VMs, eliminating errors with manual configuration.

Note: You must have Nutanix AOS version 5.0.2 or above installed for the CNOS Network Policy Agent to work properly.

Table 45 explains the advantages of using the Network Policy Agent.

Table 45. *Advantages of Using the Network Policy Agent*

Action	Without Network Policy Agent	With Network Policy Agent
Network settings update based on VM and virtual network configuration	Pending changes are communicated to Network Administrator, who manually reconfigures each physical switch connected to new VM. This process repeats each time changes in virtual network occurs.	CNOS Network Policy Agent communicates with VMM to automatically learn when VM changes occur and update switch configuration accordingly. No scheduled maintenance window or user intervention is required.
VM Live Migration	The Network Administrator must identify the new location of a VM and must configure new connectivity information. This results in network downtime.	A new VM location is detected, and the physical switch automatically updates. No network interruption occurs.
Virtual Network Visibility	Physical switch has no visibility into the virtual network and is only aware of physical host connectivity.	An unprecedented view of a virtual network's VMs, hosts, and their connectivity is now available to the physical switch.

You use the switch command-line interface to set up the Nutanix VDM Plugin.

Setting up the Nutanix VDM Plugin

To set up the switch to work with the Nutanix VDM Plugin:

Note: LLDP feature is required for Nutanix VDM Plugin. Please make sure it is enabled on your switch.

1. Modify the LLDP transmit delay value to one:

```
Switch(config-if)# lldp transmit-delay 1
```

2. Modify the LLDP transmit interval to five seconds (default value is 30 seconds) for faster transmission of LLDP packets:

```
Switch(config-if)# lldp timer 5
```

3. Enable provisioning of virtual domain information on the interface.

   Note: Make sure the interface belongs on server facing ports, and not on uplinks of VLAG ISL ports or any port that will not be controlled by the VDM Plugin.

```
Switch(config-if)# auto-policy enable
Switch(config-if)# exit
```

   Note: The no auto-policy enable command turns off provisioning of virtual domain information.

4. Enable the REST API:

   - via HTTPS (default):

```
Switch(config)# feature restApi
```

   - via HTTP:

```
Switch(config)# feature restApi http
```

   Note: Depending on the RestAPI server configuration, the subscription URLS for VM and VNETWORK on the client URL will reflect the corresponding changes.

5. Move into the Nutanix VDM context:

```
Switch(config)# vdm nutanix
```

6. Specify the IP address of the Nutanix cluster:

```
Switch(config-vdm)# ip address <IP_address> vrf management
```

where:

Parameter	Description
IP address	Virtual IP address of the Nutanix cluster

7. Specify the user name and password to access the Nutanix cluster:

```
Switch(config-vdm)# user <username> password <password>
```

8. Specify the switch port interfaces or aggregations on which you want to have the Nutanix cluster provision:

```
Switch(config-vdm)# add [interface Ethernet <slot>/<port> | port-aggregation
<aggregation ID>]
```

where:

Parameter	Description
slot	The ethernet slot
port	The ethernet port
aggregation ID	The port aggregation ID

9. Configure the Virtual Machine information for inbound packets:

```
Switch(config-vdm)# vm {name "<VM name>" | uuid <VM UUID>} vnic {vlan
<VID> | ip <IP address> | mac <MAC address>} attach security-policy <policy name>
in
```

where:

Parameter	Description
VM name	The Virtual Machine name, in quotes.
VM UUID	The VM UUID.
VID	The VLAN ID.
IP address	The IP address.
MAC address	The MAC address in *HHHH.HHHH.HHHH* format.
policy name	The name of the security policy.

10. Configure the URL for refreshing the Virtual Machine information on the switch:

```
Switch(config-vdm)# refresh-vms-url <URL>
```

where:

Parameter	Description
URL	The URL to get the list of VMs

This is often:

```
Switch(config-vdm)# refresh-vms-url
https://<IP address>:<port>/api/nutanix/v3/vms/list
```

where:

Argument	Description
IP address	Virtual IP address of the Nutanix cluster
port	The port for the Nutanix cluster

11. Configure the URL for refreshing the virtual network information on the switch:

```
Switch(config-vdm)# refresh-vnet-url <URL>
```

where:

Argument	Description
URL	The URL to get the list of virtual networks

This is often:

```
Switch(config-vdm)# refresh-vnet-url
https://<IP address>:<port>/api/nutanix/v3/networks/list
```

where:

Argument	Description
IP address	Virtual IP address of the Nutanix cluster
port	The port for the Nutanix cluster

12. Configure the Virtual Machine information QoS policy:

```
Switch(config-vdm)# vm {name "<VM name>" | uuid <VM UUID>} vnic {vlan
<VID> | ip <IP address> | mac <MAC address>} attach qos-policy <policy name> in
```

where:

Parameter	Description
VM name	The Virtual Machine name, in quotes.
UUID	The VM UUID.
VID	The VLAN ID.
IP address	The IP address.

Parameter	Description
MAC address	The MAC address.
policy name	The name of the QoS policy.

13. The switch needs to query the topology discovery URL on each VM to get the switch, port, and VLAN information pertaining to each VM. Configure the topology discovery URL:

```
Switch(config-vdm)# topo-discovery-url <server URL> query-delay <seconds>
```

where:

Argument	Description
server URL	The URL of the Nutanix VDM
seconds	The query delay time, in seconds; an integer from 1-60. Default value: 5.

