Cisco Nexus 5600 Series NX-OS Layer 2 Switching Configuration Guide, Release 7.x

First Published: 2014-03-15

Last Modified: 2018-05-28

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

The preface contains the following sections:

Audience

This publication is for network administrators who configure and maintain Cisco Nexus devices.

Document Conventions

Note As part of our constant endeavor to remodel our documents to meet our customers' requirements, we have modified the manner in which we document configuration tasks. As a result of this, you may find a deviation in the style used to describe these tasks, with the newly included sections of the document following the new format.

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.

Convention	Description
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Related Documentation for Cisco Nexus 5600 Series NX-OS Software

The entire Cisco NX-OS 5600 Series documentation set is available at the following URL:

http://www.cisco.com/c/en/us/support/switches/nexus-5000-series-switches/tsd-products-support-series-home.html

Release Notes

The release notes are available at the following URL:

http://www.cisco.com/c/en/us/support/switches/nexus-5000-series-switches/products-release-notes-list.html

Configuration Guides

These guides are available at the following URL:

http://www.cisco.com/c/en/us/support/switches/nexus-5000-series-switches/products-installation-and-configuration-guides-list.html

The documents in this category include:

- *Cisco Nexus 5600 Series NX-OS Adapter-FEX Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS FabricPath Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS Fibre Channel over Ethernet Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS Fundamentals Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS Interfaces Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS Layer 2 Switching Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS Multicast Routing Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS Quality of Service Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS SAN Switching Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS Security Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS System Management Configuration Guide*
- *Cisco Nexus 5600 Series NX-OS Unicast Routing Configuration Guide*

Licensing Guide

The *License and Copyright Information for Cisco NX-OS Software* is available at http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_0/nx-os/license_agreement/nx-ossw_lisns.html.

Command References

These guides are available at the following URL:

http://www.cisco.com/c/en/us/support/switches/nexus-5000-series-switches/products-command-reference-list.html

The documents in this category include:

- *Cisco Nexus 5600 Series NX-OS Fabric Extender Command Reference*
- *Cisco Nexus 5600 Series NX-OS FabricPath Command Reference*
- *Cisco Nexus 5600 Series NX-OS Fibre Channel Command Reference*
- *Cisco Nexus 5600 Series NX-OS Fundamentals Command Reference*
- *Cisco Nexus 5600 Series NX-OS Interfaces Command Reference*
- *Cisco Nexus 5600 Series NX-OS Layer 2 Interfaces Command Reference*

- *Cisco Nexus 5600 Series NX-OS Multicast Routing Command Reference*
- *Cisco Nexus 5600 Series NX-OS QoS Command Reference*
- *Cisco Nexus 5600 Series NX-OS Security Command Reference*
- *Cisco Nexus 5600 Series NX-OS System Management Command Reference*
- *Cisco Nexus 5600 Series NX-OS TrustSec Command Reference*
- *Cisco Nexus 5600 Series NX-OS Unicast Routing Command Reference*
- *Cisco Nexus 5600 Series NX-OS Virtual Port Channel Command Reference*

Error and System Messages

The *Cisco Nexus 5600 Series NX-OS System Message Guide* is available at http://www.cisco.com/en/US/docs/switches/datacenter/nexus5500/sw/system_messages/reference/sl_nxos_book.html.

Troubleshooting Guide

The *Cisco Nexus 5600 Series NX-OS Troubleshooting Guide* is available at http://www.cisco.com/c/en/us/support/switches/nexus-5000-series-switches/products-troubleshooting-guides-list.html.

Documentation Feedback

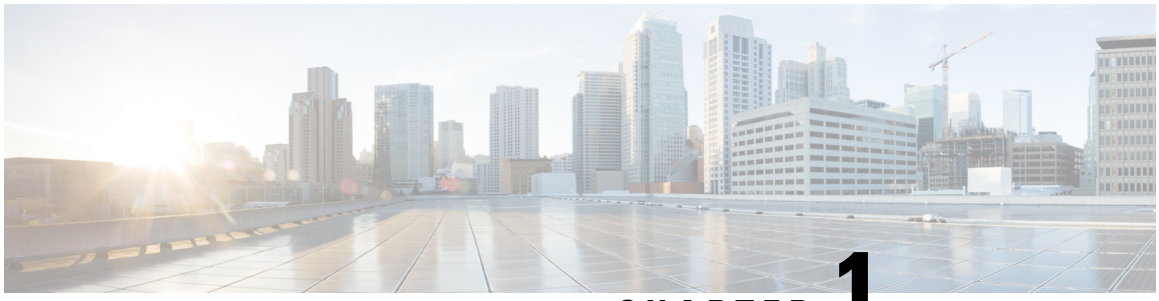
To provide technical feedback on this document, or to report an error or omission, please send your comments to: .

We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the . RSS feeds are a free service.



CHAPTER 1

New and Changed Information

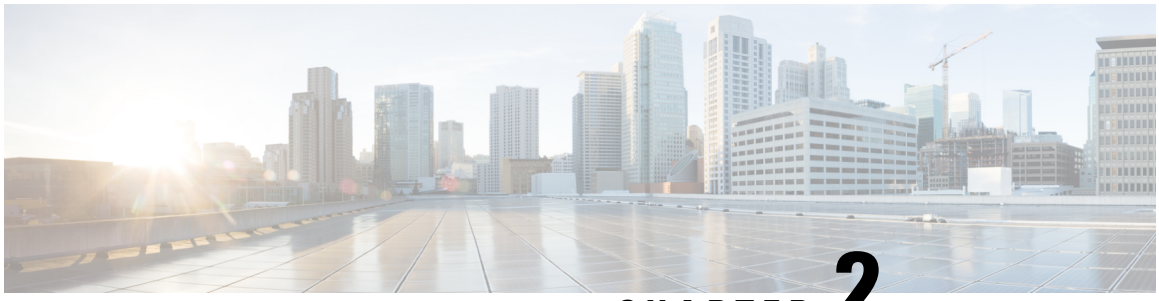
This chapter contains the following sections:

New and Changed Information

The following table provides an overview of the significant changes made to this configuration guide. The table does not provide an exhaustive list of all changes made to this guide or all new features in a particular release.

Cisco NX-OS Release Number	Platform Supported	New/Enhanced Features	Chapter/Topic Where Documented
7.1(0)N1(1)	Cisco Nexus 5500, 5600, and 6000 Series Switches	VLAN Translation	Configuring VLANs
7.3(0)N1(1)	Cisco Nexus 5500, 5600, and 6000 Series Switches	VLAN Long Name support	Configuring VLANs
7.3(2)N1(1)	Cisco Nexus 5500, 5600, and 6000 Series Switches	Added the support to disable autonegotiation in Cisco Nexus 2248PQ 10GE Fabric Extender, Cisco Nexus 2232PP 10GE Fabric Extender, and Cisco Nexus 2348UPQ 10GE Fabric Extender with 1G-based SFP.	Ethernet Interfaces and Disabling Autonegotiation
7.3(2)N1(1)	Cisco Nexus 5500, 5600, and 6000 Series Switches	Added the support for **speed 100** command in Cisco Nexus 2348UPQ 10GE Fabric Extender for ports with 1G Cu SFP GLC-T.	Ethernet Interfaces

Cisco NX-OS Release Number	Platform Supported	New/Enhanced Features	Chapter/Topic Where Documented
7.3(2)N1(1)	Cisco Nexus 5500, 5600, and 6000 Series Switches	Added the support for **speed auto 100** command in Cisco Nexus 2248TP-E Fabric Extender.	Ethernet Interfaces



CHAPTER 2

Overview

This chapter contains the following sections:

Layer 2 Ethernet Switching Overview

The device supports simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

The device assigns a domain (for example, a server) to each device to solve traffic congestion caused by high-bandwidth devices and large number of users.

Because collisions cause significant congestion in Ethernet networks, an effective solution is full-duplex communication. Typically, 10/100-Mbps Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, which is configurable on these interfaces, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, the effective Ethernet bandwidth doubles. 1/10-Gigabit Ethernet operates in full-duplex only.

VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered as a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a bridge or a router.

All ports are assigned to the default VLAN (VLAN1) when the device comes up.

The devices support 4094 VLANs in accordance with the IEEE 802.1Q standard. These VLANs are organized into several ranges, and you use each range slightly differently. Some of these VLANs are reserved for internal use by the device and are not available for configuration.

Note Inter-Switch Link (ISL) trunking is not supported.

Private VLANs

Private VLANs provide traffic separation and security at the Layer 2 level.

A private VLAN is one or more pairs of a primary VLAN and a secondary VLAN, all with the same primary VLAN. The two types of secondary VLANs are isolated and community VLANs. Hosts on isolated VLANs communicate only with hosts in the primary VLAN. Hosts in a community VLAN can communicate only among themselves and with hosts in the primary VLAN but not with hosts in isolated VLANs or in other community VLANs.

Regardless of the combination of isolated and community secondary VLANs, all interfaces within the primary VLAN comprise one Layer 2 domain, and therefore, require only one IP subnet.

Spanning Tree

This section discusses the implementation of the Spanning Tree Protocol (STP). Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. When the IEEE 802.1D Spanning Tree Protocol is referred to in the publication, 802.1D is stated specifically.

STP Overview

STP provides a loop-free network at the Layer 2 level. Layer 2 LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Network devices do not forward these frames but use the frames to construct a loop-free path.

802.1D is the original standard for STP, and many improvements have enhanced the basic loop-free STP. You can create a separate loop-free path for each VLAN, which is named Per VLAN Spanning Tree (PVST+). Additionally, the entire standard was reworked to make the loop-free convergence process faster to keep up with the faster equipment. This STP standard with faster convergence is the 802.1w standard, which is known as Rapid Spanning Tree (RSTP).

Finally, the 802.1s standard, Multiple Spanning Trees (MST), allows you to map multiple VLANs into a single spanning tree instance. Each instance runs an independent spanning tree topology.

Although the software can interoperate with legacy 802.1D systems, the device runs Rapid PVST+ and MST. You can use either Rapid PVST+ or MST in a given VDC; you cannot mix both in one VDC. Rapid PVST+ is the default STP protocol.

Note Cisco NX-OS uses the extended system ID and MAC address reduction; you cannot disable these features.

In addition, Cisco has created some proprietary features to enhance the spanning tree activities.

Rapid PVST+

Rapid PVST+ is the default spanning tree mode for the software and is enabled by default on the default VLAN and all newly created VLANs.

A single instance, or topology, of RSTP runs on each configured VLAN, and each Rapid PVST+ instance on a VLAN has a single root device. You can enable and disable STP on a per-VLAN basis when you are running Rapid PVST+.

MST

The software also supports MST. The multiple independent spanning tree topologies enabled by MST provide multiple forwarding paths for data traffic, enable load balancing, and reduce the number of STP instances required to support a large number of VLANs.

MST incorporates RSTP, so it also allows rapid convergence. MST improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

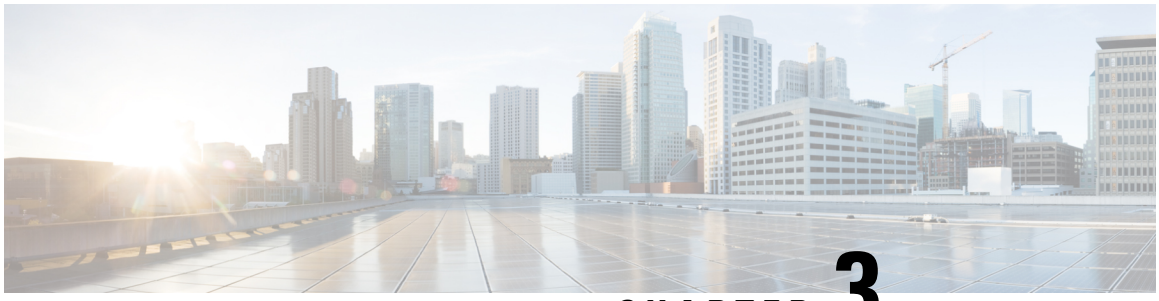
Note Changing the spanning tree mode disrupts the traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

You can force specified interfaces to send prestandard, rather than standard, MST messages using the command-line interface.

STP Extensions

The software supports the following Cisco proprietary features:

- Spanning tree port types—The default spanning tree port type is normal. You can configure interfaces connected to Layer 2 hosts as edge ports and interfaces connected to Layer 2 switches or bridges as network ports.
- Bridge Assurance—Once you configure a port as a network port, Bridge Assurance sends BPDUs on all ports and moves a port into the blocking state if it no longer receives BPDUs. This enhancement is available only when you are running Rapid PVST+ or MST.
- BPDU Guard—BPDU Guard shuts down the port if that port receives a BPDU.
- BPDU Filter—BPDU Filter suppresses sending and receiving BPDUs on the port.
- Loop Guard— Loop guard prevents the occurrence of loop bridging because of unidirectional link failure in a point-to-point link.
- Root Guard— Root guard prevents a port from becoming a root port or a blocked port. If you configure a port with root guard then the port receives a superior BPDU and it immediately goes to root-inconsistent (blocked) state.



CHAPTER 3

Configuring Ethernet Interfaces

This chapter contains the following sections:

Information About Ethernet Interfaces

The Ethernet ports can operate as standard Ethernet interfaces connected to servers or to a LAN.

The Ethernet interfaces are enabled by default.

Interface Command

You can enable the various capabilities of the Ethernet interfaces on a per-interface basis using the **interface** command. When you enter the **interface** command, you specify the following information:

- Interface type—All physical Ethernet interfaces use the **ethernet** keyword.
- Slot number:
  - Slot 1 includes all the fixed ports.
  - Slot 2 includes the ports on the upper expansion module (if populated).
  - Slot 3 includes the ports on the lower expansion module (if populated).
  - Slot 4 includes the ports on the lower expansion module (if populated).
- Port number— Port number within the group.

The interface numbering convention is extended to support use with a Cisco Nexus Fabric Extender as follows:

switch(config)# **interface ethernet** [*chassis*/]*slot*/*port*

- The chassis ID is an optional entry that you can use to address the ports of a connected Fabric Extender. The chassis ID is configured on a physical Ethernet or EtherChannel interface on the switch to identify the Fabric Extender discovered through the interface. The chassis ID ranges from 100 to 199.

Unidirectional Link Detection Parameter

The Cisco-proprietary Unidirectional Link Detection (UDLD) protocol allows ports that are connected through fiber optics or copper (for example, Category 5 cabling) Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. When the switch detects a unidirectional link, UDLD shuts down the affected LAN port and alerts the user. Unidirectional links can cause a variety of problems, including spanning tree topology loops.

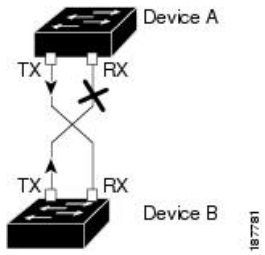
UDLD is a Layer 2 protocol that works with the Layer 1 protocols to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected LAN ports. When you enable both autonegotiation and UDLD, Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, and if autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally at Layer 1, then UDLD at Layer 2 determines whether those fibers are connected correctly and whether traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by autonegotiation, because autonegotiation operates at Layer 1.

A Cisco Nexus device periodically transmits UDLD frames to neighbor devices on LAN ports with UDLD enabled. If the frames are echoed back within a specific time frame and they lack a specific acknowledgment (echo), the link is flagged as unidirectional and the LAN port is shut down. Devices on both ends of the link must support UDLD in order for the protocol to successfully identify and disable unidirectional links.

The following figure shows an example of a unidirectional link condition. Device B successfully receives traffic from Device A on the port. However, Device A does not receive traffic from Device B on the same port. UDLD detects the problem and disables the port.

Figure 1: Unidirectional Link



Default UDLD Configuration

The following table shows the default UDLD configuration.

Table 1: UDLD Default Configuration

Feature	Default Value
UDLD global enable state	Globally disabled
UDLD aggressive mode	Disabled
UDLD per-port enable state for fiber-optic media	Enabled on all Ethernet fiber-optic LAN ports
UDLD per-port enable state for twisted-pair (copper) media	Enabled

UDLD Aggressive and Nonaggressive Modes

UDLD aggressive mode is disabled by default. You can configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. If UDLD aggressive mode is enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving UDLD frames, UDLD tries to reestablish the connection with the neighbor. After eight failed retries, the port is disabled.

To prevent spanning tree loops, nonaggressive UDLD with the default interval of 15 seconds is fast enough to shut down a unidirectional link before a blocking port transitions to the forwarding state (with default spanning tree parameters).

When you enable the UDLD aggressive mode, the following occurs:

- One side of a link has a port stuck (both transmission and receive)
- One side of a link remains up while the other side of the link is down

In these cases, the UDLD aggressive mode disables one of the ports on the link, which prevents traffic from being discarded.

Interface Speed

Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a device discovery protocol that runs over Layer 2 (the data link layer) on all Cisco-manufactured devices (routers, bridges, access servers, and switches) and allows network management applications to discover Cisco devices that are neighbors of already known devices. With CDP, network management applications can learn the device type and the Simple Network Management Protocol (SNMP) agent address of neighboring devices that are running lower-layer, transparent protocols. This feature enables applications to send SNMP queries to neighboring devices.

CDP runs on all media that support Subnetwork Access Protocol (SNAP). Because CDP runs over the data-link layer only, two systems that support different network-layer protocols can learn about each other.

Each CDP-configured device sends periodic messages to a multicast address, advertising at least one address at which it can receive SNMP messages. The advertisements also contain time-to-live, or holdtime information, which is the length of time a receiving device holds CDP information before discarding it. Each device also listens to the messages sent by other devices to learn about neighboring devices.

The switch supports both CDP Version 1 and Version 2.

Default CDP Configuration

The following table shows the default CDP configuration.

Table 2: Default CDP Configuration

Feature	Default Setting
CDP interface state	Enabled
CDP timer (packet update frequency)	60 seconds
CDP holdtime (before discarding)	180 seconds
CDP Version-2 advertisements	Enabled

Error-Disabled State

An interface is in the error-disabled (err-disabled) state when the inteface is enabled administratively (using the **no shutdown** command) but disabled at runtime by any process. For example, if UDLD detects a unidirectional link, the interface is shut down at runtime. However, because the interface is administratively enabled, the interface status displays as err-disabled. Once an interface goes into the err-disabled state, you must manually reenable it or you can configure an automatic timeout recovery value. The err-disabled detection is enabled by default for all causes. The automatic recovery is not configured by default.

When an interface is in the err-disabled state, use the **errdisable detect cause** command to find information about the error.

You can configure the automatic err-disabled recovery timeout for a particular err-disabled cause by changing the time variable.

The **errdisable recovery cause** command provides automatic recovery after 300 seconds. To change the recovery period, use the **errdisable recovery interval** command to specify the timeout period. You can specify 30 to 65535 seconds.

If you do not enable the err-disabled recovery for the cause, the interface stays in the err-disabled state until you enter the **shutdown** and **no shutdown** commands. If the recovery is enabled for a cause, the interface is brought out of the err-disabled state and allowed to retry operation once all the causes have timed out. Use the **show interface status err-disabled** command to display the reason behind the error.

About Port Profiles

You can create a port profile that contains many interface commands and apply that port profile to a range of interfaces on the . Port profiles can be applied to the following interface types:

- Ethernet
- VLAN network interface
- Port channel

A command that is included in a port profile can be configured outside of the port profile. If the new configuration in the port profile conflicts with the configurations that exist outside the port profile, the

commands configured for an interface in configuration terminal mode have higher priority than the commands in the port profile. If changes are made to the interface configuration after a port profile is attached to it, and the configuration conflicts with that in the port profile, the configurations in the interface will be given priority.

You inherit the port profile when you attach the port profile to an interface or range of interfaces, When you attach, or inherit, a port profile to an interface or range of interfaces, the switch applies all the commands in that port profile to the interfaces.

You can have one port profile inherit the settings from another port profile. Inheriting another port profile allows the initial port profile to assume all of the commands of the second, inherited, port profile that do not conflict with the initial port profile. Four levels of inheritance are supported. The same port profile can be inherited by any number of port profiles.

To apply the port profile configurations to the interfaces, you must enable the specific port profile. You can configure and inherit a port profile onto a range of interfaces prior to enabling the port profile; you then enable that port profile for the configurations to take effect on the specified interfaces.

When you remove a port profile from a range of interfaces, the switch undoes the configuration from the interfaces first and then removes the port profile link itself. When you remove a port profile, the switch checks the interface configuration and either skips the port profile commands that have been overridden by directly entered interface commands or returns the command to the default value.

If you want to delete a port profile that has been inherited by other port profiles, you must remove the inheritance before you can delete the port profile.

You can choose a subset of interfaces from which to remove a port profile from among that group of interfaces that you originally applied the profile. For example, if you configured a port profile and configured ten interfaces to inherit that port profile, you can remove the port profile from just some of the specified ten interfaces. The port profile continues to operate on the remaining interfaces to which it is applied.

If you delete a specific configuration for a specified range of interfaces using the interface configuration mode, that configuration is also deleted from the port profile for that range of interfaces only. For example, if you have a channel group inside a port profile and you are in the interface configuration mode and you delete that port channel, the specified port channel is also deleted from the port profile as well.