For example, to query a topology server URL with a Nutanix cluster at 10.20.30.40 with a query delay time of 10 seconds, the command would be:

```
Switch(config-vdm)# topo-discovery-url
https://10.20.30.40:9440/PrismGateway/services/rest/v1/vms query-delay 10
```

14. Configure the Virtual Machine information queueing policy:

```
Switch(config-vdm)# vm {name "<VM name>" | uuid <VM UUID>} vnic {vlan
<VID> | ip <IP address> | mac <MAC address>} attach queueing-policy <policy name>
in
```

where:

Parameter	Description
VM name	The Virtual Machine name, in quotes.
UUID	The VM UUID.
VID	The VLAN ID.
IP address	The IP address.
MAC address	The MAC address.
policy name	The name of the queueing policy.

15. To get updated LLDP information, you need to wait at least until the vLAG startup-delay expires before performing the LLDP query so that the query response can reflect the UP ports. This is called the *topology startup delay*. To configure the topology startup delay:

```
Switch(config-vdm)# topo-startup-delay <seconds>
```

where:

Argument	Description
seconds	The topology startup delay time, in seconds; an integer from 1-300. Default value: 90. **Note:** This must be higher than the vLAG startup delay.

16. Subscribe to the Nutanix cluster for virtual machine events.

- via HTTPS (default)::

```
Switch(config-vdm)# subscribe server-url <server URL> client-url <client
URL> vm
```

- via HTTP::

```
Switch(config-vdm)# subscribe server-url
https://10.20.30.40:9940/api/nutanix/v3/webhooks client-url
http://10.20.30.50:8090/nos/api/cfg/nutanix/vm vm
```

where:

Argument	Description
server URL	The URL for subscribing to Nutanix webhooks.
client URL	The URL on the switch for receiving virtual machine events.

For example, given a Nutanix cluster at 10.20.30.40 and switch IP 10.20.30.50, the command might be:

```
Switch(config-vdm)# subscribe server-url
https://10.20.30.40:9440/api/nutanix/v3/webhooks client-url
https://10.20.30.50:443/nos/api/cfg/nutanix/vm vm
```

Note: Both HTTP and HTTPS are supported for the client URL.

17. Subscribe to the Nutanix cluster for virtual network events.

- via HTTPS (default)::

```
Switch(config-vdm)# subscribe server-url <server URL> client-url <client
URL> vmnetwork
```

- via HTTP::

```
Switch(config-vdm)# subscribe server-url
https://10.20.30.40:9940/api/nutanix/v3/webhooks client-url
http://10.20.30.50:8090/nos/api/cfg/nutanix/vmnetwork vmnetwork
```

where:

Argument	Description
server URL	The URL for subscribing to Nutanix webhooks
client URL	The URL on the switch for receiving virtual network events

For example, given a Nutanix cluster at 10.20.30.40 and switch IP 10.20.30.50, the command might be:

```
Switch(config-vdm)# subscribe server-url
https://10.20.30.40:9440/api/nutanix/v3/webhooks client-url
https://10.20.30.50:443/nos/api/cfg/nutanix/network vnetwork
```

Note: Both HTTP and HTTPS are supported for the server URL.

To display the running VDM configuration, enter:

```
Switch(config-vdm)# display running-config vdm
```

Note: Policies are not in effect for traffic local to the hypervisor. When the source and destination MAC or IP addresses reside on the same host, the traffic will not come out of the internal virtual switch of Nutanix.

Viewing Virtual Domain Information

To display virtual domain manager information, enter:

```
Switch(config-vdm)# display vdm information [<VDM name>]
```

where:

Parameter	Description
VDM name	(Optional) The name of the virtual domain manager

To display virtual network information, enter:

```
Switch(config-vdm)# display vnetworks [uuid <network UUID>]
```

where:

Parameter	Description
network UUID	The Universal Unique Identifier (UUID) of the virtual machine

To display virtual machine information, enter:

```
Switch(config-vdm)# display virtual-machine information [interface
Ethernet <slot>/<port> | port-aggregation <aggregation ID>] [vm <VM-UUID>]
```

where:

Parameter	Description
slot	The ethernet slot
port	The ethernet port
aggregation ID	The port aggregation ID
VM-UUID	The Universal Unique Identifier (UUID) of the virtual machine

To display the ACL policy, enter:

```
Switch# display virtual-machine security-policy information [vm {name
"<VM name>" | uuid <VM UUID>}]
```

To display the QoS policy, enter:

```
Switch# display virtual-machine qos-policy information [vm {name "<VM
name>" | uuid <VM-UUID>}]
```

To display the queueing policy, enter:

```
Switch# display virtual-machine queueing-policy information [vm {name
<VM name> | uuid <VM-UUID>}]
```

Unsubscribing to Nutanix VDM Notifications

To stop receiving notifications of events from the Nutanix VDM:

1. To unsubscribe to the Nutanix cluster for virtual machine events, enter::

```
Switch(config-vdm)# unsubscribe server-url <server URL> client-url <client URL>
vm
```

where:

Argument	Description
server URL	The URL for subscribing to Nutanix webhooks
client URL	The URL on the switch for receiving virtual machine events

2. To unsubscribe to the Nutanix cluster for virtual network events:

```
Switch(config-vdm)# unsubscribe server-url <server URL> client-url <client
URL> vnetwork
```

where:

Argument	Description
server URL	The URL for subscribing to Nutanix webhooks
client URL	The URL on the switch for receiving virtual network events

Dynamic VLANs and the VDM

Dynamic VLANs are VLANs provisioned by the VDM when it attaches the Network Interface Card to powered ON VMs.

Dynamic VLAN Considerations

Note the following guidelines for using the Dynamic VLAN feature with the VDM:

● The main intention of the dynamic VLAN feature is to have VLAN provisioning controlled by the VDM and not by the switch administrator. This is one of the reasons why we recommend setting the switch-port mode to trunk plus allowed VLAN all on trunk ports.