After you inherit a port profile on an interface or range of interfaces and you delete a specific configuration value, that port profile configuration will not operate on the specified interfaces.

If you attempt to apply a port profile to the wrong type of interface, the switch returns an error.

When you attempt to enable, inherit, or modify a port profile, the switch creates a checkpoint. If the port profile configuration fails, the switch rolls back to the prior configuration and returns an error. A port profile is never only partially applied.

Guidelines and Limitations for Port Profiles

Port profiles have the following configuration guidelines and limitations:

- Each port profile must have a unique name across interface types and the network.
- Commands that you enter under the interface mode take precedence over the port profile's commands if there is a conflict. However, the port profile retains that command in the port profile.
- The port profile's commands take precedence over the default commands on the interface, unless the default command explicitly overrides the port profile command.
- After you inherit a port profile onto an interface or range of interfaces, you can override individual configuration values by entering the new value at the interface configuration level. If you remove the

individual configuration values at the interface configuration level, the interface uses the values in the port profile again.

- There are no default configurations associated with a port profile.
- A subset of commands are available under the port profile configuration mode, depending on which interface type that you specify.
- You cannot use port profiles with Session Manager.

Debounce Timer Parameters

MTU Configuration

The Cisco Nexus device switch does not fragment frames. As a result, the switch cannot have two ports in the same Layer 2 domain with different maximum transmission units (MTUs). A per-physical Ethernet interface MTU is not supported. Instead, the MTU is set according to the QoS classes. You modify the MTU by setting class and policy maps.

Note When you show the interface settings, a default MTU of 1500 is displayed for physical Ethernet interfaces.

Information About Default Interfaces

You can use the default interface feature to clear the configured parameters for both physical and logical interfaces such as the Ethernet, loopback, VLAN network, and the port-channel interface.

The default interface feature allows you to clear the existing configuration of multiple interfaces such as Ethernet, loopback, VLAN network, and port-channel interfaces. All user configuration under a specified interface will be deleted. You can optionally create a checkpoint before clearing the interface configuration so that you can later restore the deleted configuration.

Note The default interfaces feature is supported for management interfaces but is not recommended because the device might be in an unreachable state.

Default Physical Ethernet Settings

The following table lists the default settings for all physical Ethernet interfaces:

Parameter	Default Setting
Debounce	Enable, 100 milliseconds
Duplex	Auto (full-duplex)

Parameter	Default Setting
Encapsulation	ARPA
MTU[1]	1500 bytes
Port Mode	Access
Speed	Auto (10000)

[1] MTU cannot be changed per-physical Ethernet interface. You modify MTU by selecting maps of QoS classes.

Information About Access and Trunk Interfaces

Understanding Access and Trunk Interfaces

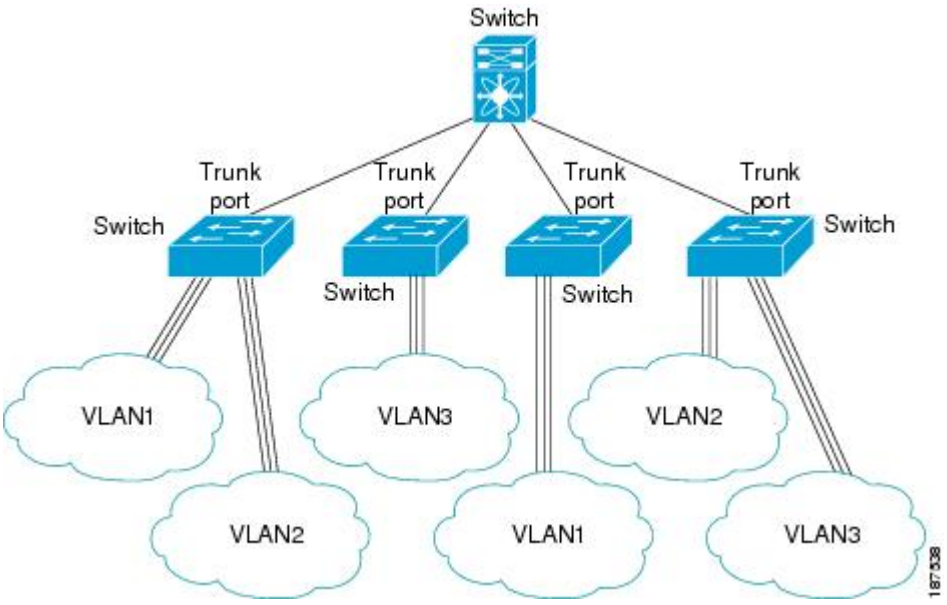
Ethernet interfaces can be configured either as access ports or a trunk ports, as follows:

- An access port can have only one VLAN configured on the interface; it can carry traffic for only one VLAN.
- A trunk port can have two or more VLANs configured on the interface; it can carry traffic for several VLANs simultaneously.

Note Cisco NX-OS supports only IEEE 802.1Q-type VLAN trunk encapsulation.

The following figure shows how you can use trunk ports in the network. The trunk port carries traffic for two or more VLANs.

Figure 2: Devices in a Trunking Environment



In order to correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation or tagging method.

To optimize the performance on access ports, you can configure the port as a host port. Once the port is configured as a host port, it is automatically set as an access port, and channel grouping is disabled. Use the host designation to decrease the time it takes the designated port to begin to forward packets.

Note Only an end station can be set as a host port; you will receive an error message if you attempt to configure other ports as hosts.

If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.

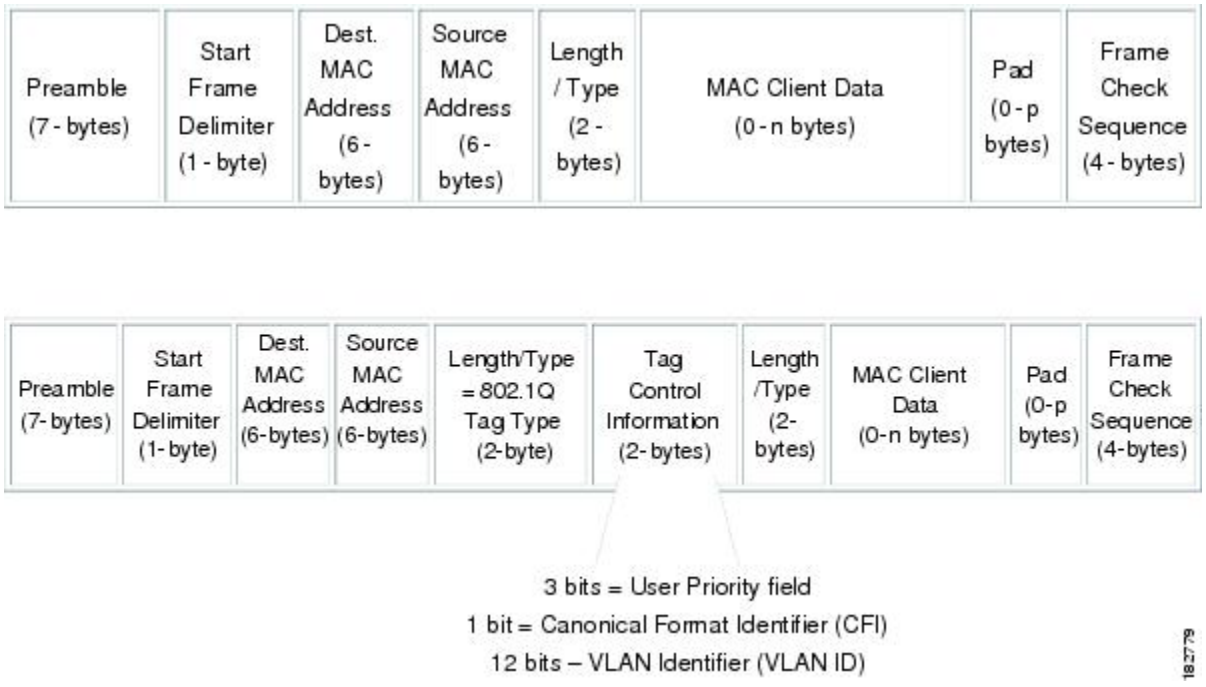
Note An Ethernet interface can function as either an access port or a trunk port; it cannot function as both port types simultaneously.

Understanding IEEE 802.1Q Encapsulation

A trunk is a point-to-point link between the device and another networking device. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

To correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation (tagging) method. This tag carries information about the specific VLAN to which the frame and packet belong. This method allows packets that are encapsulated for several different VLANs to traverse the same port and maintain traffic separation between the VLANs. The encapsulated VLAN tag also allows the trunk to move traffic end-to-end through the network on the same VLAN.

Figure 3: Header Without and With 802.1Q Tag Included



Understanding Access VLANs

When you configure a port in access mode, you can specify which VLAN will carry the traffic for that interface. If you do not configure the VLAN for a port in access mode, or an access port, the interface carries traffic for the default VLAN (VLAN1).

You can change the access port membership in a VLAN by specifying the new VLAN. You must create the VLAN before you can assign it as an access VLAN for an access port. If you change the access VLAN on an access port to a VLAN that is not yet created, the system will shut that access port down.

Note If you change the VLAN on an access port or a trunk port it will flap the interface. However, if the port is part of a vPC, then first change the native VLAN on the secondary vPC, and then to primary vPC.

If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.

Note If you assign an access VLAN that is also a primary VLAN for a private VLAN, all access ports with that access VLAN will also receive all the broadcast traffic for the primary VLAN in the private VLAN mode.

Understanding the Native VLAN ID for Trunk Ports

A trunk port can carry untagged packets simultaneously with the 802.1Q tagged packets. When you assign a default port VLAN ID to the trunk port, all untagged traffic travels on the default port VLAN ID for the trunk port, and all untagged traffic is assumed to belong to this VLAN. This VLAN is referred to as the native VLAN ID for a trunk port. The native VLAN ID is the VLAN that carries untagged traffic on trunk ports.

The trunk port sends an egressing packet with a VLAN that is equal to the default port VLAN ID as untagged; all the other egressing packets are tagged by the trunk port. If you do not configure a native VLAN ID, the trunk port uses the default VLAN.

Note Native VLAN ID numbers *must* match on both ends of the trunk.

Note We recommend that you configure the native VLAN in the trunk allowed VLAN list.

Understanding Allowed VLANs

By default, a trunk port sends traffic to and receives traffic from all VLANs. All VLAN IDs are allowed on each trunk. However, you can remove VLANs from this inclusive list to prevent traffic from the specified VLANs from passing over the trunk. You can add any specific VLANs later that you may want the trunk to carry traffic for back to the list.

To partition the Spanning Tree Protocol (STP) topology for the default VLAN, you can remove VLAN1 from the list of allowed VLANs. Otherwise, VLAN1, which is enabled on all ports by default, will have a very big STP topology, which can result in problems during STP convergence. When you remove VLAN1, all data traffic for VLAN1 on this port is blocked, but the control traffic continues to move on the port.

Understanding Native 802.1Q VLANs

To provide additional security for traffic passing through an 802.1Q trunk port, the **vlan dot1q tag native** command was introduced. This feature provides a means to ensure that all packets going out of a 802.1Q trunk port are tagged and to prevent reception of untagged packets on the 802.1Q trunk port.

Without this feature, all tagged ingress frames received on a 802.1Q trunk port are accepted as long as they fall inside the allowed VLAN list and their tags are preserved. Untagged frames are tagged with the native VLAN ID of the trunk port before further processing. Only those egress frames whose VLAN tags are inside the allowed range for that 802.1Q trunk port are received. If the VLAN tag on a frame happens to match that of the native VLAN on the trunk port, the tag is stripped off and the frame is sent untagged.

This behavior could potentially be exploited to introduce "VLAN hopping" in which a hacker could try and have a frame jump to a different VLAN. It is also possible for traffic to become part of the native VLAN by sending untagged packets into an 802.1Q trunk port.

To address the above issues, the **vlan dot1q tag native** command performs the following functions:

- On the ingress side, all untagged data traffic is dropped.
- On the egress side, all traffic is tagged. If traffic belongs to native VLAN it is tagged with the native VLAN ID.

This feature is supported on all the directly connected Ethernet and Port Channel interfaces. It is also supported on all the host interface ports of any attached Fabric Extender (FEX).

Note You can enable the **vlan dot1q tag native** command by entering the command in the global configuration mode.

Configuring Access and Trunk Interfaces

Configuring a LAN Interface as an Ethernet Access Port

You can configure an Ethernet interface as an access port. An access port transmits packets on only one, untagged VLAN. You specify which VLAN traffic that the interface carries. If you do not specify a VLAN for an access port, the interface carries traffic only on the default VLAN. The default VLAN is VLAN1.

The VLAN must exist before you can specify that VLAN as an access VLAN. The system shuts down an access port that is assigned to an access VLAN that does not exist.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport mode** {**access** | **trunk**}	Sets the interface as a nontrunking nontagged single-VLAN Ethernet interface. An access port can carry traffic in one VLAN only. By default, an access port carries traffic for VLAN1; to set the access port to carry traffic for a different VLAN, use the **switchport access vlan** command.
Step 4	switch(config-if)# **switchport access vlan** *vlan-id*	Specifies the VLAN for which this access port will carry traffic. If you do not enter this command, the access port carries traffic on VLAN1 only; use this command to change the VLAN for which the access port carries traffic.

Example

This example shows how to set an interface as an Ethernet access port that carries traffic for a specific VLAN only:

```
switch# configure terminal
switch(config)# interface ethernet 1/10
```

```
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
```

Configuring Access Host Ports

By using a switchport host, you can make an access port a spanning-tree edge port, and enable BPDU Filtering and BPDU Guard at the same time.

Before you begin

Ensure that you are configuring the correct interface; it must be an interface that is connnected to an end station.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport host**	Sets the interface to spanning-tree port type edge, turns onBPDU Filtering and BPDU Guard. **Note** Apply this command only to switchports that connect to hosts.

Example

This example shows how to set an interface as an Ethernet access host port with EtherChannel disabled:

```
switch# configure terminal
switch(config)# interface ethernet 1/10
switch(config-if)# switchport host
```

Configuring Trunk Ports

You can configure an Ethernet port as a trunk port; a trunk port transmits untagged packets for the native VLAN plus encapsulated, tagged, packets for multiple VLANs.

Note Cisco NX-OS supports only 802.1Q encapsulation.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {*type slot*/*port* | **port-channel** *number*}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport mode** {**access** | **trunk**}	Sets the interface as an Ethernet trunk port. A trunk port can carry traffic in one or more VLANs on the same physical link (VLANs are based on the trunk-allowed VLANs list). By default, a trunk interface can carry traffic for all VLANs. To specify that only certain VLANs are allowed on the specified trunk, use the **switchport trunk allowed vlan** command.

Example

This example shows how to set an interface as an Ethernet trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport mode trunk
```

Configuring the Native VLAN for 802.1Q Trunking Ports

If you do not configure this parameter, the trunk port uses the default VLAN as the native VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {*type slot*/*port* | **port-channel** *number*}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport trunk native vlan** *vlan-id*	Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 4094, except those VLANs reserved for internal use. The default value is VLAN1.

Example

This example shows how to set the native VLAN for an Ethernet trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport trunk native vlan 5
```

Configuring the Allowed VLANs for Trunking Ports

You can specify the IDs for the VLANs that are allowed on the specific trunk port.

Before you configure the allowed VLANs for the specified trunk ports, ensure that you are configuring the correct interfaces and that the interfaces are trunks.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {*type slot*/*port* | **port-channel** *number*}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport trunk allowed vlan** {*vlan-list* **all** | **none** [**add** |except | none | **remove** {*vlan-list*}]}	Sets allowed VLANs for the trunk interface. The default is to allow all VLANs on the trunk interface: 1 to 3967 and 4048 to 4094. VLANs 3968 to 4047 are the default VLANs reserved for internal use by default; this group of VLANs is configurable. By default, all VLANs are allowed on all trunk interfaces. **Note** You cannot add internally allocated VLANs as allowed VLANs on trunk ports. The system returns a message if you attempt to list an internally allocated VLAN as an allowed VLAN.

Example

This example shows how to add VLANs to the list of allowed VLANs on an Ethernet trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport trunk allow vlan 15-20
```

Configuring Native 802.1Q VLANs

Typically, you configure 802.1Q trunks with a native VLAN ID, which strips tagging from all packets on that VLAN. This configuration allows all untagged traffic and control traffic to transit the Cisco Nexus device. Packets that enter the switch with 802.1Q tags that match the native VLAN ID value are similarly stripped of tagging.

To maintain the tagging on the native VLAN and drop untagged traffic, enter the **vlan dot1q tag native** command. The switch will tag the traffic received on the native VLAN and admit only 802.1Q-tagged frames, dropping any untagged traffic, including untagged traffic in the native VLAN.

Control traffic continues to be accepted untagged on the native VLAN on a trunked port, even when the **vlan dot1q tag native** command is enabled.

Note The **vlan dot1q tag native** command is enabled on global basis.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan dot1q tag native** [**tx-only**]	Enables dot1q (IEEE 802.1Q) tagging for all native VLANs on all trunked ports on the Cisco Nexus device. By default, this feature is disabled.
Step 3	(Optional) switch(config)# **no vlan dot1q tag native** [**tx-only**]	Disables dot1q (IEEE 802.1Q) tagging for all native VLANs on all trunked ports on the switch.
Step 4	(Optional) switch# **show vlan dot1q tag native**	Displays the status of tagging on the native VLANs.

Example

This example shows how to enable 802.1Q tagging on the switch:

```
switch# configure terminal
switch(config)# vlan dot1q tag native
switch(config)# exit
switch# show vlan dot1q tag native
vlan dot1q native tag is enabled
```

Verifying the Interface Configuration

Use the following commands to display access and trunk interface configuration information.

Command	Purpose
switch# **show interface**	Displays the interface configuration
switch# **show interface switchport**	Displays information for all Ethernet interfaces, including access and trunk interfaces.
switch# **show interface brief**	Displays interface configuration information.

Configuring Ethernet Interfaces

The section includes the following topics:

Configuring a Layer 3 Interface on a Cisco Nexus Device

On Cisco Nexus devices, you can configure a Layer 3 interface.

You can change a Layer 3 interface into a Layer 2 interface by using the **switchport** command. You can change a Layer 2 interface into a Layer 3 interface by using the **no switchport** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface ethernet** *slot*/*port*	Enters configuration mode for the specified interface. **Note** If this is a QSFP+ GEM or a breakout port, the *port* syntax is *QSFP-module*/*port*.
Step 3	switch(config-if)# **no switchport**	Selects the Layer 3 interface.
Step 4	switch(config-if)# **no shutdown**	Restarts the interface.

Example

This example shows how to configure a Layer 3 interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# no shutdown
```

Configuring the UDLD Mode

You can configure normal or aggressive unidirectional link detection (UDLD) modes for Ethernet interfaces on devices configured to run UDLD. Before you can enable a UDLD mode for an interface, you must make sure that UDLD is already enabled on the device that includes the interface. UDLD must also be enabled on the other linked interface and its device.

To use the normal UDLD mode, you must configure one of the ports for normal mode and configure the other port for the normal or aggressive mode. To use the aggressive UDLD mode, you must configure both ports for the aggressive mode.

Note Before you begin, UDLD must be enabled for the other linked port and its device.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature udld**	Enables UDLD for the device.
Step 3	switch(config)# **no feature udld**	Disables UDLD for the device.
Step 4	switch(config)# **show udld global**	Displays the UDLD status for the device.
Step 5	switch(config)# **interface** *type slot*/*port*	Specifies an interface to configure, and enters interface configuration mode.
Step 6	switch(config-if)# **udld** {**enable** | **disable** | **aggressive**}	Enables the normal UDLD mode, disables UDLD, or enables the aggressive UDLD mode.
Step 7	switch(config-if)# **show udld** *interface*	Displays the UDLD status for the interface.

Example

This example shows how to enable UDLD for the switch:

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

This example shows how to enable the normal UDLD mode for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# udld enable
```

This example shows how to enable the aggressive UDLD mode for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# udld aggressive
```

This example shows how to disable UDLD for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# udld disable
```

This example shows how to disable UDLD for the switch:

```
switch# configure terminal
switch(config)# no feature udld
```

Configuring Interface Speed

Note If the interface and transceiver speed is mismatched, the SFP validation failed message is displayed when you enter the **show interface ethernet** *slot*/*port* command. For example, if you insert a 1-Gigabit SFP transceiver into a port without configuring the speed 1000 command, you will get this error. By default, all ports are 10 Gigabits.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface. This interface must have a 1-Gigabit Ethernet SFP transceiver inserted into it.
Step 3	switch(config-if)# **speed** *speed*	Sets the speed for a physical Ethernet interface.

Example

The following example shows how to set the speed for a 1-Gigabit Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# speed 1000
```

Disabling Link Negotiation

You can disable link negotiation using the **no negotiate auto** command. By default, auto-negotiation is enabled on 1-Gigabit ports and disabled on 10-Gigabit ports and 40-Gigabit ports.