● The dynamic VLAN works exactly like the static VLAN in terms of functionality and protocol interactions.

● Turning off provisioning will remove all dynamic VLANs.

● A static VLAN that is created and is auto-provisioned is called a *hybrid VLAN*. Turning off provisioning will cause a hybrid VLAN to revert to a static VLAN.

● You can add non-auto-policy ports can to a dynamic VLAN. The dynamic VLAN will continue to operate as dynamic and will be responsive to VM events and will not be converted to a static or hybrid VLAN.

● Any configuration done on the dynamic VLAN is automatically removed when the dynamic VLAN is removed, whether by configuration or by not being provisioned any more.

● The vLAG consistency check feature is disabled for dynamic VLANs. If a vLAG consistency check is triggered for a dynamic VLAN, there will be a consistency check fail before the other switch gets the same event. This will cause the vLAG ports to be disabled and will lead to the VDM migrating the associated VMs. A VNIC attached to a VM on a particular host will cause Virtual Network Interface Cards (NICs) to attach to VMs on both the switches. Since the same Dynamic VLAN will get configured on both the switches, there is no need for a vLAG consistency check here.

Dynamic VLAN Commands

To view all VLANs, use the command:

```
Switch(config-vdm)# display vlan
```

To view the total number of dynamic VLANs, use the command:

```
Switch(config-vdm)# display vlan summary
```

To delete a VLAN from the switch, whether the VLAN is static, dynamic, or hybrid, use the command:

```
Switch(config-vdm)# no vlan <VLAN ID>
```

where:

Argument	Description
VLAN ID	The ID of the VLAN being deleted

Part 8: Monitoring

The ability to monitor traffic passing through the switch can be invaluable for troubleshooting some types of networking problems. This section covers the following monitoring features:

Chapter 34. Port Mirroring

The Lenovo Cloud Network Operating System port mirroring feature allows you to mirror (copy) the packets of a target port and forward them to a monitoring port. Port mirroring functions for all layer 2 and layer 3 traffic on a port. This feature can be used as a troubleshooting tool or to enhance the security of your network. For example, an IDS server or other traffic sniffer device or analyzer can be connected to the monitoring port to detect intruders attacking the network.
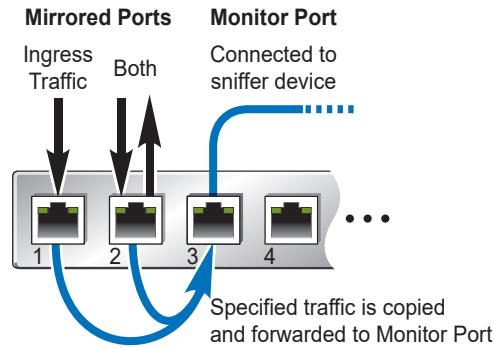
The following topics are discussed in this section:

Port Mirroring Overview

The switch supports a "many to one" mirroring model. As shown in Figure 45, selected traffic for ports 1 and 2 is being monitored by port 3. In the example, both ingress traffic and egress traffic on port 2 are copied and forwarded to the monitor. However, port 1 mirroring is configured so that only ingress traffic is copied and forwarded to the monitor. A device attached to port 3 can analyze the resulting mirrored traffic.

Figure 45. Mirroring Ports

The switch supports four monitor sessions for mirroring rx or tx traffic and two monitor sessions for mirroring both rx and tx traffic. Each monitor port can receive mirrored traffic from any number of target ports.

Cloud NOS does not support "one to many" or "many to many" mirroring models where traffic from a specific port traffic is copied to multiple monitor ports. For example, port 1 traffic cannot be monitored by both port 3 and 4 at the same time, nor can port 2 ingress traffic be monitored by a different port than its egress traffic.

Ingress and egress traffic is duplicated and sent to the monitor port before processing.

SPAN Configuration

The Switched Port Analyzer (SPAN) feature selects network traffic for analysis by a network analyzer. The network analyzer can be any Remote Monitoring mechanism with the capability to perform packet captures.

The switch supports up to a maximum of 18 SPAN sessions.

Note: Only two SPAN sessions can be simultaneously active.

SPAN feature is applicable for the following types of ports:

- Ethernet ports (but not sub-interfaces);
- Link Aggregation Groups (LAGs).

Sources

SPAN sources refer to the interfaces from which traffic can be monitored. The Cloud NOS supports Ethernet and LAGs as SPAN sources. The SPAN traffic can be performed in the ingress direction, the egress direction or both directions:

- Ingress source (Rx)—Traffic entering the device through this source port is copied to the SPAN destination port.
- Egress source (Tx)—Traffic exiting the device through this source port is copied to the SPAN destination port.

The command to add or remove a source interface for the current SPAN session is as follows:

```
Switch(config)# monitor session {<1-18>|all>
Switch(config-monitor)# [no] source interface {ethernet <chassis number/port
number>|port-aggregation <LAG number>} {both|rx|tx}
```

Destinations

SPAN destinations refer to the interfaces that monitors source ports.

The command to add or remove a destination interface for the current SPAN session is as follows:

```
Switch(config)# monitor session {<1-18>|all>
Switch(config-monitor)# [no] destination interface {ethernet <chassis
number/port number>|port-aggregation <LAG number>}
```

Sessions

Lenovo N/OS supports SPAN sessions to designate sources and destinations to monitor.

The command to add or remove a SPAN session is as follows:

```
Switch(config)# [no] monitor session {<1-18>|all}
```

Configuration Example

Follow this procedure to configure SPAN feature:

1. Enter the monitor configuration mode.

```
Switch(config)# monitor session 3
Switch(config-monitor)
```

2. Configure sources and the traffic direction in which to copy packets. Source interface is Ethernet 1/1 and both incoming and outgoing traffic is mirrored.

```
Switch(config-monitor)# source interface ethernet 1/1 both
```

3. Configure a destination for copied source packets.

```
Switch(config-monitor)# destination interface ethernet 1/3
```

4. Optionally, configure a description for the session. By default, no description is defined.

```
Switch(config-monitor)# description span_session_3
```

5. Enable the SPAN session. By default, the session is created in the down state.

```
Switch(config-monitor)# no shutdown
```

6. Display the SPAN configuration.

```
Switch# display monitor session 3
```

To clear the configuration of the specified SPAN session, use the following command:

```
Switch(config)# no monitor session 3
```

ERSPAN Configuration

Lenovo N/OS supports the Encapsulated Remote Switching Port Analyser (ERSPAN) feature on both source and destination ports. ERSPAN transports mirrored traffic over an IP network. The traffic is encapsulated using generic routing encapsulation (GRE) at the source router and is transferred across the network. The packet is decapsulated at the destination router and then sent to the destination interface.

The switch supports up to a maximum of 18 ERSPAN sessions.

Note: Only two ERSPAN sessions can be simultaneously active.

ERSPAN feature is applicable for the following types of ports:

- Ethernet ports (but not sub-interfaces);

- LAGs.

Session Types

Lenovo N/OS supports two types of ERSPAN session: source sessions and destination sessions.

ERSPAN source sessions are created on a switch for which traffic needs to be monitored from a remote device across an IP network. The ERSPAN source session mirrors traffic from an interface, encapsulates that traffic and sends it across the network to the specified IPv4 address.

ERSPAN destination sessions are created on a switch to receive mirrored traffic from another switch (on which an ERSPAN source session was created) and then forward that traffic to a monitoring device for analysis. The ERSPAN destination session receives GRE encapsulated packets from an IPv4 address, decapsulates the packets and sends them to a monitoring device reachable through one of its interfaces.

ERSPAN source sessions can be configured to directly send encapsulated GRE packets to a monitoring device, if that device is reachable. If the target host is not reachable from the source switch, an ERSPAN destination session must be created on the switch to which the monitoring device is directly connected.

Creating an ERSPAN destination session is also required when the mirrored traffic needs to be received by the monitoring device without GRE/ERSPAN headers. The ERSPAN destination switch will decapsulate the packets and then forward them to the monitoring device in the same format in which they we're received on the ERSPAN source switch.

The command to select an ERSPAN session type is as follows:

```
Switch(config)# [no] monitor session <1-18> type {erspan-destination|
erspan-source}
```

To configure the global origin IPv4 address of ethernet ERSPAN sessions, use the following command:

```
Switch(config)# monitor erspan origin ip-address <IPv4 address> global
```

Chapter 34: Port Mirroring **701**

Sources

For ERSPAN source sessions, sources refer to the interfaces from which traffic can be monitored. The Cloud NOS supports ethernet ports and LAGs as ERSPAN sources. The ERSPAN traffic can be performed in the ingress direction, the egress direction or both directions:

- Ingress source (Rx)—Traffic entering the device through this source port is sent to the ERSPAN destination IP address.

- Egress source (Tx)—Traffic exiting the device through this source port is sent to the ERSPAN destination IP address.

In an ERSPAN source session, there is a source interface and a destination IPv4 address. Use the following commands to add or remove a source interface for the current ERSPAN source session:

```
Switch(config)# monitor session <1-18> type erspan-source
Switch(config-erspan-src)# [no] source interface {ethernet <chassis number>/
/<port number>|port-aggregation <LAG number>} [both|rx|tx]
```

For ERSPAN source sessions, destinations refer to the IPv4 addresses of the hosts that monitor the mirrored traffic.

In an ERSPAN source session, there is a source interface and a destination IPv4 address. Use the following commands to configure the destination IPv4 address for the current ERSPAN source session:

```
Switch(config)# monitor session <1-18> type erspan-source
Switch(config-erspan-dst)# destination ip <IP address>
```

For a detailed list of the ERSPAN source configuring commands, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

Destinations

For an ERSPAN destination session, there is a source IPv4 address and a destination interface. Use the following commands to add or remove a source IPv4 address for the current ERSPAN destination session:

```
Switch(config)# monitor session <1-18> type erspan-destination
Switch(config-erspan-dst)# [no] source ip <IP address>
```

For an ERSPAN destination session, there is a source IPv4 address and a destination interface. Use the following commands to configure the destination interface for the current ERSPAN destination session:

```
Switch(config)# monitor session <1-18> type erspan-destination
Switch(config-erspan-dst)# destination interface {ethernet <chassis number/
/port number>|port-aggregation <LAG number>}
```

For a detailed list of the ERSPAN destination configuring commands, see the *Lenovo Network Command Reference for Lenovo Cloud Network Operating System 10.6*.

ERSPAN Source Session Configuration Example

Follow this procedure to configure an ERSPAN source session.

1. Configure the ERSPAN global origin IP address.

```
Switch(config)# monitor erspan origin ip-address 10.1.1.1 global
```

2. Configure a monitor session of type ERSPAN source.

```
Switch(config)# monitor session 3 type erspan-source
Switch(config-erspan-src)#
```

3. Optionally, configure a description for the session. By default, no description is defined.

```
Switch(config-erspan-src)# description erspan_source_session_3
```

4. Configure the source interface and the traffic direction in which to copy packets.

```
Switch(config-erspan-src)# source interface port-aggregation 2 both
```

5. Configure the destination IP address (to which the traffic is mirrored) in the ERSPAN session.

Only one destination IP address is supported per ERSPAN source session.

```
Switch(config-erspan-src)# destination ip 10.2.0.0
```

6. Enable the ERSPAN session. By default, the session is created in the shutdown state.

```
Switch(config-erspan-src)# no shutdown
```

7. Verify the ERSPAN configuration.

```
Switch# display monitor session 3
```

ERSPAN Destination Session Configuration Example

Follow this procedure to configure an ERSPAN destination session.