This command is equivalent to the Cisco IOS **speed non-negotiate** command.

Note The auto-negotiation configuration is not applicable on 10-Gigabit or 40-Gigabit Ethernet ports. When auto-negotiation is configured on a 10-Gigabit port or 40-Gigabit port , the following error message is displayed:

```
ERROR: Ethernet1/40: Configuration does not match the port capability
```

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot*/*port*	Selects the interface and enters interface mode.
Step 3	switch(config-if)# **no negotiate auto**	Disables link negotiation on the selected Ethernet interface (1-Gigabit port).
Step 4	(Optional) switch(config-if)# **negotiate auto**	Enables link negotiation on the selected Ethernet interface. The default for 1-Gigabit Ethernet ports is enabled. **Note** This command is not applicable for 10GBASE-T ports. It should not be used on 10-GBASE-T ports.

Example

This example shows how to disable auto-negotiation on a specified Ethernet interface (1-Gigabit port):

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# no negotiate auto
switch(config-if)#
```

This example shows how to enable auto-negotiation on a specified Ethernet interface (1-Gigabit port):

```
switch# configure terminal
switch(config)# interface ethernet 1/5
switch(config-if)# negotiate auto
switch(config-if)#
```

Configuring the CDP Characteristics

You can configure the frequency of Cisco Discovery Protocol (CDP) updates, the amount of time to hold the information before discarding it, and whether or not to send Version-2 advertisements.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# [**no**] **cdp advertise** {**v1** | **v2** }	Configures the version to use to send CDP advertisements. Version-2 is the default state. Use the **no** form of the command to return to its default setting.

	Command or Action	Purpose
Step 3	(Optional) switch(config)# [**no**] **cdp format device-id** {**mac-address** | **serial-number** | **system-name**}	Configures the format of the CDP device ID. The default is the system name, which can be expressed as a fully qualified domain name. Use the **no** form of the command to return to its default setting.
Step 4	(Optional) switch(config)# [**no**] **cdp holdtime** *seconds*	Specifies the amount of time a receiving device should hold the information sent by your device before discarding it. The range is 10 to 255 seconds; the default is 180 seconds. Use the **no** form of the command to return to its default setting.
Step 5	(Optional) switch(config)# [**no**] **cdp timer** *seconds*	Sets the transmission frequency of CDP updates in seconds. The range is 5 to 254; the default is 60 seconds. Use the **no** form of the command to return to its default setting.

Example

This example shows how to configure CDP characteristics:

```
switch# configure terminal
switch(config)# cdp timer 50
switch(config)# cdp holdtime 120
switch(config)# cdp advertise v2
```

Enabling or Disabling CDP

You can enable or disable CDP for Ethernet interfaces. This protocol works only when you have it enabled on both interfaces on the same link.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **cdp enable**	Enables CDP for the interface. To work correctly, this parameter must be enabled for both interfaces on the same link.
Step 4	switch(config-if)# **no cdp enable**	Disables CDP for the interface.

Example

This example shows how to enable CDP for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# cdp enable
```

This command can only be applied to a physical Ethernet interface.

Enabling the Error-Disabled Detection

You can enable error-disable (err-disabled) detection in an application. As a result, when a cause is detected on an interface, the interface is placed in an err-disabled state, which is an operational state that is similar to the link-down state.

Note Base ports in Cisco Nexus 5500 never get error disabled due to pause rate-limit like in the Cisco Nexus 5020 or 5010 switch.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **errdisable detect cause** {*all* | *link-flap* | *loopback*}	Specifies a condition under which to place the interface in an err-disabled state. The default is enabled.
Step 3	switch(config)# **shutdown**	Brings the interface down administratively. To manually recover the interface from the err-disabled state, enter this command first.
Step 4	switch(config)# **no shutdown**	Brings the interface up administratively and enables the interface to recover manually from the err-disabled state.
Step 5	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable the err-disabled detection in all cases:

```
switch# configure terminal
switch(config)# errdisable detect cause all
switch(config)# shutdown
switch(config)# no shutdown
switch(config)# show interface status err-disabled
switch(config)# copy running-config startup-config
```

Enabling the Error-Disabled Recovery

You can specify the application to bring the interface out of the error-disabled (err-disabled) state and retry coming up. It retries after 300 seconds, unless you configure the recovery timer (see the **errdisable recovery interval** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit*}	Specifies a condition under which the interface automatically recovers from the err-disabled state, and the device retries bringing the interface up. The device waits 300 seconds to retry. The default is disabled.
Step 3	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable err-disabled recovery under all conditions:

```
switch# configure terminal
switch(config)# errdisable recovery cause all
switch(config)# show interface status err-disabled
switch(config)# copy running-config startup-config
```

Configuring the Error-Disabled Recovery Interval

You can use this procedure to configure the err-disabled recovery timer value. The range is from 30 to 65535 seconds. The default is 300 seconds.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **errdisable recovery interval** *interval*	Specifies the interval for the interface to recover from the err-disabled state. The range is from 30 to 65535 seconds. The default is 300 seconds.
Step 3	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable err-disabled recovery under all conditions:

```
switch# configure terminal
switch(config)# errdisable recovery interval 32
switch(config)# show interface status err-disabled
switch(config)# copy running-config startup-config
```

Port Profiles

Creating a Port Profile

You can create a port profile on the switch. Each port profile must have a unique name across interface types and the network.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**port-profile** [**type** {**ethernet** | **interface-vlan** | **port channel**}] *name*
	Example:
	`switch(config)# port-profile type ethernet test`
	`switch(config-port-prof)#`
		Creates and names a port profile for the specified type of interface and enters the port profile configuration mode.

Step 3	**exit**
	Example:
	`switch(config-port-prof)# exit`
	`switch(config)#`
		Exits port profile configuration mode.

	Command or Action	Purpose
Step 4	(Optional) **show port-profile** **Example:** `switch(config)# show port-profile name`	Displays the port profile configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to create a port profile named test for Ethernet interfaces:

```
switch# configure terminal
switch(config)# port-profile type ethernet test
switch(config-port-prof)#
```

This example shows how to add the interface commands to a port profile named ppEth configured for Ethernet interfaces:

```
switch# configure terminal
switch(config)# port-profile ppEth
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport trunk allowed vlan 300-400
switch(config-port-prof)# flowcontrol receive on
switch(config-port-prof)# speed 10000
switch(config-port-prof)#
```

Modifying a Port Profile

You can modify a port profile in port-profile configuration mode.

You can remove commands from a port profile using the **no** form of the command. When you remove a command from the port profile, the corresponding command is removed from the interface that is attached to the port profile.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**port-profile** [**type** {**ethernet** | **interface-vlan** | **port channel**}] *name*
	Example:
	`switch(config)# port-profile type ethernet test`
	`switch(config-port-prof)#`
		Enters the port profile configuration mode for the specified port profile and allows you to add or remove configurations to the profile.

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config-port-prof)# exit`
	`switch(config)#`
		Exits the port profile configuration mode.

Step 4	(Optional) **show port-profile** **Example:** `switch(config)# show port-profile name`	Displays the port profile configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to remove commands from the port profile named ppEth configured for an Ethernet interface:

```
switch# configure terminal
switch(config)# port-profile ppEth
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport trunk allowed vlan 300-400
switch(config-port-prof)# flowcontrol receive on
switch(config-port-prof)# no speed 10000
switch(config-port-prof)#
```

Enabling a Specific Port Profile

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**port-profile** [**type** {**ethernet** | **interface-vlan** | **port channel**}] *name*
	Example:
	`switch(config)# port-profile type ethernet test`
	`switch(config-port-prof)# no shutdown`
	`switch(config-port-prof)#`
		Enters the port profile configuration mode for the specified port profile.

Step 3	**state enabled** *name* **Example:**	Enables the port profile.

	Command or Action	Purpose
	`switch(config-port-prof)# state enabled` `switch(config-port-prof)#`	

Step 4	**exit**
	Example:
	`switch(config-port-prof)# exit`
	`switch(config)#`
		Exits the port profile configuration mode.

Step 5	(Optional) **show port-profile** **Example:** `switch(config)# show port-profile name`	Displays the port profile configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enter port profile configuration mode and enable the port profile:

```
switch# configure terminal
switch(config)# port-profile type ethernet test
switch(config-port-prof)# state enabled
switch(config-port-prof)#
```

Inheriting a Port Profile

You can inherit a port profile onto an existing port profile. The switch supports four levels of inheritance.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**port-profile** *name*
	Example:
	`switch(config)# port-profile test`
	`switch(config-port-prof)#`
		Enters port profile configuration mode for the specified port profile.

Step 3	**inherit port-profile** *name*
	Example:
	`switch(config-port-prof)#  inherit port-profile adam`
	`switch(config-port-prof)#`
		Inherits another port profile onto the existing one. The original port profile assumes all the configurations of the inherited port profile.

	Command or Action	Purpose

Step 4	**exit**
	Example:
	`switch(config-port-prof)# exit`
	`switch(config)#`
		Exits the port profile configuration mode.

Step 5	(Optional) **show port-profile** **Example:** `switch(config)# show port-profile name`	Displays the port profile configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to inherit the port profile named adam onto the port profile named test:

```
switch# configure terminal
switch(config)# port-profile test
switch(config-ppm)# inherit port-profile adam
switch(config-ppm)#
```

This example shows how to add the interface commands to a port profile named ppEth configured for Ethernet interfaces:

```
switch# configure terminal
switch(config)# port-profile ppEth
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport trunk allowed vlan 300-400
switch(config-port-prof)# flowcontrol receive on
switch(config-port-prof)# speed 10000
switch(config-port-prof)#
```

This example shows how to inherit a port profile named ppEth configured for Ethernet interfaces into an existing port profile named test:

```
switch# configure terminal
switch(config)# port-profile test
switch(config-port-prof)# inherit port-profile ppEth
switch(config-port-prof)#
```

Removing an Inherited Port Profile

You can remove an inherited port profile.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

Step 2	**port-profile** *name*
	Example:
	`switch(config)# port-profile test`
	`switch(config-port-prof)#`
		Enters port profile configuration mode for the specified port profile.

Step 3	**no inherit port-profile** *name*
	Example:
	`switch(config-port-prof)#  no inherit port-profile adam`
	`switch(config-port-prof)#`
		Removes an inherited port profile from this port profile.

Step 4	**exit**
	Example:
	`switch(config-port-prof)# exit`
	`switch(config)#`
		Exits the port profile configuration mode.

Step 5	(Optional) **show port-profile** **Example:** `switch(config)# show port-profile name`	Displays the port profile configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to remove the inherited port profile named adam from the port profile named test:

```
switch# configure terminal
switch(config)# port-profile test
switch(config-ppm)# no inherit port-profile adam
switch(config-ppm)#
```

Assigning a Port Profile to a Range of Interfaces

You can assign a port profile to an interface or to a range of interfaces. All of the interfaces must be the same type.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface** [**ethernet** *slot/port* | **interface-vlan** *vlan-id* | **port-channel** *number*]	Selects the range of interfaces.
Step 3	**inherit port-profile** *name*	Assigns the specified port profile to the selected interfaces.
Step 4	**exit**	Exits port profile configuration mode.
Step 5	(Optional) **show port-profile** *name*	Displays the port profile configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to assign the port profile named adam to Ethernet interfaces 2/3 to 2/5, 3/2, and 1/20 to 1/25:

```
switch# configure terminal
switch(config)# interface ethernet 2/3 to 2/5, 3/2, and 1/20 to 1/25
switch(config-if)# inherit port-profile adam
switch(config-if)# exit
switch(config)# show port-profile adam
switch(config)# copy running-config startup-config
```

Removing a Port Profile from a Range of Interfaces

You can remove a port profile from some or all of the interfaces to which you have applied the profile.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**interface** [**ethernet** *slot/port* | **interface-vlan** *vlan-id* | **port-channel** *number*]	Selects the range of interfaces.
Step 3	**no inherit port-profile** *name*	Removes the specified port profile from the selected interfaces.
Step 4	**exit**	Exits port profile configuration mode.
Step 5	(Optional) **show port-profile**	Displays the port profile configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how tos remove the port profile named adam from Ethernet interfaces 1/3-5:

```
switch# configure terminal
switch(config)# interface ethernet 1/3-5
switch(config-if)# no inherit port-profile adam
switch(config-if)# exit
switch(config)# show port-profile
switch(config)# copy running-config startup-config
```

Configuration Examples for Port Profiles

The following example shows how to configure a port profile, inherit the port profile on an Ethernet interface, and enabling the port profile.

```
switch(config)#
switch(config)# show running-config interface Ethernet1/14

!Command: show running-config interface Ethernet1/14
!Time: Thu Aug 26 07:01:32 2010

version 5.0(2)N1(1)

interface Ethernet1/14

switch(config)# port-profile type ethernet alpha
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport trunk allowed vlan 10-15
switch(config-port-prof)#
switch(config-port-prof)# show running-config port-profile alpha

!Command: show running-config port-profile alpha
!Time: Thu Aug 26 07:02:29 2010

version 5.0(2)N1(1)
port-profile type ethernet alpha
  switchport mode trunk
  switchport trunk allowed vlan 10-15


switch(config-port-prof)# int eth 1/14
switch(config-if)# inherit port-profile alpha
switch(config-if)#
switch(config-if)# port-profile type ethernet alpha
switch(config-port-prof)# state enabled
switch(config-port-prof)#
switch(config-port-prof)# sh running-config interface ethernet 1/14

!Command: show running-config interface Ethernet1/14
!Time: Thu Aug 26 07:03:17 2010

version 5.0(2)N1(1)

interface Ethernet1/14
  inherit port-profile alpha

switch(config-port-prof)# sh running-config interface ethernet 1/14 expand-port-profile

!Command: show running-config interface Ethernet1/14 expand-port-profile
!Time: Thu Aug 26 07:03:21 2010
```

```
version 5.0(2)N1(1)

interface Ethernet1/14
  switchport mode trunk
  switchport trunk allowed vlan 10-15

switch(config-port-prof)#
```

Configuring the Debounce Timer

You can enable the debounce timer for Ethernet ports by specifying a debounce time (in milliseconds) or disable the timer by specifying a debounce time of 0.

To enable or disable the debounce timer, perform this task:

Procedure

Step 1 switch# **configure terminal**

Enters global configuration mode.

Step 2 switch(config)# **interface** *type slot*/*port*

Enters interface configuration mode for the specified interface.

Step 3 switch(config-if)# **link debounce link-up time** *milliseconds*

Delays link-up declaration by configured time, in milliseconds. The range is 1 to 5000 milliseconds.

Step 4 switch(config-if)# **link debounce time** *milliseconds*

Delays link-down notification by configured time, in milliseconds. The range is 1 to 5000 milliseconds.

Disables the debounce timer if you specify 0 milliseconds.

Step 5 switch(config-if)# **no link debounce**

Sets debounce timer to the default value of 100 milliseconds.

Step 6 switch(config-if)# **no link debounce link-up**

Disables link debounce link-up.

Example

This example shows how to enable the debounce timer 1000 milliseconds for an ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# link debounce time 1000
```

This example shows how to disable debounce timer for an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# link debounce time 0
```

This example shows how to enable link-up debounce timer of 200 milliseconds for an interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# link debounce link-up time 200
```

This example shows how to disable link-up debounce for an interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# no link debounce link-up
```

Configuring the Description Parameter

You can provide textual interface descriptions for the Ethernet ports.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **description** *test*	Specifies the description for the interface.

Example

This example shows how to set the interface description to Server 3 interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# description Server 3 Interface
```

Disabling and Restarting Ethernet Interfaces

You can shut down and restart an Ethernet interface. This action disables all of the interface functions and marks the interface as being down on all monitoring displays. This information is communicated to other network servers through all dynamic routing protocols. When shut down, the interface is not included in any routing updates.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface.

	Command or Action	Purpose
Step 3	switch(config-if)# **shutdown**	Disables the interface.
Step 4	switch(config-if)# **no shutdown**	Restarts the interface.

Example

This example shows how to disable an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# shutdown
```

This example shows how to restart an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# no shutdown
```

Configuring Slow Drain Device Detection and Congestion Avoidance

Fibre Channel Slow Drain Device Detection and Congestion Avoidance- An Overview

All data traffic between end devices in the SAN fabric is carried by Fibre Channel Class 3, and in some cases, Class 2 services, that use link-level, per-hop-based, and buffer-to-buffer flow control. These classes of service do not support end-to-end flow control. When slow devices are attached to the fabric, the end devices do not accept the frames at the configured or negotiated rate. The slow devices lead to an Inter-Switch Link (ISL) credit shortage in the traffic that is destined for these devices and they congest the links. The credit shortage affects the unrelated flows in the fabric that use the same ISL link even though destination devices do not experience a slow drain.

This feature provides various enhancements that enable you to detect slow drain devices are cause congestion in the network and also provide congestion avoidance.

The enhancements are mainly on the edge ports that connect to the slow drain devices to minimize the frames stuck condition in the edge ports due to slow drain devices that are causing an ISL blockage. To avoid or minimize the stuck condition, configure lesser frame timeout for the ports. You can use the no-credit timeout to drop all packets after the slow drain is detected using the configured thresholds. A smaller frame timeout value helps to alleviate the slow drain condition that affects the fabric by dropping the packets on the edge ports sooner than the time they actually get timed out (358 ms). This function frees the buffer space in ISL, which can be used by other unrelated flows that do not experience slow drain condition.

Note This feature supports edge ports that are connected to slow edge devices. Even though you can apply this feature to ISLs as well, we recommend that you apply this feature only for edge F ports and retain the default configuration for ISLs as E and TE ports. This feature is not supported on Generation 1 modules.

Configuring a Stuck Frame Timeout Value

The default stuck frame timeout value is 358 ms. The timeout value can be incremented in steps of 10. We recommend that you retain the default configuration for ISLs and configure a value that does not exceed 500 ms (100 to 200 ms) for fabric F ports.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **system timeout congestion-drop** *seconds* **mode E | F**	Specifies the stuck frame timeout value in milliseconds and the port mode for the switch.
Step 3	switch(config)# **system timeout congestion-drop default mode E | F**	Specifies the default stuck frame timeout port mode for the switch.

Example

This example shows how to configure a stuck frame timeout value of 100 ms:

```
switch# configure terminal
switch(config)# system timeout congestion-drop 100 mode F
switch(config)# system timeout congestion-drop default mode F
```

Configuring a No-Credit Timeout Value

When the port does not have the credits for the configured period, you can enable a no-credit timeout on that port, which results in all frames that come to that port getting dropped in the egress. This action frees the buffer space in the ISL link, which helps to reduce the fabric slowdown and congestion on other unrelated flows that use the same link.

The dropped frames are the frames that have just entered the switch or have stayed in the switch for the configured timeout value. These drops are preemptive and clear the congestion completely.

The no-credit timeout feature is disabled by default. We recommend that you retain the default configuration for ISLs and configure a value that does not exceed 358 ms (200 to 300 ms) for fabric F ports.

You can disable this feature by entering the **no system timeout no-credit-drop mode F** command.

Note The no-credit timeout value and stuck frame timeout value are interlinked. The no-credit timeout value must always be greater than the stuck frame timeout value.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **system timeout no-credit-drop** *seconds* **mode F**	Specifies the no-credit timeout value and port mode for the switch. The *seconds* value is 500ms by default. This value can be incremented in steps of 100.
Step 3	switch(config)# **system timeout no-credit-drop default mode F**	Specifies the default no-credit timeout value port mode for the switch.

Example

This example shows how to configure a no-credit timeout value:

```
switch# configure terminal
switch(config)# system timeout no-credit-drop 100 mode F
switch(config)# system timeout no-credit-drop default mode F
```

Displaying Credit Loss Counters

Use the following commands to display the credit loss counters per module per interface for the last specified minutes, hours, and days:

Procedure

	Command or Action	Purpose
Step 1	**show process creditmon** {**credit-loss-event-history** | **credit-loss-events** | **force-timeout-events** | **timeout-discards-events**}	Displays Onboard Failure Logging (OBFL) credit loss logs.

Displaying Credit Loss Events

Use one of the following commands to display the total number of credit loss events per interface with the latest three credit loss time stamps:

Command	Purpose
show process creditmon credit-loss-events [**module** *module number*]	Displays the credit loss event information for a module.
show process creditmon credit-loss-event-history [**module** *module number*]	Displays the credit loss event history information.

Displaying Timeout Drops

Use the following command to display the timeout drops per module per interface for the last specified minutes, hours, and days:

Command	Purpose
show logging onboard flow-control timeout-drops [**last** *mm* **minutes**] [**last** *hh* **hours**] [**last** *dd* **days**] [**module** *module number*]	Displays the Onboard Failure Logging (OBFL) timeout drops log.

Displaying the Average Credit Not Available Status

When the average credit nonavailable duration exceeds the set threshold, you can error-disable the port, send a trap with interface details, and generate a syslog with interface details. In addition, you can combine or more actions or turn on or off an action. The port monitor feature provides the command line interface to configure the thresholds and action. The threshold configuration can be a percentage of credit non-available duration in an interval.

The thresholds for the credit nonavailable duration can be 0 percent to 100 percent in multiples of 10, and the interval can be from 1 second to 1 hour. The default is 10 percent in 1 second and generates a syslog.

Use the following command to display the average credit-not-available status:

Command	Purpose
show system internal snmp credit-not-available {**module** | **module-id**}	Displays the port monitor credit-not-available counter logs.

Port Monitoring

You can use port monitoring to monitor the performance of fabric devices and to detect slow drain devices. You can monitor counters and take the necessary action depending on whether the portguard is enabled or disabled. You can configure the thresholds for various counters and trigger an event when the values cross the threshold settings. Port monitoring provides a user interface that you can use to configure the thresholds and action. By default, portguard is disabled in the port monitoring policy.

Two default policies, default and default slowdrain, are created during snmpd initialization. The default slowdrain policy is activated when the switch comes online when no other policies are active at that time. The default slowdrain policy monitors only credit-loss-reco and tx-credit-not-available counters.

When you create a policy, it is created for both access and trunk links. The access link has a value of F and the trunk link has a value of E.

Enabling Port Monitor

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **port-monitor enable**	Enables (default) the port monitoring feature. The **no** version of this command disables the port monitoring feature.

Configuring a Port Monitor Policy

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-monitor name** *policyname*	Specifies the policy name and enters the port monitor policy configuration mode.
Step 3	switch(config-port-monitor)# **port-type all**	Applies the policy to all ports.
Step 4	switch(config-port-monitor)# **counter** {**credit-loss-reco** | **timeout-discards** | **tx-credit-not-available** } **poll-interval** *seconds* {**absolute** | **delta**} **rising-threshold** *value1* **event** *event-id1* **falling-threshold** *value2* **event** *event-id2*	Specifies the poll interval in seconds, the thresholds in absolute numbers, and the event IDs of events to be triggered for the following reasons: • credit-loss-reco—Credit loss recovery • timeout-discards—Timeout discards • tx-credit-not-available—Average credit non-available duration
Step 5	switch(config-port-monitor)# [**no**] **counter** {**credit-loss-reco** | **timeout-discards** | **tx-credit-not-available** } **poll-interval** *seconds* {**absolute** | **delta**} **rising-threshold** *value1* **event** *event-id1* **falling-threshold** *value2* **event** *event-id2*	Turns on monitoring for the specified counter. The **no** form of this command turns off monitoring for the specified counter.

Example

This example shows how to specify the poll interval and threshold for timeout discards:

```
switch# configure terminal
switch(config)# port-monitor cisco
switch(config-port-monitor)# counter timeout-discards poll-interval 10
```

This example show how to specify the poll interval and threshold for credit loss recovery:

```
switch# configure terminal
switch(config)# port-monitor cisco
```

```
switch(config-port-monitor)# counter credit-loss-reco poll-interval 20 delta rising-threshold
 10 event 4 falling-threshold 3 event 4
```

Activating a Port Monitor Policy

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-monitor activate** *policyname*	Activates the specified port monitor policy.
Step 3	(Optional) switch(config)# **port-monitor activate**	Activates the default port monitor policy.
Step 4	(Optional) switch(config)# **no port-monitor activate** *policyname*	Deactivates the specified port monitor policy.

Example

This example shows how to activate a specific port monitor policy:

```
switch# configure terminal
switch(config)# port-monitor activate cisco
```

Displaying Port Monitor Policies

Use the following command to display port monitor policies:

Command	Purpose
switch# **show port-monitor** *policyname*	Displays details of the specified port monitor policy.

Example

This example shows how to display a specific port monitor policy:

FCoE Slow Drain Device Detection and Congestion Avoidance

The data traffic between end devices in Fibre Channel over Ethernet (FCoE) uses link level, per-hop Priority Flow Control (PFC). This allows the FCoE class on a link to be paused independently in each direction, while other classes continue to transmit and receive on the link. When end devices transmit PFC pause frames to the switch port they prevent the switch port from being able to transmit FCoE frames to the end device. Although some of this occurs normally, if it occurs in large amounts it can cause congestion in the fabric. End devices doing this are called a slow devices, or slow drain devices. When this occurs it can cause frames to queue at the switch which results in the switch transmitting its own PFC pause frames back towards the source of the incoming frames. If the switch port where the frames are being received (the source of the incoming

frames) is connected to an end device, then this end device will temporarily be paused. It will not be able to transmit any frames into the switch for any destination (not just for the slow device). If switch port where the frames are being received on is an Inter-Switch-Link (ISL) then all inbound traffic across that ISL will be paused. This will affect all devices transiting that ISL.

There are two ways to mitigate FCoE slowdrain on a Cisco Nexus 5500 switch:

- Congestion timeout, on page 45
- Pause timeout, on page 45

Congestion timeout

Congestion timeout measures the age of frames that have been received by the switch. It automatically drops the FCoE frames that have been received by the switch, but are not able to transmit for 358 milliseconds. You cannot modify the congestion timeout value for FCoE.

Pause timeout

Pause timeout automatically drops all the FCoE frames that have been received by the switch and queued for an egress port when the egress port is in a continual paused state for the associated time. By default this feature is off, but it can be configured to be 90 milliseconds, 180 milliseconds, 358 milliseconds, 716 milliseconds, or 1433 milliseconds. The lower the value the quicker the switch will react to a port in a continual state of a pause. When a port reaches the pause timeout threshold, all the FCoE frames queued for egress on that port are emptied from the queue regardless of their exact age. The threshold is detected by a software process that runs every 100 milliseconds. Since all the frames queued to a given egress port are dropped this can have a dramatic effect on reducing the congestion on affected ISLs (ISLs from which the frames originated). When this condition is detected it is called a "Pause Event". The switch issues the following message when a pause event is detected:

```
switchname %$ VDC-1 %$ %CARMELUSD-2-CARMEL_SYSLOG_CRIT: FCoE Pause Event Occurred on interface
 ethernet 1/1
```

For every pause event that lasts for the specified timeout value, a pause event is published to the Embedded Event Manager (EEM). The EEM maintains the count of pause events per port and triggers the policy action when the threshold is reached.

The following are the two EEM policies that exist by default. Use the **show event manager system-policy** command to view the EEM policies.

- ```
  switch# show event manager system-policy
    Name : __ethpm_slow_drain_core
    Description : 10 Pause Events in 1 minute. Action: None by default
    Overridable : Yes
  ```

- ```
  switch# show event manager system-policy
    Name : __ethpm_slow_drain_edge
       Description : 5 Pause Events in 1 minute. Action: None by default
       Overridable : Yes
  ```

You can override the default policy with the new thresholds and actions. If you try to override the EEM system policies _ethpm_slow_drain_edge and _ethpm_slow_drain_core, the default-action, default syslog, will also appear. We recommend that you specify action err-disable to isolate the faulty port where this condition occurs. This can be done by overriding the _ethpm_slow_drain_edge EEM policy.

The following is a sample output to override the EEM system policy:

```
event manager applet custom_edge_policy override __ethpm_slow_drain_edge
event policy-default count 5 time 360
action 1.0 syslog msg FCoE Slowdrain Policy Was Hit
exit
```

In the above example, the EEM policy generates a syslog if five pause events occur in 360 seconds on an edge port.

Configuring a Pause Frame Timeout Value

You can enable or disable a pause frame timeout value on a port. The system periodically checks the ports for a pause condition and enables a pause frame timeout on a port if it is in a continuous pause condition for a configured period of time. This situation results in all frames that come to that port getting dropped in the egress. This function empties the buffer space in the ISL link and helps to reduce the fabric slowdown and the congestion on the other unrelated flows using the same link.

When a pause condition is cleared on a port or when a port flaps, the system disables the pause frame timeout on that particular port.

The pause frame timeout is disabled by default. We recommend that you retain the default configuration for the ISLs and configure a value that does not exceed the default value for the edge ports.

For a faster recovery from the slow drain device behavior, you should configure a pause frame timeout value because it drops all the frames in the edge port that face the slow drain whether the frame is in the switch for a congested timeout or not. This process instantly clears the congestion in the ISL. You should configure a pause frame timeout value to clear the congestion completely instead of configuring a congestion frame timeout value.

Use the **no system default interface pause timeout milliseconds mode {core | edge}** command to disable the pause frame timeout value on the edge ports. The default pause timeout value is 358 milliseconds.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch# **system default interface pause timeout** *milliseconds* **mode** {**core** | **edge**}	Configures a new pause frame timeout value in milliseconds and the port mode for the device.
Step 3	switch# **system default interface pause mode** {**core** | **edge**}	Configures the default pause frame timeout value in milliseconds and the port mode for the device.
Step 4	switch# **no system default interface pause timeout** *milliseconds* **mode** {**core** | **edge**}	Disables the pause frame timeout for the device.
Step 5	switch# **no system default interface pause mode** {**core** | **edge**}	Disables the default pause frame timeout for the device.
Step 6	(Optional) switch# **show logging onboard flow-control pause-event**	Displays the total number of the pause events per module per interface.
Step 7	(Optional) switch# **show logging onboard flow-control timeout-drop**	Displays the timeout drops per module per interface with the time-stamp information.

Example

This example shows how to configure a pause frame timeout value:

```
switch# configure terminal
switch(config)# system default interface pause timeout 358 mode core
switch(config)# system default interface pause mode edge
switch(config)# no system default interface pause timeout 358 mode core
switch(config)# no system default interface pause mode edge
switch(config)# end
switch# show logging onboard flow-control pause-event
switch# show logging onboard flow-control timeout-drop
```

This example shows how to display the total number of the pause events for the entire switch:

```
switch# show logging onboard flow-control pause-events
List of Pause Events
--------------------------------------------------
Ethernet      Timestamp
Interface
--------------------------------------------------
 1/1          01/01/2009 10:15:20.262951
 1/1          01/01/2009 10:15:21.462869
 1/1          01/01/2009 10:15:22.173349
 1/1          01/01/2009 10:15:22.902929
 1/1          01/01/2009 10:15:23.642984
 1/1          01/01/2009 10:15:24.382961
 1/1          01/01/2009 10:15:25.100497
 1/1          01/01/2009 10:15:25.842915
```

This example shows how to display the timeout drops per interface with time-stamp information for the supervisor CLI:

```
switch# show logging onboard flow-control timeout-drops
Number of Pause Events per Port
----------------------------
Ethernet   Number of
Interface  Pause Events
----------------------------
1/1        38668
1/15       232
2/16       2233
2/17       2423
```

Displaying Interface Information

To view configuration information about the defined interfaces, perform one of these tasks:

Command	Purpose
switch# **show interface** *type slot*/*port*	Displays the detailed configuration of the specified interface.
switch# **show interface** *type slot*/*port* **capabilities**	Displays detailed information about the capabilities of the specified interface. This option is available only for physical interfaces.

Command	Purpose
switch# **show interface** *type slot/port* **transceiver**	Displays detailed information about the transceiver connected to the specified interface. This option is available only for physical interfaces.
switch# **show interface brief**	Displays the status of all interfaces.
switch# **show interface flowcontrol**	Displays the detailed listing of the flow control settings on all interfaces.
switch# **show interface debounce**	Displays the debounce status of all interfaces.

The **show interface** command is invoked from EXEC mode and displays the interface configurations. Without any arguments, this command displays the information for all the configured interfaces in the switch.

This example shows how to display the physical Ethernet interface:

```
switch# show interface ethernet 1/1
  Ethernet1/1 is up
  Hardware is 1000/10000 Ethernet, address is 000d.eca3.5f08 (bia 000d.eca3.5f08)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 190/255, rxload 192/255
  Encapsulation ARPA
  Port mode is trunk
  full-duplex, 10 Gb/s, media type is 1/10g
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned on
  Rate mode is dedicated
  Switchport monitor is off
  Last clearing of "show interface" counters never
  5 minute input rate 942201806 bytes/sec, 14721892 packets/sec
  5 minute output rate 935840313 bytes/sec, 14622492 packets/sec
  Rx
    129141483840 input packets 0 unicast packets 129141483847 multicast packets
    0 broadcast packets 0 jumbo packets 0 storm suppression packets
    8265054965824 bytes
    0 No buffer 0 runt 0 Overrun
    0 crc 0 Ignored 0 Bad etype drop
    0 Bad proto drop
  Tx
    119038487241 output packets 119038487245 multicast packets
   0 broadcast packets 0 jumbo packets
    7618463256471 bytes
    0 output CRC 0 ecc
    0 underrun 0 if down drop     0 output error 0 collision 0 deferred
    0 late collision 0 lost carrier 0 no carrier
    0 babble
    0 Rx pause 8031547972 Tx pause 0 reset
```

This example shows how to display the physical Ethernet capabilities:

```
switch# show interface ethernet 1/1 capabilities
Ethernet1/1
  Model:                734510033
  Type:                 10Gbase-(unknown)
  Speed:                1000,10000
  Duplex:               full
  Trunk encap. type:    802.1Q
  Channel:              yes
```

```
  Broadcast suppression: percentage(0-100)
  Flowcontrol:           rx-(off/on),tx-(off/on)
  Rate mode:             none
  QOS scheduling:        rx-(6q1t),tx-(1p6q0t)
  CoS rewrite:           no
  ToS rewrite:           no
  SPAN:                  yes
  UDLD:                  yes
  Link Debounce:         yes
  Link Debounce Time:    yes
  MDIX:                  no
  FEX Fabric:            yes
```

This example shows how to display the physical Ethernet transceiver:

```
switch# show interface ethernet 1/1 transceiver
Ethernet1/1
    sfp is present
    name is CISCO-EXCELIGHT
    part number is SPP5101SR-C1
    revision is A
    serial number is ECL120901AV
    nominal bitrate is 10300 MBits/sec
    Link length supported for 50/125mm fiber is 82 m(s)
    Link length supported for 62.5/125mm fiber is 26 m(s)
    cisco id is --
    cisco extended id number is 4
```

This example shows how to display a brief interface status (some of the output has been removed for brevity):

```
switch# show interface brief

--------------------------------------------------------------------------------
Ethernet      VLAN   Type Mode   Status  Reason                   Speed     Port
Interface                                                                  Ch #
--------------------------------------------------------------------------------
Eth1/1        200    eth  trunk  up      none                       10G(D) --
Eth1/2        1      eth  trunk  up      none                       10G(D) --
Eth1/3        300    eth  access down    SFP not inserted           10G(D) --
Eth1/4        300    eth  access down    SFP not inserted           10G(D) --
Eth1/5        300    eth  access down    Link not connected        1000(D) --
Eth1/6        20     eth  access down    Link not connected         10G(D) --
Eth1/7        300    eth  access down    SFP not inserted           10G(D) --
...
```

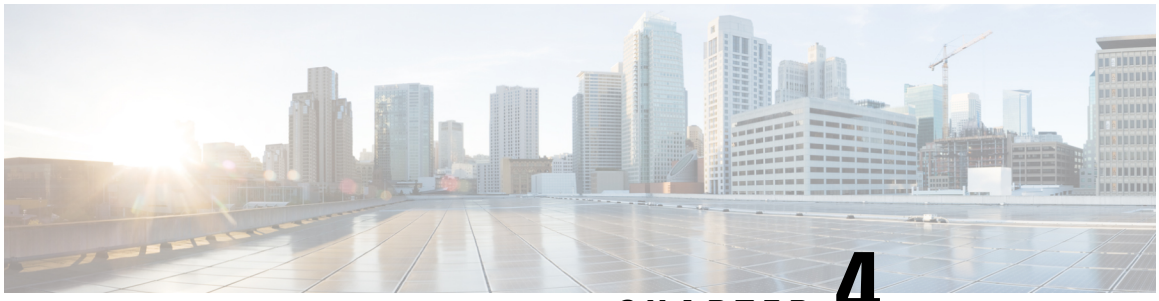
This example shows how to display the link debounce status (some of the output has been removed for brevity):

```
switch# show interface debounce

--------------------------------------------------------------------------------
Port          Debounce time   Value(ms)
--------------------------------------------------------------------------------
...
Eth1/1        enable              100
Eth1/2        enable              100
Eth1/3        enable              100
...
```

This example shows how to display the CDP neighbors:

```
switch# show cdp neighbors
```

```
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID              Local Intrfce   Hldtme  Capability  Platform      Port ID
d13-dist-1               mgmt0           148     S I      WS-C2960-24TC  Fas0/9
n5k(FLC12080012)         Eth1/5          8       S I s    N5K-C5020P-BA  Eth1/5
```



CHAPTER 4

Configuring VLANs

This chapter contains the following sections:

Information About VLANs

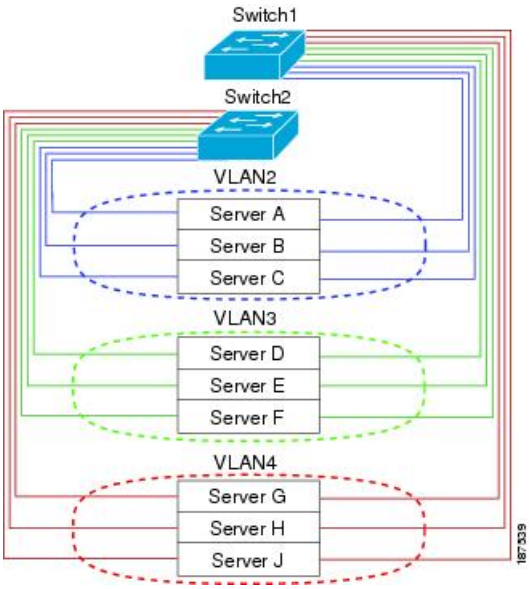
Understanding VLANs

A VLAN is a group of end stations in a switched network that is logically segmented by function, project team, or application, without the limitation to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any port can belong to a VLAN; all unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered a logical network. If a packet destination address does not belong to the VLAN, it must be forwarded through a router.

The following figure shows VLANs as logical networks. In this diagram, the stations in the engineering department are assigned to one VLAN, the stations in the marketing department are assigned to another VLAN, and the stations in the accounting department are assigned to yet another VLAN.

Figure 4: VLANs as Logically Defined Networks



VLANs are usually associated with IP subnetworks. For example, all the end stations in a particular IP subnet belong to the same VLAN. To communicate between VLANs, you must route the traffic.

By default, a newly created VLAN is operational. To disable the VLAN use the **shutdown** command. Additionally, you can configure VLANs to be in the active state (passing traffic), or the suspended state (in which the VLANs are not passing packets). By default, the VLANs are in the active state and pass traffic.

Note The VLAN Trunking Protocol (VTP) mode is OFF. VTP BPDUs are dropped on all interfaces of the switch. This process has the effect of partitioning VTP domains if other switches have VTP turned on.

Understanding VLAN Ranges

The Cisco Nexus device supports VLAN numbers 1 to 4094 in accordance with the IEEE 802.1Q standard. These VLANs are organized into ranges. The switch is physically limited in the number of VLANs it can support. For information about VLAN configuration limits, see the configuration limits documentation for your device.

The following table describes the details of the VLAN ranges.

Table 3: VLAN Ranges

VLANs Numbers	Range	Usage
1	Normal	Cisco default. You can use this VLAN, but you cannot modify or delete it.
2—1005	Normal	You can create, use, modify, and delete these VLANs.

VLANs Numbers	Range	Usage
1006—4094	Extended	You can create, name, and use these VLANs. You cannot change the following parameters: • State is always active. • VLAN is always enabled. You cannot shut down these VLANs.
3968—4049 and 4094	Internally allocated	These 82 VLANs, plus VLAN 4094, are allocated for internal use. You cannot create, delete, or modify any VLANs within the block reserved for internal use.

Note You cannot configure the internally allocated VLANs (reserved VLANs).

Note VLANs 3968 to 4049 and 4094 are reserved for internal use; these VLANs cannot be changed or used.

Cisco NX-OS allocates a group of 82 VLAN numbers for those features, such as multicast and diagnostics, that need to use internal VLANs for their operation. By default, the system allocates VLANs numbered 3968 to 4049 for internal use. VLAN 4094 is also reserved for internal use by the switch.

You cannot use, modify, or delete any of the VLANs in the reserved group. You can display the VLANs that are allocated internally and their associated use.

Creating, Deleting, and Modifying VLANs

VLANs are numbered from 1 to 4094. All configured ports belong to the default VLAN when you first bring up the switch. The default VLAN (VLAN1) uses only default values. You cannot create, delete, or suspend activity in the default VLAN.

You create a VLAN by assigning a number to it. You can delete VLANs as well as move them from the active operational state to the suspended operational state. If you attempt to create a VLAN with an existing VLAN ID, the switch goes into the VLAN submode but does not create the same VLAN again.

Newly created VLANs remain unused until ports are assigned to the specific VLAN. All the ports are assigned to VLAN1 by default.

Depending on the range of the VLAN, you can configure the following parameters for VLANs (except the default VLAN):

- VLAN name
- Shutdown or not shutdown

When you delete a specified VLAN, the ports associated to that VLAN are shut down and no traffic flows. However, the system retains all the VLAN-to-port mapping for that VLAN, and when you reenable, or recreate, the specified VLAN, the system automatically reinstates all the original ports to that VLAN.

Note Commands entered in the VLAN configuration submode are immediately executed.

VLANs 3968 to 4049 and 4094 are reserved for internal use; these VLANs cannot be changed or used.

About the VLAN Trunking Protocol

VLAN Trunking Protocol (VTP) is a distributed VLAN database management protocol that synchronizes the VTP VLAN database across domains. A VTP domain includes one or more network switches that share the same VTP domain name and are connected with trunk interfaces.

Guidelines and Limitations for VTP

VTP has the following configuration guidelines and limitations:

- When a switch is configured as a VTP client, you cannot create VLANs on the switch in the range of 1 to 1005.
- VLAN 1 is required on all trunk ports used for switch interconnects if VTP is supported in the network. Disabling VLAN 1 from any of these ports prevents VTP from functioning properly.
- If you enable VTP, you must configure either version 1 or version 2. On the Cisco Nexus device, 512 VLANs are supported. If these switches are in a distribution network with other switches, the limit remains the same.

  On the Cisco Nexus device, 512 VLANs are supported. If these switches are in a distribution network with other switches, the VLAN limit for the VTP domain is 512. If a Cisco Nexus device client/server receives additional VLANs from a VTP server, they transition to transparent mode.
- If **system vlan long-name** knob is enabled, then VTP configurations will come up in OFF mode and users can change the mode to Transparent. However, changing the mode to Server or Client is not allowed.
- The **show running-configuration** command does not show VLAN or VTP configuration information for VLANs 1 to 1000.
- When deployed with vPC, both vPC switches must be configured identically. vPC performs a Type 2 consistency check for VTP configuration parameters.
- VTP advertisements are not sent out on Cisco Nexus Fabric Extender ports.
- Private VLANs (PVLANs) are supported only when the switch is in transparent mode.
- If you are using VTP in a Token Ring environment, you must use version 2.
- When a switch is configured in VTP client or server mode, VLANs 1002 to1005 are reserved VLANs.
- VTPv3 pruning is supported from Cisco NX-OS Release 7.2(0)N1(1) onwards.
- You must enter the **copy running-config startup-config** command followed by a reload after changing a reserved VLAN range. For example:

```
switch(config)# system vlan 2000 reserve
This will delete all configs on vlans 2000-2081. Continue anyway? (y/n) [no] y
```

After the switch reload, VLANs 2000 to 2081 are reserved for internal use, which requires that you enter the **copy running-config startup-config** command before the switch reload. Creating VLANs within this range is not allowed.

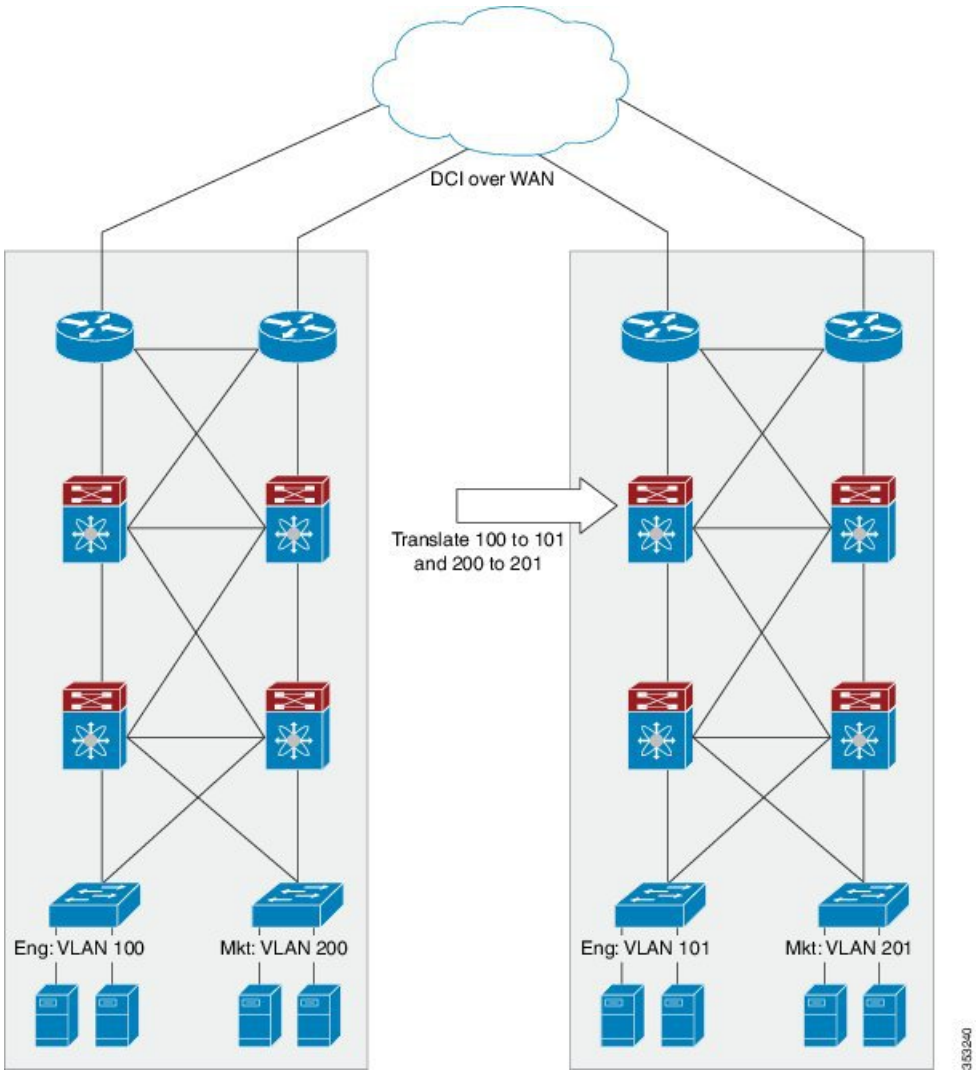
- Ensure VLAN 1 is not STP blocked for VTP interfaces in VTP transparent mode.
- In SNMP, the vlanTrunkPortVtpEnabled object indicates whether the VTP feature is enabled or not.

About VLAN Translation

In a data center there are often instances when you want to merge separate Layer 2 domains. For example, you might have two data centers that are connected via some form of Data Center Interconnect (DCI) such as Overlay Transport Virtualization (OTV). Both data centers might have an engineering group that has its own VLAN in each data center. Due to differences such as different administrators, the VLAN number might be different in each data center. Once the two data centers are connected via DCI, it makes sense that all engineering traffic should be visible in both data centers. In complex installations reconfiguration is not worth the collateral damage reconfiguration can cause. This is a scenario where VLAN translation would be useful to merge the two Layer 2 domains without actually changing their VLAN number.

This document describes the functionality of the VLAN translation feature on NX-OS and its interaction with other features on the Cisco Nexus device. The following diagram shows a possible datacenter application for VLAN translation.

Figure 5: DC VLAN Translation



The first datacenter on the left has an engineering VLAN with number 100 and a marketing VLAN with number 200. The second datacenter on the right has an engineering VLAN with number 101 and a marketing VLAN with number 201. For the engineering machines in the second datacenter to see data from the engineering machines in the first datacenter, the core Cisco Nexus device in the second datacenter must translate the VLAN ID in the ingress packets on the trunk port from the ingress VLAN 100 to the local VLAN 101. The local VLAN tag is a function of the port on which the traffic arrives and the ingress VLAN tag on which it arrives. Upon egress from the trunk port, the reverse translation must be to convert VLAN 101 to VLAN 100.

For example, VLAN translation can be enabled on a port such that packets with ingress VLANs V1,V2…V10 are mapped to local VLANs V101, V102,…,V110, the packets coming in to the second network are tagged as follows:

V1, V2, V10 map to V101, V102, V110 respectively (Packets are single tagged and tag is a function of ingress VLAN tag and port).

For a given port, there is a strict one-to-one mapping of the ingress VLAN to local VLAN and more than one ingress VLAN is not allowed to map to the same local VLAN.

Guidelines and Limitations for Configuring VLANs

VLANs have the following configuration guidelines and limitations:

- The maximum number of VLANs per VDC is 4094.
- You can configure a single VLAN or a range of VLANs.

  When you configure a large number of VLANs, first create the VLANs using the **vlan** command (for example, **vlan** *200 to 300*, *303 to 500*). After the VLANs have been successfully created, name or configure those VLANs sequentially.
- VLAN 4094 is a reserved VLAN.
- You cannot create, modify, or delete any VLANs that are within the group of VLANs reserved for internal use.
- VLAN1 is the default VLAN. You cannot create, modify, or delete this VLAN.
- VLANs 1006 to 4094 are always in the active state and are always enabled. You cannot suspend the state or shut down these VLANs.

VLAN translation has the following guidelines and limitations:

- A VLAN translation configuration is only applicable to Layer 2 trunks. It is inactive when applied to ports that are not Layer 2 trunks.
- Do not configure translation of ingress native VLAN traffic on an 802.1Q trunk. The 802.1Q native VLAN traffic is untagged and cannot be recognized for translation. However, you can translate traffic from other VLANs to the native VLAN of an 802.1Q trunk.
- The VLANs to which you are translating must be present in the trunk's allowed VLAN list. In addition, the VLANs that need to be forwarded on a trunk port, that are not involved in VLAN translation must also be included in the trunk ports allowed VLAN list. With per-port VLAN translation enabled, VLAN translation entries are consumed in hardware for all VLANs in the trunk ports allowed VLAN list.
- Do not change the VLAN on an access port or a trunk port it will flap the interface. However, if the port is part of a vPC, then first change the native VLAN on the secondary vPC, and then on the primary vPC.
- A VLAN translation must ensure that the original and translated VLANs are within the same MST instance.
- The number of supported VLAN translation maps is 4000. Layer 2 ports that have the same VLAN maps and the same trunk allowed VLAN list can benefit from sharing translation entries in hardware.
- For VLAN translation on a FEX, the VLAN translation maps are applicable to all FEX host interfaces and must be applied to all the FEX fabric or network interfaces. In addition, the translated VLANs specified in the FEX VLAN translation maps must be individually applied to the trunk allowed VLAN list of each of the FEX HIF interfaces. All the FEX interfaces must be configured as Layer 2 trunks.
- VLAN translation is not configurable on FEX HIF ports.
- The VLAN translation feature is only applicable to trunk ports. Hence, in the case of a FEX, all FEX HIF ports must be in trunk mode. When VLAN translation is first enabled on a FEX, a syslog is issued stating that all FEX HIF ports must be in trunk mode.
- For VLAN translation with vPC, the VLAN translation configuration on vPC primary and secondary interfaces must be consistent, otherwise the vPC interface on vPC secondary is brought down.

- If VLAN translation is enabled on a port channel, the configuration is applied to all member ports in the port channel bundle.
- SPAN is supported on trunk ports with VLAN translation enabled.
- PVLAN mode behavior cannot be overlaid on top of ports with VLAN translation enabled.
- To enable DHCP snooping on a port on which VLAN translation is enable, the translated/mapped local VLAN must be used.
- Do not configure VLAN translation on a Peer-Link.
- Do not use VLAN translation on FabricPath core ports.
- Global VLAN translation is not supported.
- To enable IGMP snooping on a VLAN, the VLAN interface must be capable of multicast routing. If VLAN translation is enabled on a port, IGMP snooping has to be enabled on the translated VLAN, that is the local VLAN.
- The following should be taken into consideration when spanning tree (STP) mode is enabled:

Figure 6: VLAN Mapping with SSTP

  - SW1 and SW2 are connected using trunk T that carries VLAN 101. On SW2, per port VLAN mapping is enabled on trunk port P and one of the mappings is 101 to 202. In the previous diagram, on the wire BPDU from SW1 has .1q VLAN and TLV VLAN as 101. When this BPDU reaches port P , its dot1q VLAN is changed from 101 to 202 per the VLAN mapping on Port P. However, the BPDU TLV VLAN remains 101. When it reaches the spanning tree process, spanning tree concludes that VLAN 101's BPDU is received on VLAN 202 and spanning tree reports this as an inconsistent port. To correct the problem, spanning tree should process this BPDU in VLAN 202 and the TLV VLAN should be mapped to translate VLAN and check for consistency. Spanning tree instance 101 of SW1 is merged with spanning tree instance 202 of SW2. The same process is done on the transmit side. You should take this merging on VLANs into consideration before designing the spanning tree topology. With VLAN translation in conjunction with MST, VLAN translation must ensure that the original and translated VLANs are within the same MST instance. You should also ensure that the original VLAN (101) is not present in the trunk allowed VLAN list of local switch (SW2) on its trunk port (P), and that the translated VLAN (202) is not present in the trunk allowed VLAN list of the neighboring switch (SW1), on SW1’s trunk port.

Configuring a VLAN

Creating and Deleting a VLAN

You can create or delete all VLANs except the default VLAN and those VLANs that are internally allocated for use by the switch. Once a VLAN is created, it is automatically in the active state.

Note When you delete a VLAN, ports associated to that VLAN shut down. The traffic does not flow and the packets are dropped.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** {*vlan-id* | *vlan-range*}	Creates a VLAN or a range of VLANs. If you enter a number that is already assigned to a VLAN, the switch moves into the VLAN configuration submode for that VLAN. If you enter a number that is assigned to an internally allocated VLAN, the system returns an error message. However, if you enter a range of VLANs and one or more of the specified VLANs is outside the range of internally allocated VLANs, the command takes effect on *only* those VLANs outside the range. The range is from 2 to 4094; VLAN1 is the default VLAN and cannot be created or deleted. You cannot create or delete those VLANs that are reserved for internal use.
Step 3	switch(config-vlan)# **no vlan** {*vlan-id* | *vlan-range*}	Deletes the specified VLAN or range of VLANs and removes you from the VLAN configuration submode. You cannot delete VLAN1 or the internally allocated VLANs.

Example

This example shows how to create a range of VLANs from 15 to 20:

```
switch# configure terminal
switch(config)# vlan 15-20
```

Note You can create and delete VLANs in the VLAN configuration submode.

Configuring VLAN Long-Name

Note If VTP is enabled, it must be in transparent or in off mode. VTP cannot be in client or server mode. For more details about VTP, see the Configuring VTP chapter.

Beginning with Cisco NX-OS Release 7.3(0)N1(1), the length of VLAN name that you can configure is increased from 32 to 128 characters. In the earlier release version, you could configure the length of VLAN name up to 32 characters.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
```

Enters global configuration mode.

Step 2 **system vlan long-name**

Example:

```
switch(config)# system vlan long-name
```

Allows you to configure the length of VLAN names up to 128 characters.

Note Enabling or disabling the **system vlan long-name** command will trigger a system log message that will let you know if the VLAN long name is enabled or disabled.

If you try to enable or disable the **system vlan long-name** command, when it is already enabled or disabled, the system will throw error message. We recommend you view the status of the VLAN long-name knob before enabling or disabling this command.

Use the **no** form of this command to disable this feature.

Step 3 (Optional) **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Step 4 **show running-config | sec long-name**

Example:

```
switch(config)# show running-config | sec long-name
```

Displays the VLAN long-name status information.

Note When you configure a VLAN name of more than 32 characters, the **show vlan** commands will show the output in mulitple lines with each line containing a maximum of 32 characters.

Example

This example shows how to configure VLAN long-names of up to 128 characters.

```
switch# configure terminal
switch(config)# system vlan long-name
!2001 Sep 29 02:24:11 N72-3 %$ VDC-1 %$ %VLAN_MGR-2-CRITICAL_MSG: VLAN long name is Enabled!
```

```
switch(config)# copy running config startup config
switch(config)# show running-config | sec long-name
switch# configure terminal
switch(config)# vlan 2
switch(config-vlan)# name
VLAN128Char0000000000000000400000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000002

switch(config-vlan)# exit
switch# show vlan id 2

VLAN Name Status Ports
---- -------------------------------- --------- -------------------------------
2 VLAN128Char0000000000000000040000 active
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000002
.
.
.
```

The following example displays the error output if you try to configure a VLAN long name of more than 128 characters.

```
switch# system vlan long-name
switch(config)# vlan 2
switch(config-vlan)# name
129Char12345678900000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000987654321CiscoBangalore

!% String exceeded max length of (128) at '^' marker.!
Switch(config-vlan)# exit
```

The following example displays the error output if you try to configure VLAN name (more than 32 characters) without enabling the **system vlan long- name** command.

```
switch# configure terminal
switch(config)# vlan 2
switch(config-vlan)# name 33Char1234567890987CiscoBangalore
!ERROR: Long VLAN name is not enabled: Vlan name greater than 32 is not allowed!
Switch(config-vlan)# exit
```

Changing the Range of Reserved VLANs

To change the range of reserved VLANs, you must be in global configuration mode. After entering this command, you must do the following tasks:

- Enter the **copy running-config startup-config** command
- Reload the device

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**system vlan** *start-vlan* **reserve** **Example:** `switch(config)# system vlan 3968 reserve`	Allows you to change the reserved VLAN range by specifying the starting VLAN ID for your desired range. You can change the reserved VLANs to any other 128 contiguous VLAN ranges. When you reserve such a range, it frees up the range of VLANs that were allocated for internal use by default, and all of those VLANs are available for user configuration except for VLAN 4094. **Note** To return to the default range of reserved VLANs (3968-4049 and 4094), you must enter the **no system vlan** *start-vlan* **reserve** command.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration. **Note** You must enter this command if you change the reserved block.
Step 4	**reload** **Example:** `switch(config)# reload`	Reloads the software, and modifications to VLAN ranges become effective. For more details about this command, see the *Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 6.x.*
Step 5	(Optional) **show system vlan reserved** **Example:** `switch(config)# show system vlan reserved`	Displays the configured changes to the VLAN range.

Example

This example shows how to change the range of reserved VLANs:

```
switch# configuration terminal
switch(config)# system vlan 2000 reserve
This will delete all configs on vlans 2000-2081. Continue anyway? (y/n) [no] y
Note: After switch reload, VLANs 2000-2081 will be reserved for internal use.
      This requires copy running-config to startup-config before
      switch reload. Creating VLANs within this range is not allowed.
switch(config)#
```

Note You must reload the device for this change to take effect.

Configuring a VLAN

To configure or modify the VLAN for the following parameters, you must be in the VLAN configuration submode:

- Name

Note VLAN name can be either a short name (up to 32 characters) or long name (up to 128 characters). To configure VLAN long-name of up to 128 characters, you must enable **system vlan long-name** command.

- Shut down

Note You cannot create, delete, or modify the default VLAN or the internally allocated VLANs. Additionally, some of these parameters cannot be modified on some VLANs.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** {*vlan-id* | *vlan-range*}	Enters VLAN configuration submode. If the VLAN does not exist, the system first creates the specified VLAN.
Step 3	switch(config-vlan)# **name** *vlan-name*	Names the VLAN. You can enter up to 32 alphanumeric characters to name the VLAN. You cannot change the name of VLAN1 or the internally allocated VLANs. The default value is VLANxxxx where xxxx represents four numeric digits (including leading zeroes) equal to the VLAN ID number.
Step 4	switch(config-vlan)# **state** {**active** | **suspend**}	Sets the state of the VLAN to active or suspend. While the VLAN state is suspended, the ports associated with this VLAN are shut down, and that VLAN does not pass any traffic. The default state is active. You cannot suspend the state for the default VLAN or VLANs 1006 to 4094.
Step 5	(Optional) switch(config-vlan)# **no shutdown**	Enables the VLAN. The default value is **no shutdown** (or enabled). You cannot shut down the default VLAN, VLAN1, or VLANs 1006 to 4094.

Example

This example shows how to configure optional parameters for VLAN 5:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# name accounting
switch(config-vlan)# state active
switch(config-vlan)# no shutdown
```

Adding Ports to a VLAN

After you have completed the configuration of a VLAN, assign ports to it.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {**ethernet** *slot*/*port* | **port-channel** *number*}	Specifies the interface to configure, and enters the interface configuration mode. The interface can be a physical Ethernet port or an EtherChannel.
Step 3	switch(config-if)# **switchport access vlan** *vlan-id*	Sets the access mode of the interface to the specified VLAN.

Example

This example shows how to configure an Ethernet interface to join VLAN 5:

```
switch# configure terminal
switch(config)# interface ethernet 1/13
switch(config-if)# switchport access vlan 5
```

Configuring VTP

You can configure VTP in the client or server mode on Cisco Nexus devices.

You can enable VTP and then configure the VTP mode (server [default], client, transparent, or off). If you enable VTP, you must configure either version 1 or version 2. If you are using VTP in a Token Ring environment, you must use version 2.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature vtp**	Enables VTP on the device. The default is disabled.
Step 3	switch(config)# **vtp domain** *domain-name*	Specifies the name of the VTP domain that you want this device to join. The default is blank.
Step 4	switch(config)# **vtp version** {**1** | **2**}	Sets the VTP version that you want to use. The default is version 1.
Step 5	switch(config)# **vtp file** *file-name*	Specifies the ASCII filename of the IFS file system file where the VTP configuration is stored.
Step 6	switch(config)# **vtp password** *password-value*	Specifies the password for the VTP administrative domain.
Step 7	switch(config)# **exit**	Exits the configuration submode.
Step 8	(Optional) switch# **show vtp status**	Displays information about the VTP configuration on the device, such as the version, mode, and revision number.
Step 9	(Optional) switch# **show vtp counters**	Displays information about VTP advertisement statistics on the device.
Step 10	(Optional) switch# **show vtp interface**	Displays the list of VTP-enabled interfaces.
Step 11	(Optional) switch# **show vtp password**	Displays the password for the management VTP domain.
Step 12	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows the VTP status and that the switch is capable of supporting Version 2 and that the switch is running Version 1:

```
switch(config)# show vtp status
VTP Status Information
----------------------
VTP Version                     : 2 (capable)
Configuration Revision          : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 502
VTP Operating Mode              : Transparent
VTP Domain Name                 :
VTP Pruning Mode                : Disabled (Operationally Disabled)
VTP V2 Mode                     : Disabled
```

```
VTP Traps Generation            : Disabled
MD5 Digest                      : 0xF5 0xF1 0xEC 0xE7 0x29 0x0C 0x2D 0x01
Configuration last modified by 60.10.10.1 at 0-0-00 00:00:00
VTP version running             : 1
```

Configuring VLAN Translation on a Trunk Port

You can configure VLAN translation between the ingress VLAN and a local VLAN on a port. The traffic arriving on the ingress VLAN maps to the local VLAN at the ingress of the trunk port and the traffic that is internally tagged with the translated VLAN ID is mapped back to the original VLAN ID before leaving the switch port.

Before you begin

- Ensure that the physical or port channel on which you want to implement VLAN translation is configured as a Layer 2 trunk port.
- Ensure that the translated VLANs are created on the switch and are also added to the Layer 2 trunk ports trunk-allowed VLAN vlan-list.
- For FEX port-channel trunk interfaces, the last VLAN in the allowed VLAN list must be associated with a translated VLAN in one of the VLAN maps configured on the FEX fabric interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type port*	Enters interface configuration mode.
Step 3	(Optional) switch(config-if)# [**no**] **switchport vlan mapping enable**	Enables VLAN translation on the switch port after VLAN translation is explicitly disabled. VLAN translation is enabled by default. **Note** Use the **no** form of this command to disable VLAN translation.
Step 4	switch(config-if)# [**no**] **switchport vlan mapping** *vlan-id translated-vlan-id*	Translates a VLAN to another VLAN. • The range for both the *vlan-id* and *translated-vlan-id* arguments is from 1 to 4094. • When you configure a VLAN mapping between a VLAN and a (local) VLAN on a port, traffic arriving on the VLAN gets mapped or translated to the local VLAN at the ingress of the switch port, and the traffic internally tagged with the translated VLAN ID gets mapped to the original VLAN ID before leaving the switch port.

	Command or Action	Purpose
		This method of VLAN mapping is a two-way mapping. **Note** Use the **no** form of this command to clear the mappings between a pair of VLANs.
Step 5	switch(config-if)# [**no**] **switchport vlan translation all**	Removes all VLAN translations configured on the interface.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration. **Note** The VLAN translation configuration does not become effective until the switch port becomes an operational trunk port
Step 7	(Optional) switch(config-if)# **show interface** [*if-identifier*] **vlan mapping**	Displays VLAN mapping information for all interfaces or for the specified interface.

Example

This example shows how to configure VLAN translation between (the ingress) VLAN 10 and (the local) VLAN 100:

```
switch# config t
switch(config)# interface ethernet1/1
switch(config-if)# switchport vlan mapping 10 100
switch(config-if)# show interface ethernet1/1 vlan mapping

Interface eth1/1:
Original VLAN          Translated VLAN
------------------     ---------------
10                          100
```

Configuring VLAN Translation with a FEX

VLAN translation on a FEX operates on a per-FEX basis. The VLAN translation enable and mapping configurations must be applied to all the fabric interfaces for a FEX and take effect on all FEX host trunk ports.

You can configure VLAN translation between the ingress/original VLAN and a translated/local VLAN on a FEX trunk port.

For traffic ingressing a FEX trunk port, the original VLAN is mapped to the local VLAN based on the VLAN translations configured on the FEX fabric interfaces. Similarly for traffic egressing a FEX trunk port, the local VLAN is translated to the original VLAN based on the VLAN translation configured on the FEX fabric interfaces.

Note The vlan-list must include the translated VLANs that need to be translated on a FEX trunk interface.

Before you begin

- Ensure that all operational FEX interfaces are configured as Layer 2 trunk ports.
- Ensure that the translated VLANs are created on the switch and that the FEX Layer 2 trunk ports specify the translated VLANs in their trunk allowed vlan-list.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type port*	Specifies an Ethernet interface to configure.
Step 3	switch(config-if)# **channel-group** *number*	Configures port channel parameters.
Step 4	switch(config-if)# **exit**	Exits the configuration submode.
Step 5	switch(config)# **interface** *type port*	Specifies an Ethernet interface to configure.
Step 6	switch(config-if)# **switchport mode fex-fabric**	Set the interface to support an external Fabric Extender.
Step 7	switch(config-if)# **switchport vlan map** *vlan-id translated-id*	*vlan-id* is the ingress. Range is from 1 to 4094. *translated-id* is the local VLAN. Range is from 1 to 4094.
Step 8	switch(config-if)# **fex associate** *number*	Associates a Fabric Extender with a fabric interface.
Step 9	switch(config-if)# **exit**	Exits the configuration submode.
Step 10	switch(config)# **interface** *type port*	Specifies an Ethernet interface to configure. **Note** Applies to the FEX trunk interfaces.
Step 11	switch(config-if)# **switchport mode trunk**	Configures the interface as a trunk port. **Note** Applies to the FEX trunk interfaces.
Step 12	switch(config-if)# **switchport trunk allowed vlan** *vlan-id*	Configures the allowed VLANs for a virtual Ethernet interface.

	Command or Action	Purpose	
		Note	Applies to the FEX trunk interfaces. For FEX port-channel trunk interfaces, the last vlan in the allowed vlan list must be associated with a translated vlan in one of the vlan maps configured on the FEX fabric interface.

Example

This example shows how to configure VLAN translation with a FEX.

```
switch# configure terminal
switch(config)# interface ethernet1/1
switch(config-if)# channel-group 100
switch(config-if)# exit
switch(config)# interface Po100
switch(config-if)# switchport mode fex-fabric
switch(config-if)# switchport vlan map 10 20
switch(config-if)# fex associate 100
switch(config-if)# exit
switch(config)# interface ethernet100/1/1
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk allowed vlan 20
```

Verifying the VLAN Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
switch# **show running-config vlan** [*vlan_id* | *vlan_range*]	Displays VLAN information.
switch# **show vlan** [**brief** | **id** [*vlan_id* | *vlan_range*] | **name** *name* | **summary**]	Displays selected configuration information for the defined VLAN(s).

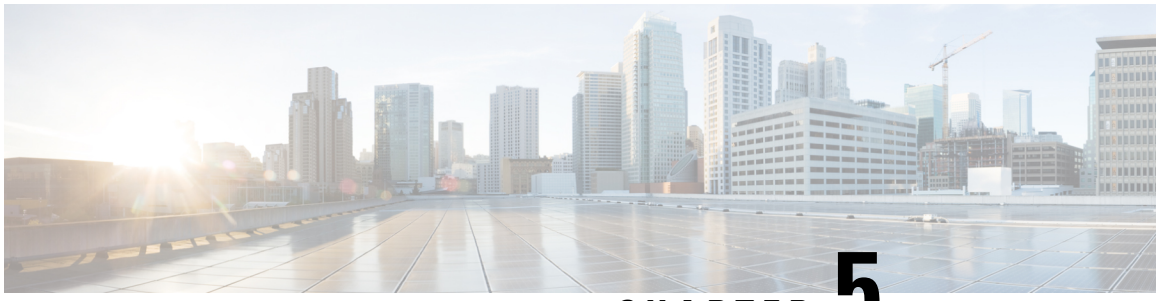
Feature History for Configuring VLANs

This table lists the release history for this feature.

Note The feature history table is added/updated in this guide from Cisco Nexus Release 7.3(0)N1(1) onwards.

Table 4: Feature History for Configuring VLANs

Feature Name	Releases	Feature Information
Configure VLAN long-name.	7.3(0)N1(1)	You can configure VLAN long-names of up to 128 characters. The following command was introduced: • **system vlan long-name**



CHAPTER 5

Configuring Private VLANs

This chapter contains the following sections:

Guidelines and Limitations for Private VLANs

When configuring PVLANs, follow these guidelines:

- You must create a VLAN before you can assign the specified VLAN as a private VLAN.
- You must enable PVLANs before the switch can apply the PVLAN functionality.
- You cannot disable PVLANs if the switch has any operational ports in a PVLAN mode.
- Enter the **private-vlan synchronize** command from within the Multiple Spanning Tree (MST) region definition to map the secondary VLANs to the same MST instance as the primary VLAN.
- You must disable all the FEX isolated trunk ports before configuring FEX trunk ports.
- You cannot connect a second switch to a promiscuous or isolated PVLAN trunk. The promiscuous or isolated PVLAN trunk is supported only on host-switch.
- You cannot configure promiscuous ports and promiscuous trunk ports on the FEX interfaces (HIF) ports.
- If you configure a **private-vlan association** under a VLAN, but do not configure the **private-vlan type** as primary, this association will reappear in the running configuration under the same VLAN when the VLAN is deleted and re-created. Note that this earlier association cannot be removed by using the **no private-vlan association** command. It can be removed only by performing either of the following tasks:
  - Disable the PVLAN feature.

    Or

  - Configure the **private-vlan type** as primary, configure the same **private-vlan association** under that VLAN, and then remove the association using the **no private-vlan association** command.

Limitations with Other Features

Consider the following configuration limitations with other features when configuring private VLANs:

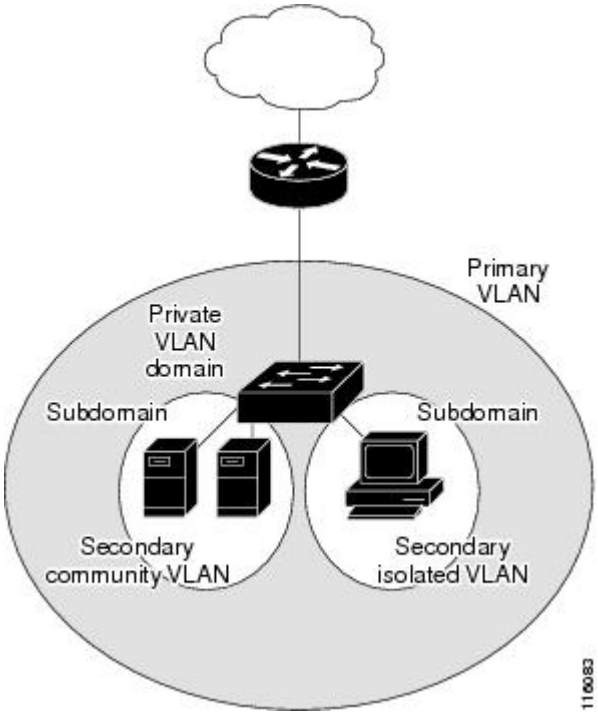
- IGMP snooping runs only on the primary VLAN and uses the configuration of the primary VLAN for all secondary VLANs.

  Any IGMP snooping join request in the secondary VLAN is treated as if it is received in the primary VLAN.

Information About Private VLANs

A private VLAN (PVLAN) partitions the Ethernet broadcast domain of a VLAN into subdomains, allowing you to isolate the ports on the switch from each other. A subdomain consists of a primary VLAN and one or more secondary VLANs (see the following figure). All VLANs in a PVLAN domain share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another. The secondary VLAN can either be isolated VLAN or community VLAN. A host on an isolated VLAN can communicate only with the associated promiscuous port in its primary VLAN. Hosts on community VLAN can communicate among themselves and with their associated promiscuous port but not with ports in other community VLANs.

Figure 7: Private VLAN Domain



Note You must first create the VLAN before converting it to a PVLAN, either a primary VLAN or secondary VLAN.

Primary and Secondary VLANs in Private VLANs

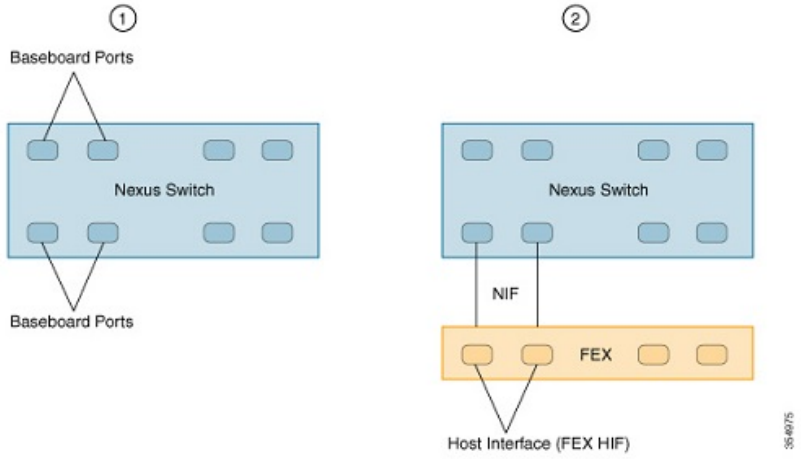
A private VLAN domain has only one primary VLAN. Each port in a private VLAN domain is a member of the primary VLAN; the primary VLAN is the entire private VLAN domain.

Secondary VLAN provide isolation between the ports within the same private VLAN domain. The following two types are secondary VLANs within a primary VLAN:

- Isolated VLANs—Ports within an isolated VLAN cannot communicate directly with each other at the Layer 2 level.
- Community VLANs—Ports within a community VLAN can communicate with each other but cannot communicate with ports in other community VLANs or in any isolated VLANs at the Layer 2 level.

Baseboard Ports and HIF Ports

The following figure shows the baseboard and host interface (HIF) ports on a Cisco Nexus switch.



1	Baseboard ports are ports on a baseboard module in a Cisco Nexus switch.
2	FEX HIF ports are ports on the FEX module.

Associating Secondary VLANs with a Primary Private VLAN

When you associate secondary VLANs with a primary VLAN, follow these guidelines:

- The *secondary-vlan-list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single secondary VLAN ID or a hyphenated range of secondary VLAN IDs.
- The *secondary-vlan-list* parameter can contain multiple community VLAN IDs and one isolated VLAN ID.
- Enter a *secondary-vlan-list* or use the **add** keyword with a *secondary-vlan-list* to associate secondary VLANs with a primary VLAN.
- Use the **remove** keyword with a *secondary-vlan-list* to clear the association between secondary VLANs and a primary VLAN.

- You can change the association between a secondary and primary VLAN by removing the existing association, and then adding the desired association.

If you delete either the primary or secondary VLAN, the VLAN becomes inactive on the port where the association is configured. When you enter the **no private-vlan** command, the VLAN returns to the normal VLAN mode. All primary and secondary associations on that VLAN are suspended, but the interfaces remain in PVLAN mode. If you convert the specified VLAN to PVLAN mode again, the original associations are reinstated.

If you enter the **no vlan** command for the primary VLAN, all the PVLAN associations with that VLAN are lost. However, if you enter the **no vlan** command for a secondary VLAN, the PVLAN associations with that VLAN are suspended and are reinstated when you recreate the specified VLAN and configure it as the previous secondary VLAN.

Before you begin

Ensure that the PVLAN feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** *primary-vlan-id*	Enters the number of the primary VLAN that you are working in for the PVLAN configuration.
Step 3	switch(config-vlan)# **private-vlan association** {[**add**] *secondary-vlan-list* | **remove** *secondary-vlan-list*}	Associates the secondary VLANs with the primary VLAN. Use the **remove** keyword with a *secondary-vlan-list* to clear the association between secondary VLANs and a primary VLAN.
Step 4	(Optional) switch(config-vlan)# **no private-vlan association**	Removes all associations from the primary VLAN and returns it to normal VLAN mode.

Example

The following example shows how to associate community VLANs 100 through 110 and isolated VLAN 200 with primary VLAN 5:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# private-vlan association 100-110, 200
```

Private VLAN Ports

The following are three types of PVLAN ports:

- Promiscuous port—A promiscuous port belongs to a primary VLAN. The promiscuous port can communicate with all interfaces, including the community and isolated host ports, that belong to those

secondary VLANs associated to the promiscuous port and associated with the primary VLAN. You can have several promiscuous ports in a primary VLAN. Each promiscuous port can have several secondary VLANs or no secondary VLANs that are associated to that port. You can associate a secondary VLAN to more than one promiscuous port, as long as the promiscuous port and secondary VLANs are within the same primary VLAN. You may want to do this for load-balancing or redundancy purposes. You can also have secondary VLANs that are not associated to any promiscuous port.

A promiscuous port can be configured either as an access port or as a trunk port.

- Isolated port—An isolated port is a host port that belongs to an isolated secondary VLAN. This port has complete isolation from other ports within the same PVLAN domain, except that it can communicate with associated promiscuous ports. PVLANs block all the traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports. You can have more than one isolated port in a specified isolated VLAN. Each port is completely isolated from all other ports in the isolated VLAN.

  An isolated port can be configured as either an access port or a trunk port.

- Community port—A community port is a host port that belongs to a community secondary VLAN. Community ports communicate with other ports in the same community VLAN and with associated promiscuous ports. These interfaces are isolated from all other interfaces in other communities and from all isolated ports within the PVLAN domain.

  A community port must be configured as an access port. A community VLAN must not be enabled on an isolated trunk port.

Note Because trunks can support VLANs that carry traffic between promiscuous, isolated, and community ports, the isolated and community port traffic might enter or leave the switch through a trunk interface.

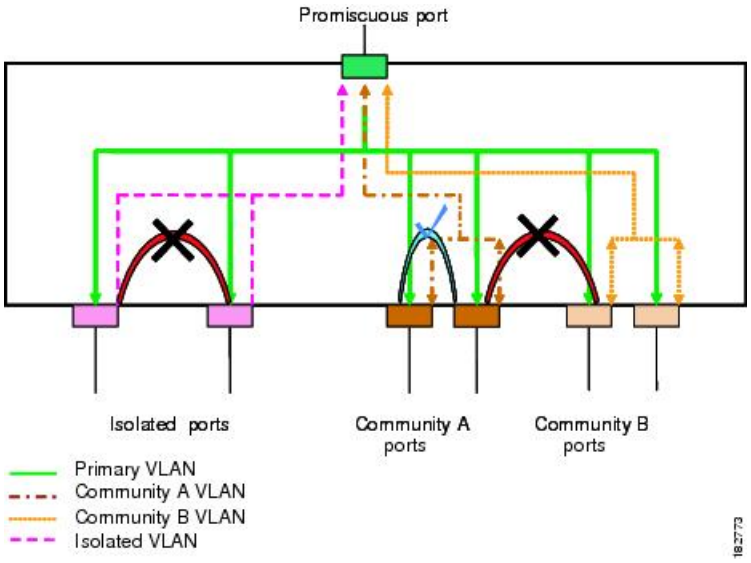
Primary, Isolated, and Community Private VLANs

Primary VLANs and the two types of secondary VLANs (isolated and community) have the following characteristics:

- Primary VLAN—The primary VLAN carries traffic from the promiscuous ports to the host ports, both isolated and community, and to other promiscuous ports.
- Isolated VLAN—An isolated VLAN is a secondary VLAN that carries unidirectional traffic upstream from the hosts toward the promiscuous ports. You can configure only one isolated VLAN in a PVLAN domain. An isolated VLAN can have several isolated ports. The traffic from each isolated port also remains completely separate.
- Community VLAN—A community VLAN is a secondary VLAN that carries upstream traffic from the community ports to the promiscuous port and to other host ports in the same community. You can configure multiple community VLANs in a PVLAN domain. The ports within one community can communicate, but these ports cannot communicate with ports in any other community or isolated VLAN in the private VLAN.

The following figure shows the traffic flow within a PVLAN, along with the types of VLANs and types of ports.

Figure 8: Private VLAN Traffic Flows



Note The PVLAN traffic flows are unidirectional from the host ports to the promiscuous ports. Traffic received on primary VLAN enforces no separation and forwarding is done as in a normal VLAN.

A promiscuous access port can serve only one primary VLAN and multiple secondary VLANs (community and isolated VLANs). A promiscuous trunk port can carry traffic for several primary VLANs. Multiple secondary VLANs under a given primary VLAN can be mapped to promiscuous trunk ports. With a promiscuous port, you can connect a wide range of devices as access points to a PVLAN. For example, you can use a promiscuous port to monitor or back up all the PVLAN servers from an administration workstation.

In a switched environment, you can assign an individual PVLAN and associated IP subnet to each individual or common group of end stations. The end stations need to communicate only with a default gateway to communicate outside the private VLAN.

Associating Primary and Secondary VLANs

To allow host ports in secondary VLANs to communicate outside the PVLAN, you associate secondary VLANs to the primary VLAN. If the association is not operational, the host ports (community and isolated ports) in the secondary VLAN are brought down.