1. Configure a monitor session of type ERSPAN source.

```
Switch(config)# monitor session 4 type erspan-destination
Switch(config-erspan-dst)#
```

2. Optionally, configure a description for the session. By default, no description is defined.

```
Switch(config-erspan-dst)# description erspan_destination_session_4
```

3. Configure the source IPv4 address.

```
Switch(config-erspan-dst)# source ip 10.2.0.0
```

4. Configure the destination interface (to which the traffic is forwarded for monitoring) in the ERSPAN session.

```
Switch(config-erspan-dst)# destination interface ethernet 1/23
```

5. Enable the ERSPAN session. By default, the session is created in the shutdown state.

```
Switch(config-erspan-dst)# no shutdown
```

6. Verify the ERSPAN configuration.

```
Switch# display monitor session 4
```

Limitations

- Entering additional monitor session commands does not clear previously configured SPAN parameters. You must enter the **no monitor session** command to clear configured SPAN parameters.

- When you specify sources and do not specify a traffic direction (ingress, egress, or both), `both` is used by default.

- For LAGs, only the first 8 members are supported, if that LAG is configured as a sniffer port.

- Up to two ERSPAN sessions are allowed to run simultaneously on the switch.

Chapter 35. Sampled Flow

Sampled Flow (sFlow) a technology for monitoring traffic in networks containing switches and routers. The sFlow monitoring system consists of an sFlow Agent and a central sFlow Collector. The sFlow architecture and sampling techniques were designed for providing continuous site-wide and enterprise-wide traffic monitoring of high speed switched and routed networks.

CNOS supports sFlow version 5 technology for monitoring traffic in data networks using an embedded sFlow agent that can be configured to provide continuous monitoring information of traffic to a central sFlow analyzer.

The switch is responsible only for forwarding sFlow information. A separate sFlow analyzer is required elsewhere on the network to interpret sFlow data.

The following topics are discussed in this chapter:

Note: CNOS only supports sending sFlow datagrams to collectors with IPv4.

Configuring sFlow

You can configure the rate at which sFlow samples network traffic, the size of packets, the polling interval, and the collector address.

To enable or disable the sFlow feature, use the command:

```
Switch(config)# [no] feature sflow
```

To enable or disable sFlow on a specific interface, enter:

```
Switch(config-if)# [no] sflow enable
```

To set the sFlow analyzer IP address, enter:

```
Switch(config-if)# sflow collector ip <IPv4 address> [port <port>] [vrf <vrf>]
```

where:

Parameter	Description
IPv4 address	The IP address of the sFlow analyzer.
port	(Optional) The IP port of the sflow analyzer; default value is 6343.
vrf	(Optional) The VRF; default value is data.

To view the sFlow configuration, enter:

```
Switch# display sflow
```

sFlow Network Polling

You can configure the switch to send network statistics to an sFlow analyzer at regular intervals. When polling is enabled, at the end of each configured polling interval, the switch reports general port statistics and port Ethernet statistics.

To enable sFlow polling and set the polling interval, in seconds, enter:

```
Switch(config)# sflow polling-interval <seconds>
```

where *seconds* is an integer from 0-86400. The default interval is 60. Setting *seconds* to 0 disables polling.

To set the maximum size of an sFlow datagram, enter:

```
Switch(config)# sflow max-datagram-size <size>
```

where *size* is an integer from 200-9000. The default maximum datagram size is 1500.

sFlow Network Sampling

In addition to statistical counters, the switch can be configured to collect periodic samples of the traffic data received on each port. For each sample, 128 bytes are copied, UDP-encapsulated, and sent to the configured sFlow analyzer.

For all ports, the sFlow sampling rate can be configured to occur once each 4096 to 1000000000 packets with a default value of 4096. A sampling rate of 4096 means that one sample will be taken for approximately every 4096 packets received on the port. The sampling rate is statistical, however. It is possible to have slightly more or fewer samples sent to the analyzer for any specific group of packets (especially under low traffic conditions). The actual sample rate becomes most accurate over time, and under higher traffic flow.

Note: The sampling mechanism is provided by the ASIC. It can be configured on a per-port-basis by specifying a sampling rate. For a sampling rate N, a sample will be sent to the CPU for each N packets. This is, however, a statistical sampling rate. It does not mean that if you have a sampling rate of 128 (one packet out of 128), a packet will be sampled after exact 128 packets. It is possible to have two or three or no samples sent to the CPU for a burst of 128 packets. The higher the traffic, the more accurate the sampling mechanism.

To set the sFlow sampling rate, enter:

```
Switch(config)# sflow sampling-rate <rate>
```

where *rate* is an integer from 4096-1000000000. The default rate is 4096.

To set the maximum size of sFlow packets to be sampled, enter:

```
Switch(config)# sflow max-sampled-size <size>
```

where *size* is an integer from 64-256. The default maximum sampled size is 128.

sFlow Example Configuration

1. Enable the sFlow feature:

```
Switch(config)# feature sflow
```

2. Specify the location where the sFlow information will be sent:

```
Switch(config)# sflow collector <IPv4 address>   (sFlow server address)
```

By default, the switch uses established sFlow service port 6343.