Note You can associate a secondary VLAN with only one primary VLAN.

For an association to be operational, the following conditions must be met:

- The primary VLAN must exist and be configured as a primary VLAN.
- The secondary VLAN must exist and be configured as either an isolated or community VLAN.

Note Use the **show vlan private-vlan** command to verify that the association is operational. The switch does not display an error message when the association is nonoperational.

If you delete either the primary or secondary VLAN, the ports that are associated with the VLAN become inactive. Use the **no private-vlan** command to return the VLAN to the normal mode. All primary and secondary associations on that VLAN are suspended, but the interfaces remain in PVLAN mode. When you convert the VLAN back to PVLAN mode, the original associations are reinstated.

If you enter the **no vlan** command for the primary VLAN, all PVLAN associations with that VLAN are deleted. However, if you enter the **no vlan** command for a secondary VLAN, the PVLAN associations with that VLAN are suspended and are restored when you recreate the specified VLAN and configure it as the previous secondary VLAN.

In order to change the association between a secondary and primary VLAN, you must first remove the current association and then add the desired association.

Private VLAN Promiscuous Trunks

A promiscuous trunk port can carry traffic for several primary VLANs. Multiple secondary VLANs under a given primary VLAN can be mapped to a promiscuous trunk port. Traffic on the promiscuous port is received and transmitted with a primary VLAN tag.

Private VLAN Isolated Trunks

An isolated trunk port can carry traffic for multiple isolated PVLANs. Traffic for a community VLAN is not carried by isolated trunk ports. Traffic on isolated trunk ports is received and transmitted with an isolated VLAN tag. Isolated trunk ports are intended to be connected to host servers.

To support isolated PVLAN ports on a Cisco Nexus Fabric Extender, the Cisco Nexus device must prevent communication between the isolated ports on the FEX; all forwarding occurs through the switch.

Caution You must disable all the FEX isolated trunk ports before configuring PVLANs on the FEX trunk ports. If the FEX isolated trunk ports and the FEX trunk ports are both enabled, unwanted network traffic might occur.

For unicast traffic, you can prevent such a communication without any side effects.

For multicast traffic, the FEX provides replication of the frames. To prevent communication between isolated PVLAN ports on the FEX, the switch prevents multicast frames from being sent back through the fabric ports. This restriction prevents communication between an isolated VLAN and a promiscuous port on the FEX. However, as host interfaces are not intended to be connected to another switch or router, you cannot enable a promiscuous port on a FEX.

Broadcast Traffic in Private VLANs

Broadcast traffic from ports in a private VLAN flows in the following ways:

- The broadcast traffic flows from a promiscuous port to all ports in the primary VLAN (which includes all the ports in the community and isolated VLANs). This broadcast traffic is distributed to all ports within the primary VLAN, including those ports that are not configured with private VLAN parameters.
- The broadcast traffic from an isolated port is distributed only to those promiscuous ports in the primary VLAN that are associated to that isolated port.
- The broadcast traffic from community ports is distributed to all ports within the port's community and to all promiscuous ports that are associated to the community port. The broadcast packets are not distributed to any other communities within the primary VLAN or to any isolated ports.

Private VLAN Port Isolation

You can use PVLANs to control access to end stations as follows:

- Configure selected interfaces connected to end stations as isolated ports to prevent any communication. For example, if the end stations are servers, this configuration prevents communication between the servers.
- Configure interfaces connected to default gateways and selected end stations (for example, backup servers) as promiscuous ports to allow all end stations access to a default gateway.

Configuring a Private VLAN

Enabling Private VLANs

You must enable PVLANs on the switch to use the PVLAN functionality.

Note The PVLAN commands do not appear until you enable the PVLAN feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature private-vlan**	Enables the PVLAN feature on the switch.
Step 3	(Optional) switch(config)# **no feature private-vlan**	Disables the PVLAN feature on the switch. **Note** You cannot disable PVLANs if there are operational ports on the switch that are in PVLAN mode.

Example

This example shows how to enable the PVLAN feature on the switch:

```
switch# configure terminal
switch(config)# feature private-vlan
```

Configuring a VLAN as a Private VLAN

To create a PVLAN, you must first create a VLAN, and then configure that VLAN to be a PVLAN.

Before you begin

Ensure that the PVLAN feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** {*vlan-id* | *vlan-range*}	Enters VLAN configuration submode.
Step 3	switch(config-vlan)# **private-vlan** {**community** | **isolated** | **primary**}	Configures the VLAN as either a community, isolated, or primary PVLAN. In a PVLAN, you must have one primary VLAN. You can have multiple community and isolated VLANs.
Step 4	(Optional) switch(config-vlan)# **no private-vlan** {**community** | **isolated** | **primary**}	Removes the PVLAN configuration from the specified VLAN(s) and returns it to normal VLAN mode. If you delete either the primary or secondary VLAN, the ports that are associated with the VLAN become inactive.

Example

The following example shows how to assign VLAN 5 to a PVLAN as the primary VLAN:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# private-vlan primary
```

The following example shows how to assign VLAN 100 to a PVLAN as a community VLAN:

```
switch# configure terminal
switch(config)# vlan 100
switch(config-vlan)# private-vlan community
```

The following example shows how to assign VLAN 200 to a PVLAN as an isolated VLAN:

```
switch# configure terminal
switch(config)# vlan 200
switch(config-vlan)# private-vlan isolated
```

Configuring an Interface as a Private VLAN Host Port

In PVLANs, host ports are part of the secondary VLANs, which are either community VLANs or isolated VLANs. Configuring a PVLAN host port involves two steps. First, you define the port as a PVLAN host port and then you configure a host association between the primary and secondary VLANs.

Note We recommend that you enable BPDU Guard on all interfaces configured as a host ports.

Before you begin

Ensure that the PVLAN feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type* [*chassis*/]*slot*/*port*	Selects the port to configure as a PVLAN host port. This port can be on a FEX (identified by the chassis option).
Step 3	switch(config-if)# **switchport**	Configures the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.
Step 4	switch(config-if)# **switchport mode private-vlan host**	Configures the port as a host port for a PVLAN.
Step 5	switch(config-if)# **switchport private-vlan host-association** {*primary-vlan-id*} {*secondary-vlan-id*}	Associates the port with the primary and secondary VLANs of a PVLAN. The secondary VLAN can be either an isolated or community VLAN.
Step 6	(Optional) switch(config-if)# **no switchport private-vlan host-association**	Removes the PVLAN association from the port.

Example

This example shows how to configure Ethernet port 1/12 as a host port for a PVLAN and associate it to primary VLAN 5 and secondary VLAN 101:

```
switch# configure terminal
switch(config)# interface ethernet 1/12
switch(config-if)# switchport
switch(config-if)# switchport mode private-vlan host
switch(config-if)# switchport private-vlan host-association 5 101
```

Configuring an Interface as a Private VLAN Promiscuous Port

In a PVLAN domain, promiscuous ports are part of the primary VLAN. Configuring a promiscuous port involves two steps. First, you define the port as a promiscuous port and then you configure the mapping between a secondary VLAN and the primary VLAN.

Before you begin

Ensure that the PVLAN feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Selects the port to configure as a PVLAN promiscuous port. A base-board interface is required. This port cannot be on a FEX interface (HIF interface). **Note** If this is a QSFP+ GEM or a breakout port, the *port* syntax is *QSFP-module*/*port*.
Step 3	switch(config-if)# **switchport**	Configures the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.
Step 4	switch(config-if)# **switchport mode private-vlan promiscuous**	Configures the port as a promiscuous port for a PVLAN. You can enable promiscuous ports and promiscuous trunk ports only on base-board ports (base-board ports are the ports on the switch). You cannot configure promiscuous ports on FEX (HIF) ports. **Note** If you try to configure promiscuous ports on FEX (HIF) ports, the device will display an error.
Step 5	switch(config-if)# **switchport private-vlan mapping** {*primary-vlan-id*} {*secondary-vlan-list* | **add** *secondary-vlan-list* | **remove** *secondary-vlan-list*}	Configures the port as a promiscuous port and associates the specified port with a primary VLAN and a selected list of secondary VLANs. The secondary VLAN can be either an isolated or community VLAN.
Step 6	(Optional) switch(config-if)# **no switchport private-vlan mapping**	Clears the mapping from the PVLAN.

Example

The following example shows how to configure Ethernet interface 1/4 as a promiscuous port associated with primary VLAN 5 and secondary VLAN 200:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport
switch(config-if)# switchport mode private-vlan promiscuous
switch(config-if)# switchport private-vlan mapping 5 200
```

Configuring a Promiscuous Trunk Port

In a PVLAN domain, promiscuous trunks are part of the primary VLAN. Promiscuous trunk ports can carry multiple primary VLANs. Multiple secondary VLANs under a given primary VLAN can be mapped to a promiscuous trunk port.

Configuring a promiscuous port involves two steps. First, you define the port as a promiscuous port and then you configure the mapping between a secondary VLAN and the primary VLAN. Multiple primary VLANs can be enabled by configuring multiple mappings.

Note The number of mappings on a PVLAN trunk port is limited to 128.

Before you begin

Ensure that the PVLAN feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Selects the port to configure as a PVLAN promiscuous trunk port. A base-board interface is required. This port cannot be on a FEX interface (HIF interface). **Note** If this is a QSFP+ GEM or a breakout port, the *port* syntax is *QSFP-module*/*port*.
Step 3	switch(config-if)# **switchport**	Configures the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.
Step 4	switch(config-if)# **switchport mode private-vlan trunk promiscuous**	Configures the port as a promiscuous trunk port for a PVLAN. You can enable promiscuous trunk ports only on base-board ports (base-board ports are the ports on the switch). You cannot

	Command or Action	Purpose
		configure promiscuous trunk ports on FEX (HIF) ports. **Note** If you try to configure promiscuous trunk ports on FEX (HIF) ports, the device will display an error.
Step 5	switch(config-if)# **switchport private-vlan mapping trunk** {*primary-vlan-id*} {*secondary-vlan-id*}	Maps the trunk port with the primary and secondary VLANs of a PVLAN. The secondary VLAN can be either an isolated or community VLAN.
Step 6	(Optional) switch(config-if)# **no switchport private-vlan mapping trunk** [*primary-vlan-id*]	Removes the PVLAN mapping from the port. If the *primary-vlan-id* is not supplied, all PVLAN mappings are removed from the port.

Example

The following example shows how to configure Ethernet interface 1/1 as a promiscuous trunk port for a PVLAN and then map the secondary VLANs to the primary VLAN:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport
switch(config-if)# switchport mode private-vlan trunk promiscuous
switch(config-if)# switchport private-vlan mapping trunk 5 100
switch(config-if)# switchport private-vlan mapping trunk 5 200
switch(config-if)# switchport private-vlan mapping trunk 6 300
```

Configuring an Isolated Trunk Port

In a PVLAN domain, isolated trunks are part of a secondary VLAN. Isolated trunk ports can carry multiple isolated VLANs. Configuring an isolated trunk port involves two steps. First, you define the port as an isolated trunk port and then you configure the association between the isolated and primary VLANs. Multiple isolated VLANs can be enabled by configuring multiple associations.

Before you begin

Ensure that the PVLAN feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type* [*chassis*/]*slot*/*port*	Selects the port to configure as a PVLAN isolated trunk port. This port can be on a FEX (identified by the chassis option). The PVLAN isolated trunk port can be configured on a Ethernet port and on a FEX port.

	Command or Action	Purpose
		Note If this is a QSFP+ GEM or a breakout port, the *port* syntax is *QSFP-module*/*port*.
Step 3	switch(config-if)# **switchport**	Configures the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.
Step 4	switch(config-if)# **switchport mode private-vlan trunk** [**secondary**]	Configures the port as a secondary trunk port for a PVLAN. **Note** The **secondary** keyword is assumed if it is not present.
Step 5	switch(config-if)# **switchport private-vlan association trunk** {*primary-vlan-id*} {*secondary-vlan-id*}	Associates the isolated trunk port with the primary and secondary VLANs of a PVLAN. The secondary VLAN should be an isolated VLAN. Only one isolated VLAN can be mapped under a given primary VLAN.
Step 6	(Optional) switch(config-if)# **no switchport private-vlan association trunk** [*primary-vlan-id*]	Removes the PVLAN association from the port. If the *primary-vlan-id* is not supplied, all PVLAN associations are removed from the port.

Example

The following example shows how to configure Ethernet interface 1/1 as an isolated trunk port for a PVLAN and then associate the secondary VLANs to the primary VLAN:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport
switch(config-if)# switchport mode private-vlan trunk secondary
switch(config-if)# switchport private-vlan association trunk 5 100
switch(config-if)# switchport private-vlan association trunk 6 200
```

Configuring the Allowed VLANs for PVLAN Trunking Ports

Isolated trunk and promiscuous trunk ports can carry traffic from regular VLANs along with PVLANs.

Before you begin

Ensure that the PVLAN feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **interface** *type* [*chassis*/]*slot*/*port*	Selects the port to configure as a PVLAN host port. This port can be on a FEX (identified by the chassis option).
Step 3	switch(config-if)# **switchport**	Configures the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.
Step 4	switch(config-if)# **switchport private-vlan trunk allowed vlan** {*vlan-list* | **all** | **none** [**add** | **except** | **none** | **remove** {*vlan-list*}]}	Sets the allowed VLANs for the private trunk interface. The default is to allow only mapped/associated VLANs on the PVLAN trunk interface. **Note** The primary VLANs do not need to be explicitly added to the allowed VLAN list. They are added automatically once there is a mapping between primary and secondary VLANs.

Example

The following example shows how to add VLANs to the list of allowed VLANs on an Ethernet PVLAN trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport
switch(config-if)# switchport private-vlan trunk allowed vlan 15-20
```

Configuring Native 802.1Q VLANs on Private VLANs

Typically, you configure 802.1Q trunks with a native VLAN ID, which strips tagging from all packets on that VLAN. This configuration allows untagged traffic and control traffic to transit the . Secondary VLANs cannot be configured with a native VLAN ID on promiscuous trunk ports. Primary VLANs cannot be configured with a native VLAN ID on isolated trunk ports.

Note A trunk can carry the traffic of multiple VLANs. Traffic that belongs to the native VLAN is not encapsulated to transit the trunk. Traffic for other VLANs is encapsulated with tags that identify the VLAN that the traffic belongs to.

Before you begin

Ensure that the PVLAN feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type* [*chassis*/]*slot*/*port*	Selects the port to configure as a PVLAN host port. This port can be on a FEX (identified by the chassis option).
Step 3	switch(config-if)# **switchport**	Configures the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.
Step 4	switch(config-if)# **switchport private-vlan trunk native** {**vlan** *vlan-id*}	Sets the native VLAN ID for the PVLAN trunk. The default is VLAN 1.
Step 5	(Optional) switch(config-if)# **no switchport private-vlan trunk native** {**vlan** *vlan-id*}	Removes the native VLAN ID from the PVLAN trunk.

Verifying the Private VLAN Configuration

Use the following commands to display PVLAN configuration information.

Command	Purpose
switch# **show feature**	Displays the features enabled on the switch.
switch# **show interface switchport**	Displays information on all interfaces configured as switch ports.
switch# **show vlan private-vlan** [**type**]	Displays the status of the PVLAN.

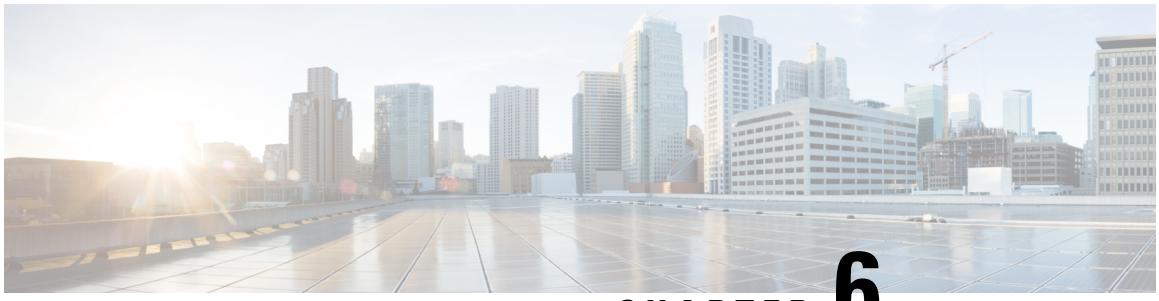
This example shows how to display the PVLAN configuration:

```
switch# show vlan private-vlan
Primary  Secondary  Type             Ports
-------  ---------  ---------------  ------------------------------------------
5        100        community
5        101        community        Eth1/12, Eth100/1/1
5        102        community
5        110        community
5        200        isolated         Eth1/2
switch# show vlan private-vlan type
Vlan Type
---- -----------------
5    primary
100  community
101  community
```

```
102  community
110  community
200  isolated
```

This example shows how to display enabled features (some of the output has been removed for brevity):