3. On a per-port basis, enable sFlow:

```
Switch(config)# interface ethernet <chassis number>/<port number>
Switch(config-if)# sflow enable
Switch(config-if)# exit
```

4. Set the statistics polling rate:

```
Switch(config)# sflow polling-interval <polling rate>   (Statistics polling rate)
```

Specify a polling rate between 1 and 86400 seconds, or 0 to disable. By default, polling is set to 60 for each port.

5. Set the data sampling rate:

```
Switch(config)# sflow sampling-rate <sampling rate>     (Data sampling rate)
```

Specify a sampling rate between 4096 and 1000000000 packets. By default, the sampling rate is 4096 for each port.

6. Save the configuration.

7. Display sFlow statistics:

```
Switch(config)# display sflow statistics
```

8. Clear sFlow statistics:

```
Switch(config)# remove sflow statistics
```

Part 9: Appendices

This section discusses the following topics:

-
-

Appendix A. Getting help and technical assistance

If you need help, service, or technical assistance or just want more information about Lenovo products, you will find a wide variety of sources available from Lenovo to assist you.

Use this information to obtain additional information about Lenovo and Lenovo products, and determine what to do if you experience a problem with your Lenovo system or optional device.

Note: This section includes references to IBM web sites and information about obtaining service. IBM is Lenovo's preferred service provider for the System x, Flex System, and NeXtScale System products.

Before you call, make sure that you have taken these steps to try to solve the problem yourself.

If you believe that you require warranty service for your Lenovo product, the service technicians will be able to assist you more efficiently if you prepare before you call.

- Check all cables to make sure that they are connected.

- Check the power switches to make sure that the system and any optional devices are turned on.

- Check for updated software, firmware, and operating-system device drivers for your Lenovo product. The Lenovo Warranty terms and conditions state that you, the owner of the Lenovo product, are responsible for maintaining and updating all software and firmware for the product (unless it is covered by an additional maintenance contract). Your service technician will request that you upgrade your software and firmware if the problem has a documented solution within a software upgrade.

- If you have installed new hardware or software in your environment, check the IBM ServerProven website to make sure that the hardware and software is supported by your product.

- Go to the IBM Support portal to check for information to help you solve the problem.

- Gather the following information to provide to the service technician. This data will help the service technician quickly provide a solution to your problem and ensure that you receive the level of service for which you might have contracted.

  - Hardware and Software Maintenance agreement contract numbers, if applicable
  - Machine type number (if applicable–Lenovo 4-digit machine identifier)
  - Model number
  - Serial number
  - Current system UEFI and firmware levels
  - Other pertinent information such as error messages and logs

- Start the process of determining a solution to your problem by making the pertinent information available to the service technicians. The IBM service technicians can start working on your solution as soon as you have completed and submitted an Electronic Service Request.

You can solve many problems without outside assistance by following the troubleshooting procedures that Lenovo provides in the online help or in the Lenovo product documentation. The Lenovo product documentation also describes the diagnostic tests that you can perform. The documentation for most systems, operating systems, and programs contains troubleshooting procedures and explanations of error messages and error codes. If you suspect a software problem, see the documentation for the operating system or program.

Appendix B. Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area.

Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
1009 Think Place - Building One
Morrisville, NC 27560
U.S.A.

Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties.

Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Trademarks

Lenovo, the Lenovo logo, Flex System, System x, NeXtScale System, and X-Architecture are trademarks of Lenovo in the United States, other countries, or both.

Intel and Intel Xeon are trademarks of Intel Corporation in the United States, other countries, or both.

Internet Explorer, Microsoft, and Windows are trademarks of the Microsoft group of companies.

Linux is a registered trademark of Linus Torvalds.

Other company, product, or service names may be trademarks or service marks of others.

Important Notes

Processor speed indicates the internal clock speed of the microprocessor; other factors also affect application performance.

CD or DVD drive speed is the variable read rate. Actual speeds vary and are often less than the possible maximum.

When referring to processor storage, real and virtual storage, or channel volume, KB stands for 1,024 bytes, MB stands for 1,048,576 bytes and GB stands for 1,073,741,824 bytes.

When referring to hard disk drive capacity or communications volume, MB stands for 1,000,000 bytes and GB stands for 1,000,000,000 bytes. Total user-accessible capacity can vary depending on operating environments.

Maximum internal hard disk drive capacities assume the replacement of any standard hard disk drives and population of all hard-disk-drive bays with the largest currently supported drives that are available from Lenovo.

Maximum memory might require replacement of the standard memory with an optional memory module.

Each solid-state memory cell has an intrinsic, finite number of write cycles that the cell can incur. Therefore, a solid-state device has a maximum number of write cycles that it can be subjected to, expressed as total bytes written (TBW). A device that has exceeded this limit might fail to respond to system-generated commands or might be incapable of being written to. Lenovo is not responsible for replacement of a device that has exceeded its maximum guaranteed number of program/erase cycles, as documented in the Official Published Specifications for the device.

Lenovo makes no representations or warranties with respect to non-Lenovo products. Support (if any) for the non-Lenovo products is provided by the third party, not Lenovo.

Some software might differ from its retail version (if available) and might not include user manuals or all program functionality.

Recycling Information

Lenovo encourages owners of information technology (IT) equipment to responsibly recycle their equipment when it is no longer needed. Lenovo offers a variety of programs and services to assist equipment owners in recycling their IT products. For information on recycling Lenovo products, go to:

http://www.lenovo.com/recycling

Particulate Contamination

Attention: Airborne particulates (including metal flakes or particles) and reactive gases acting alone or in combination with other environmental factors such as humidity or temperature might pose a risk to the device that is described in this document.

Risks that are posed by the presence of excessive particulate levels or concentrations of harmful gases include damage that might cause the device to malfunction or cease functioning altogether. This specification sets forth limits for particulates and gases that are intended to avoid such damage. The limits must not be viewed or used as definitive limits, because numerous other factors, such as temperature or moisture content of the air, can influence the impact of particulates or environmental corrosives and gaseous contaminant transfer. In the absence of specific limits that are set forth in this document, you must implement practices that maintain particulate and gas levels that are consistent with the protection of human health and safety. If Lenovo determines that the levels of particulates or gases in your environment have caused damage to the device, Lenovo may condition provision of repair or replacement of devices or parts on implementation of appropriate remedial measures to mitigate such environmental contamination. Implementation of such remedial measures is a customer responsibility..

Contaminant	Limits
Particulate	• The room air must be continuously filtered with 40% atmospheric dust spot efficiency (MERV 9) according to ASHRAE Standard 52.2[1]. • Air that enters a data center must be filtered to 99.97% efficiency or greater, using high-efficiency particulate air (HEPA) filters that meet MIL-STD-282. • The deliquescent relative humidity of the particulate contamination must be more than 60%[2]. • The room must be free of conductive contamination such as zinc whiskers.
Gaseous	• Copper: Class G1 as per ANSI/ISA 71.04-1985[3] • Silver: Corrosion rate of less than 300 Å in 30 days
[1] ASHRAE 52.2-2008 - *Method of Testing General Ventilation Air-Cleaning Devices for Removal Efficiency by Particle Size*. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.	
[2] The deliquescent relative humidity of particulate contamination is the relative humidity at which the dust absorbs enough water to become wet and promote ionic conduction.	
[3] ANSI/ISA-71.04-1985. *Environmental conditions for process measurement and control systems: Airborne contaminants*. Instrument Society of America, Research Triangle Park, North Carolina, U.S.A.	

Telecommunication Regulatory Statement

This product may not be certified in your country for connection by any means whatsoever to interfaces of public telecommunications networks. Further certification may be required by law prior to making any such connection. Contact a Lenovo representative or reseller for any questions.

Electronic Emission Notices

When you attach a monitor to the equipment, you must use the designated monitor cable and any interference suppression devices that are supplied with the monitor.

Federal Communications Commission (FCC) Statement

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Properly shielded and grounded cables and connectors must be used to meet FCC emission limits. Lenovo is not responsible for any radio or television interference caused by using other than recommended cables and connectors or by unauthorized changes or modifications to this equipment. Unauthorized changes or modifications could void the user's authority to operate the equipment.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that might cause undesired operation.

Industry Canada Class A Emission Compliance Statement

This Class A digital apparatus complies with Canadian ICES-003.

Avis de Conformité à la Réglementation d'Industrie Canada

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

Australia and New Zealand Class A Statement

Attention: This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

European Union - Compliance to the Electromagnetic Compatibility Directive

This product is in conformity with the protection requirements of EU Council Directive 2004/108/EC (until April 19, 2016) and EU Council Directive 2014/30/EU (from April 20, 2016) on the approximation of the laws of the Member States relating to electromagnetic compatibility. Lenovo cannot accept responsibility for any failure to satisfy the protection requirements resulting from a non-recommended modification of the product, including the installation of option cards from other manufacturers.

This product has been tested and found to comply with the limits for Class A equipment according to European Standards harmonized in the Directives in compliance. The limits for Class A equipment were derived for commercial and industrial environments to provide reasonable protection against interference with licensed communication equipment.

ϵ Lenovo, Einsteinova 21, 851 01 Bratislava, Slovakia

Warning: This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

Germany Class A Statement

Deutschsprachiger EU Hinweis:

Hinweis für Geräte der Klasse A EU-Richtlinie zur Elektromagnetischen Verträglichkeit

Dieses Produkt entspricht den Schutzanforderungen der EU-Richtlinie 2014/30/EU (früher 2004/108/EC) zur Angleichung der Rechtsvorschriften über die elektromagnetische Verträglichkeit in den EU-Mitgliedsstaaten und hält die Grenzwerte der Klasse A der Norm gemäß Richtlinie.

Um dieses sicherzustellen, sind die Geräte wie in den Handbüchern beschrieben zu installieren und zu betreiben. Des Weiteren dürfen auch nur von der Lenovo empfohlene Kabel angeschlossen werden. Lenovo übernimmt keine Verantwortung für die Einhaltung der Schutzanforderungen, wenn das Produkt ohne Zustimmung der Lenovo verändert bzw. wenn Erweiterungskomponenten von Fremdherstellern ohne Empfehlung der Lenovo gesteckt/eingebaut werden.

Deutschland:

Einhaltung des Gesetzes über die elektromagnetische Verträglichkeit von Betriebsmittein

Dieses Produkt entspricht dem „Gesetz über die elektromagnetische Verträglichkeit von Betriebsmitteln" EMVG (früher „Gesetz über die elektromagnetische Verträglichkeit von Geräten"). Dies ist die Umsetzung der EU-Richtlinie 2014/30/EU (früher 2004/108/EC) in der Bundesrepublik Deutschland.

Zulassungsbescheinigung laut dem Deutschen Gesetz über die elektromagnetische Verträglichkeit von Betriebsmitteln, EMVG vom 20. Juli 2007 (früher Gesetz über die elektromagnetische Verträglichkeit von Geräten), bzw. der EMV EU Richtlinie 2014/30/EU (früher 2004/108/EC), für Geräte der Klasse A.

Dieses Gerät ist berechtigt, in Übereinstimmung mit dem Deutschen EMVG das EG-Konformitätszeichen - CE - zu führen. Verantwortlich für die Konformitätserklärung nach Paragraf 5 des EMVG ist die Lenovo (Deutschland) GmbH, Meitnerstr. 9, D-70563 Stuttgart.

Informationen in Hinsicht EMVG Paragraf 4 Abs. (1) 4:

Das Gerät erfüllt die Schutzanforderungen nach EN 55024 und EN 55022 Klasse A.

Nach der EN 55022: „Dies ist eine Einrichtung der Klasse A. Diese Einrichtung kann im Wohnbereich Funkstörungen verursachen; in diesem Fall kann vom Betreiber verlangt werden, angemessene Maßnahmen durchzuführen und dafür aufzukommen.“

Nach dem EMVG: „Geräte dürfen an Orten, für die sie nicht ausreichend entstört sind, nur mit besonderer Genehmigung des Bundesministers für Post und Telekommunikation oder des Bundesamtes für Post und Telekommunikation betrieben werden. Die Genehmigung wird erteilt, wenn keine elektromagnetischen Störungen zu erwarten sind.“ (Auszug aus dem EMVG, Paragraph 3, Abs. 4). Dieses Genehmigungsverfahrenist nach Paragraph 9 EMVG in Verbindung mit der entsprechenden Kostenverordnung (Amtsblatt 14/93) kostenpflichtig.

Anmerkung: Um die Einhaltung des EMVG sicherzustellen sind die Geräte, wie in den Handbüchern angegeben, zu installieren und zu betreiben.

Japan VCCI Class A Statement

> この装置は、クラス A 情報技術装置です。この装置を家庭環境 で使用する
> と電波妨害を引き起こすことがあります。この場合には使用者が適切な対策
> を講ずるよう要求されることがあります。　　　　VCCI-A

This is a Class A product based on the standard of the Voluntary Control Council for Interference (VCCI). If this equipment is used in a domestic environment, radio interference may occur, in which case the user may be required to take corrective actions.

Japan Electronics and Information Technology Industries Association (JEITA) Statement

高調波ガイドライン適合品

Japan Electronics and Information Technology Industries Association (JEITA) Confirmed Harmonics Guidelines (products less than or equal to 20 A per phase)

高調波ガイドライン準用品

Japan Electronics and Information Technology Industries Association (JEITA) Confirmed Harmonics Guidelines with Modifications (products greater than 20 A per phase).

Korea Communications Commission (KCC) Statement

이 기기는 업무용(A급)으로 전자파적합기기로
서 판매자 또는 사용자는 이 점을 주의하시기
바라며, 가정외의 지역에서 사용하는 것을 목
적으로 합니다.

This is electromagnetic wave compatibility equipment for business (Type A). Sellers and users need to pay attention to it. This is for any areas other than home.

Russia Electromagnetic Interference (EMI) Class A statement

ВНИМАНИЕ! Настоящее изделие относится к классу А.
В жилых помещениях оно может создавать радиопомехи, для
снижения которых необходимы дополнительные меры

People's Republic of China Class A electronic emission statement

中华人民共和国 "A类" 警告声明

声 明
此为A级产品，在生活环境中，该产品可能会造成无线电干扰。在这种情况下，
可能需要用户对其干扰采取切实可行的措施。

Taiwan Class A compliance statement

警告使用者：
這是甲類的資訊產品，在
居住的環境中使用時，可
能會造成射頻干擾，在這
種情況下，使用者會被要
求採取某些適當的對策。

Index

Symbols

[ ] 29

Numerics

802.1p QoS 356
802.1Q VLAN tagging 361
802.1Qaz ETS 361
802.1Qbb PFC 358
802.3x flow control 357, 358

A

Access Control List (ACL) 333
Access Control Lists. *See ACLs.*
accessing the switch
   LDAP authentication 144
   RADIUS authentication 138
   security 125, 137
   TACACS+ 141
ACLs 161, 333
active-active redundancy 548
administrator account 62, 139
advertise flag (DCBX) 368
aggregating routes 468
   example 493
anycast address, IPv6 422
assistance, getting 715
Australia Class A statement 724
authenticating, in OSPF 520
autonomous systems (AS) 513

B

bandwidth allocation 356, 363
Border Gateway Protocol (BGP) 455
   failover configuration 491
   route aggregation 468
   route maps 465
Bridge Protocol Data Unit (BPDU) 279
broadcast domains 217

C

Canada Class A electronic emission statement 724

CEE 353, 355
   802.1p QoS 356
   bandwidth allocation 356
   DCBX 353, 367
   ETS 353, 356, 361
   FCoE 354, 355
   on/off 355
   PFC 353, 357, 358
   priority groups 362
China Class A electronic emission statement 727
CIST 290
Class A electronic emission notice 724
command conventions 29
Command Line Interface 514
configuration rules
   CEE 355
   FCoE 354
configuring
   BGP failover 491
   DCBX 369
   ETS 364
   IP routing 394
   OSPF 524
   PFC 359
contamination, particulate and gaseous 722
Converged Enhanced Ethernet. *See* CEE.

D

Data Center Bridging Capability Exchange. *See* DCBX.
DCBX 353, 367
default gateway 393
   configuration example 395
default password 62, 139
default route
   OSPF 517
Differentiated Services
   see DS
Differentiated Services Code Point (DSCP) 335
downloading software 98
DSCP 335
   and QoS 335
dynamic routing
   definition 384
dynamic VLAN
   commands 694
dynamic VLANs
   and the VDM 694

E

electronic emission Class A notice 724
End user access control
   configuring 130
Enhanced Transmission Selection. *See* ETS.

V