```
switch# show feature
Feature Name          Instance  State
--------------------  --------  --------
fcsp                  1         enabled
...
interface-vlan        1         enabled
private-vlan          1         enabled
udld                  1         disabled
...
```



CHAPTER 6

Configuring Rapid PVST+

This chapter contains the following sections:

Information About Rapid PVST+

The Rapid PVST+ protocol is the IEEE 802.1w standard, Rapid Spanning Tree Protocol (RSTP), implemented on a per VLAN basis. Rapid PVST+ interoperates with the IEEE 802.1D standard, which mandates a single STP instance for all VLANs, rather than per VLAN.

Rapid PVST+ is enabled by default on the default VLAN (VLAN1) and on all newly created VLANs in the software. Rapid PVST+ interoperates with switches that run legacy IEEE 802.1D STP.

RSTP is an improvement on the original STP standard, 802.1D, which allows faster convergence.

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

Understanding STP

STP Overview

For an Ethernet network to function properly, only one active path can exist between any two stations.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The STP algorithm calculates the best loop-free path throughout a switched network. LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Switches do not forward these frames but use the frames to construct a loop-free path.

Multiple active paths between end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages and switches might learn end station MAC addresses on multiple LAN ports. These conditions result in a broadcast storm, which creates an unstable network.

STP defines a tree with a root bridge and a loop-free path from the root to all switches in the network. STP forces redundant data paths into a blocked state. If a network segment in the spanning tree fails and a redundant path exists, the STP algorithm recalculates the spanning tree topology and activates the blocked path.

When two LAN ports on a switch are part of a loop, the STP port priority and port path cost setting determine which port on the switch is put in the forwarding state and which port is put in the blocking state.

Understanding How a Topology is Created

All switches in an extended LAN that participate in a spanning tree gather information about other switches in the network by exchanging of BPDUs. This exchange of BPDUs results in the following actions:

- The system elects a unique root switch for the spanning tree network topology.
- The system elects a designated switch for each LAN segment.
- The system eliminates any loops in the switched network by placing redundant interfaces in a backup state; all paths that are not needed to reach the root switch from anywhere in the switched network are placed in an STP-blocked state.

The topology on an active switched network is determined by the following:

- The unique switch identifier Media Access Control (MAC) address of the switch that is associated with each switch
- The path cost to the root that is associated with each interface
- The port identifier that is associated with each interface

In a switched network, the root switch is the logical center of the spanning tree topology. STP uses BPDUs to elect the root switch and root port for the switched network, as well as the root port and designated port for each switched segment.

Understanding the Bridge ID

Each VLAN on each switch has a unique 64-bit bridge ID that consists of a bridge priority value, an extended system ID (IEEE 802.1t), and an STP MAC address allocation.

Bridge Priority Value

The bridge priority is a 4-bit value when the extended system ID is enabled.

Note In Cisco NX-OS, the extended system ID is always enabled; you cannot disable the extended system ID.

Extended System ID

A 12-bit extended system ID field is part of the bridge ID.

Figure 9: Bridge ID with Extended System ID

The switches always use the 12-bit extended system ID.

Combined with the bridge ID, the system ID extension functions as the unique identifier for a VLAN.

Table 5: Bridge Priority Value and Extended System ID with the Extended System ID Enabled

Bridge Priority Value				Extended System ID (Set Equal to the VLAN ID)											
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

STP MAC Address Allocation

Note Extended system ID and MAC address reduction is always enabled on the software.

With MAC address reduction enabled on any switch, you should also enable MAC address reduction on all other connected switches to avoid undesirable root bridge election and spanning tree topology issues.

When MAC address reduction is enabled, the root bridge priority becomes a multiple of 4096 plus the VLAN ID. You can only specify a switch bridge ID (used by the spanning tree algorithm to determine the identity of the root bridge, the lowest being preferred) as a multiple of 4096. Only the following values are possible:

- 0
- 4096
- 8192
- 12288
- 16384
- 20480
- 24576
- 28672
- 32768
- 36864
- 40960
- 45056

- 49152
- 53248
- 57344
- 61440

STP uses the extended system ID plus a MAC address to make the bridge ID unique for each VLAN.

Note If another bridge in the same spanning tree domain does not run the MAC address reduction feature, it could achieve root bridge ownership because its bridge ID may fall between the values specified by the MAC address reduction feature.

Understanding BPDUs

Switches transmit bridge protocol data units (BPDUs) throughout the STP instance. Each switch sends configuration BPDUs to communicate and compute the spanning tree topology. Each configuration BPDU contains the following minimal information:

- The unique bridge ID of the switch that the transmitting switch determines is the root bridge
- The STP path cost to the root
- The bridge ID of the transmitting bridge
- Message age
- The identifier of the transmitting port
- Values for the hello, forward delay, and max-age protocol timer
- Additional information for STP extension protocols

When a switch transmits a Rapid PVST+ BPDU frame, all switches connected to the VLAN on which the frame is transmitted receive the BPDU. When a switch receives a BPDU, it does not forward the frame but instead uses the information in the frame to calculate a BPDU, and, if the topology changes, initiate a BPDU transmission.

A BPDU exchange results in the following:

- One switch is elected as the root bridge.
- The shortest distance to the root bridge is calculated for each switch based on the path cost.
- A designated bridge for each LAN segment is selected. This is the switch closest to the root bridge through which frames are forwarded to the root.
- A root port is selected. This is the port providing the best path from the bridge to the root bridge.
- Ports included in the spanning tree are selected.

Election of the Root Bridge

For each VLAN, the switch with the lowest numerical value of the bridge ID is elected as the root bridge. If all switches are configured with the default priority (32768), the switch with the lowest MAC address in the VLAN becomes the root bridge. The bridge priority value occupies the most significant bits of the bridge ID.

When you change the bridge priority value, you change the probability that the switch will be elected as the root bridge. Configuring a lower value increases the probability; a higher value decreases the probability.

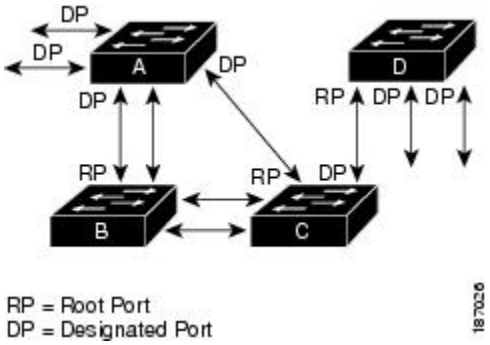
The STP root bridge is the logical center of each spanning tree topology in a network. All paths that are not needed to reach the root bridge from anywhere in the network are placed in STP blocking mode.

BPDUs contain information about the transmitting bridge and its ports, including bridge and MAC addresses, bridge priority, port priority, and path cost. STP uses this information to elect the root bridge for the STP instance, to elect the root port leading to the root bridge, and to determine the designated port for each segment.

Creating the Spanning Tree Topology

In the following figure, Switch A is elected as the root bridge because the bridge priority of all the switches is set to the default (32768) and Switch A has the lowest MAC address. However, due to traffic patterns, the number of forwarding ports, or link types, Switch A might not be the ideal root bridge. By increasing the priority (lowering the numerical value) of the ideal switch so that it becomes the root bridge, you force an STP recalculation to form a new spanning tree topology with the ideal switch as the root.

Figure 10: Spanning Tree Topology



When the spanning tree topology is calculated based on default parameters, the path between source and destination end stations in a switched network might not be ideal. For instance, connecting higher-speed links to a port that has a higher number than the current root port can cause a root-port change. The goal is to make the fastest link the root port.

For example, assume that one port on Switch B is a fiber-optic link, and another port on Switch B (an unshielded twisted-pair [UTP] link) is the root port. Network traffic might be more efficient over the high-speed fiber-optic link. By changing the STP port priority on the fiber-optic port to a higher priority (lower numerical value) than the root port, the fiber-optic port becomes the new root port.

Understanding Rapid PVST+

Rapid PVST+ Overview

Rapid PVST+ is the IEEE 802.1w (RSTP) standard implemented per VLAN. A single instance of STP runs on each configured VLAN (if you do not manually disable STP). Each Rapid PVST+ instance on a VLAN

has a single root switch. You can enable and disable STP on a per-VLAN basis when you are running Rapid PVST+.

Note Rapid PVST+ is the default STP mode for the switch.

Rapid PVST+ uses point-to-point wiring to provide rapid convergence of the spanning tree. The spanning tree reconfiguration can occur in less than 1 second with Rapid PVST+ (in contrast to 50 seconds with the default settings in the 802.1D STP).

Note Rapid PVST+ supports one STP instance for each VLAN.

Using Rapid PVST+, STP convergence occurs rapidly. Each designated or root port in the STP sends out a BPDU every 2 seconds by default. On a designated or root port in the topology, if hello messages are missed three consecutive times, or if the maximum age expires, the port immediately flushes all protocol information in the table. A port considers that it loses connectivity to its direct neighbor root or designated port if it misses three BPDUs or if the maximum age expires. This rapid aging of the protocol information allows quick failure detection. The switch automatically checks the PVID.

Rapid PVST+ provides for rapid recovery of connectivity following the failure of a network device, a switch port, or a LAN. It provides rapid convergence for edge ports, new root ports, and ports connected through point-to-point links as follows:

- Edge ports—When you configure a port as an edge port on an RSTP switch, the edge port immediately transitions to the forwarding state. (This immediate transition was previously a Cisco-proprietary feature named PortFast.) You should only configure on ports that connect to a single end station as edge ports. Edge ports do not generate topology changes when the link changes.

  Enter the **spanning-tree port type** interface configuration command to configure a port as an STP edge port.

  Note We recommend that you configure all ports connected to a host as edge ports.

- Root ports—If Rapid PVST+ selects a new root port, it blocks the old root port and immediately transitions the new root port to the forwarding state.
- Point-to-point links—If you connect a port to another port through a point-to-point link and the local port becomes a designated port, it negotiates a rapid transition with the other port by using the proposal-agreement handshake to ensure a loop-free topology.

Rapid PVST+ achieves rapid transition to the forwarding state only on edge ports and point-to-point links. Although the link type is configurable, the system automatically derives the link type information from the duplex setting of the port. Full-duplex ports are assumed to be point-to-point ports, while half-duplex ports are assumed to be shared ports.

Edge ports do not generate topology changes, but all other designated and root ports generate a topology change (TC) BPDU when they either fail to receive three consecutive BPDUs from the directly connected neighbor or the maximum age times out. At this point, the designated or root port sends out a BPDU with the TC flag set. The BPDUs continue to set the TC flag as long as the TC While timer runs on that port. The value

of the TC While timer is the value set for the hello time plus 1 second. The initial detector of the topology change immediately floods this information throughout the entire topology.

When Rapid PVST+ detects a topology change, the protocol does the following:

- Starts the TC While timer with a value equal to twice the hello time for all the non-edge root and designated ports, if necessary.
- Flushes the MAC addresses associated with all these ports.

The topology change notification floods quickly across the entire topology. The system flushes dynamic entries immediately on a per-port basis when it receives a topology change.

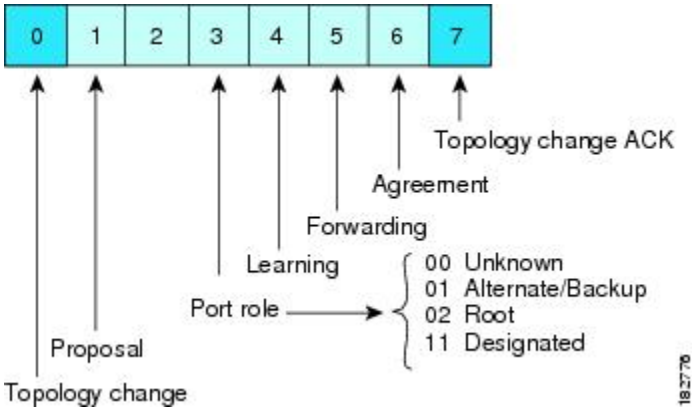
Note The TCA flag is used only when the switch is interacting with switches that are running legacy 802.1D STP.

The proposal and agreement sequence then quickly propagates toward the edge of the network and quickly restores connectivity after a topology change.

Rapid PVST+ BPDUs

Rapid PVST+ and 802.1w use all six bits of the flag byte to add the role and state of the port that originates the BPDU and the proposal and agreement handshake. The following figure shows the use of the BPDU flags in Rapid PVST+.

Figure 11: Rapid PVST+ Flag Byte in BPDU

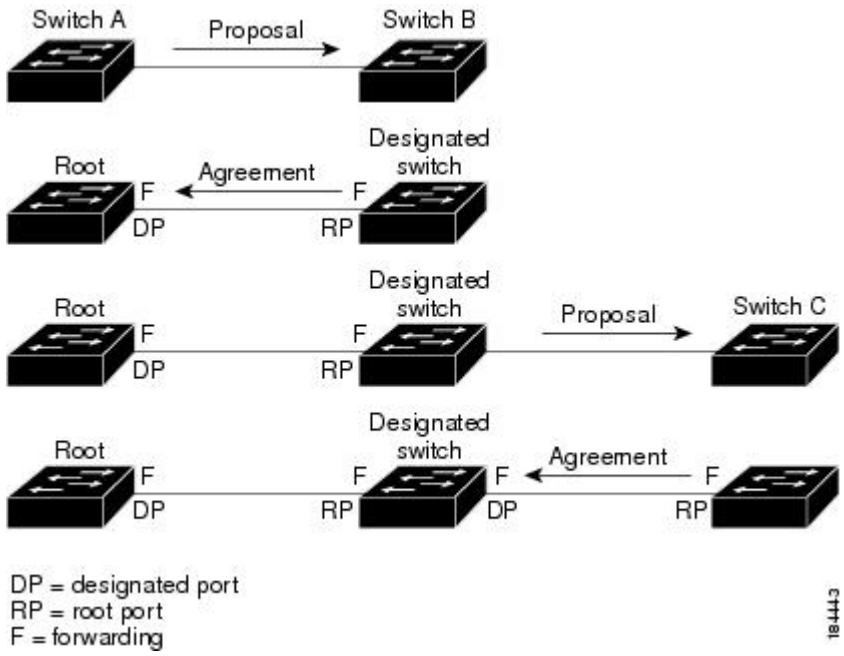


Another important change is that the Rapid PVST+ BPDU is type 2, version 2, which makes it possible for the switch to detect connected legacy (802.1D) bridges. The BPDU for 802.1D is version 0.

Proposal and Agreement Handshake

As shown in the following figure, Switch A is connected to Switch B through a point-to-point link, and all of the ports are in the blocking state. Assume that the priority of Switch A is a smaller numerical value than the priority of Switch B.

Figure 12: Proposal and Agreement Handshaking for Rapid Convergence



Switch A sends a proposal message (a configuration BPDU with the proposal flag set) to Switch B, proposing itself as the designated switch.

After receiving the proposal message, Switch B selects as its new root port the port from which the proposal message was received, forces all non-edge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

After receiving the agreement message from Switch B, Switch A also immediately transitions its designated port to the forwarding state. No loops in the network can form because Switch B blocked all of its non-edge ports and because there is a point-to-point link between Switches A and B.

When Switch C connects to Switch B, a similar set of handshaking messages are exchanged. Switch C selects the port connected to Switch B as its root port, and both ends of the link immediately transition to the forwarding state. With each iteration of this handshaking process, one more network device joins the active topology. As the network converges, this proposal-agreement handshaking progresses from the root toward the leaves of the spanning tree.

The switch learns the link type from the port duplex mode: a full-duplex port is considered to have a point-to-point connection and a half-duplex port is considered to have a shared connection. You can override the default setting that is controlled by the duplex setting by entering the **spanning-tree link-type** interface configuration command.

This proposal/agreement handshake is initiated only when a non-edge port moves from the blocking to the forwarding state. The handshaking process then proliferates step-by-step throughout the topology.

Protocol Timers

The following table describes the protocol timers that affect the Rapid PVST+ performance.

Table 6: Rapid PVST+ Protocol Timers

Variable	Description
Hello timer	Determines how often each switch broadcasts BPDUs to other switches. The default is 2 seconds, and the range is from 1 to 10.
Forward delay timer	Determines how long each of the listening and learning states last before the port begins forwarding. This timer is generally not used by the protocol but is used as a backup. The default is 15 seconds, and the range is from 4 to 30 seconds.
Maximum age timer	Determines the amount of time protocol information received on an port is stored by the switch. This timer is generally not used by the protocol, but it is used when interoperating with 802.1D spanning tree. The default is 20 seconds; the range is from 6 to 40 seconds.

Port Roles

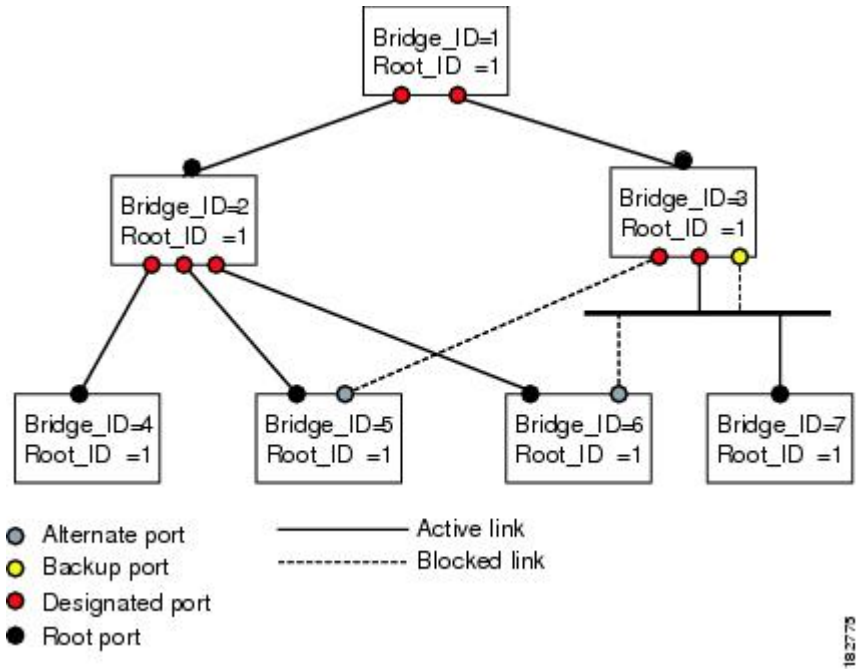
Rapid PVST+ provides rapid convergence of the spanning tree by assigning port roles and learning the active topology. Rapid PVST+ builds upon the 802.1D STP to select the switch with the highest priority (lowest numerical priority value) as the root bridge. Rapid PVST+ then assigns one of these port roles to individual ports:

- Root port—Provides the best path (lowest cost) when the switch forwards packets to the root bridge.
- Designated port—Connects to the designated switch, which incurs the lowest path cost when forwarding packets from that LAN to the root bridge. The port through which the designated switch is attached to the LAN is called the designated port.
- Alternate port—Offers an alternate path toward the root bridge to the path provided by the current root port. An alternate port provides a path to another switch in the topology.
- Backup port—Acts as a backup for the path provided by a designated port toward the leaves of the spanning tree. A backup port can exist only when two ports are connected in a loopback by a point-to-point link or when a switch has two or more connections to a shared LAN segment. A backup port provides another path in the topology to the switch.
- Disabled port—Has no role within the operation of the spanning tree.

In a stable topology with consistent port roles throughout the network, Rapid PVST+ ensures that every root port and designated port immediately transition to the forwarding state while all alternate and backup ports are always in the blocking state. Designated ports start in the blocking state. The port state controls the operation of the forwarding and learning processes.

A port with the root or a designated port role is included in the active topology. A port with the alternate or backup port role is excluded from the active topology (see the following figure).

Figure 13: Sample Topology Demonstrating Port Roles



Port States

Rapid PVST+ Port State Overview

Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When a LAN port transitions directly from nonparticipation in the spanning tree topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate through the switched LAN before starting to forward frames.

Each LAN port on a software using Rapid PVST+ or MST exists in one of the following four states:

- Blocking—The LAN port does not participate in frame forwarding.
- Learning—The LAN port prepares to participate in frame forwarding.
- Forwarding—The LAN port forwards frames.
- Disabled—The LAN port does not participate in STP and is not forwarding frames.

When you enable Rapid PVST+, every port in the software, VLAN, and network goes through the blocking state and the transitory states of learning at power up. If properly configured, each LAN port stabilizes to the forwarding or blocking state.

When the STP algorithm places a LAN port in the forwarding state, the following process occurs:

- The LAN port is put into the blocking state while it waits for protocol information that suggests it should go to the learning state.
- The LAN port waits for the forward delay timer to expire, moves the LAN port to the learning state, and restarts the forward delay timer.

- In the learning state, the LAN port continues to block frame forwarding as it learns the end station location information for the forwarding database.
- The LAN port waits for the forward delay timer to expire and then moves the LAN port to the forwarding state, where both learning and frame forwarding are enabled.

Blocking State

A LAN port in the blocking state does not participate in frame forwarding.

A LAN port in the blocking state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate the end station location into its address database. (There is no learning on a blocking LAN port, so there is no address database update.)
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to network management messages.

Learning State

A LAN port in the learning state prepares to participate in frame forwarding by learning the MAC addresses for the frames. The LAN port enters the learning state from the blocking state.

A LAN port in the learning state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Incorporates the end station location into its address database.
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to network management messages.

Forwarding State

A LAN port in the forwarding state forwards frames. The LAN port enters the forwarding state from the learning state.

A LAN port in the forwarding state performs as follows:

- Forwards frames received from the attached segment.
- Forwards frames switched from another port for forwarding.
- Incorporates the end station location information into its address database.
- Receives BPDUs and directs them to the system module.
- Processes BPDUs received from the system module.

- Receives and responds to network management messages.

Disabled State

A LAN port in the disabled state does not participate in frame forwarding or STP. A LAN port in the disabled state is virtually nonoperational.

A disabled LAN port performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate the end station location into its address database. (There is no learning, so there is no address database update.)
- Does not receive BPDUs from neighbors.
- Does not receive BPDUs for transmission from the system module.

Summary of Port States

The following table lists the possible operational and Rapid PVST+ states for ports and the corresponding inclusion in the active topology.

Table 7: Port State Active Topology

Operational Status	Port State	Is Port Included in the Active Topology?
Enabled	Blocking	No
Enabled	Learning	Yes
Enabled	Forwarding	Yes
Disabled	Disabled	No

Synchronization of Port Roles

When the switch receives a proposal message on one of its ports and that port is selected as the new root port, Rapid PVST+ forces all other ports to synchronize with the new root information.

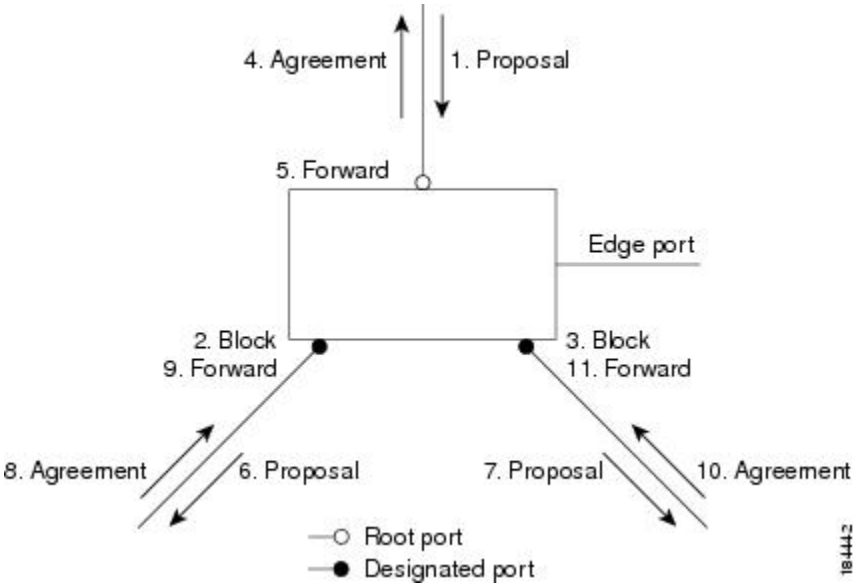
The switch is synchronized with superior root information received on the root port if all other ports are synchronized. An individual port on the switch is synchronized if either of the following applies:

- That port is in the blocking state.
- It is an edge port (a port configured to be at the edge of the network).

If a designated port is in the forwarding state and is not configured as an edge port, it transitions to the blocking state when the Rapid PVST+ forces it to synchronize with new root information. In general, when the Rapid PVST+ forces a port to synchronize with root information and the port does not satisfy any of the above conditions, its port state is set to blocking.

After ensuring that all of the ports are synchronized, the switch sends an agreement message to the designated switch that corresponds to its root port. When the switches connected by a point-to-point link are in agreement about their port roles, Rapid PVST+ immediately transitions the port states to the forwarding state. The sequence of events is shown in the following figure.

Figure 14: Sequence of Events During Rapid Convergence



Processing Superior BPDU Information

A superior BPDU is a BPDU with root information (such as a lower switch ID or lower path cost) that is superior to what is currently stored for the port.

If a port receives a superior BPDU, Rapid PVST+ triggers a reconfiguration. If the port is proposed and is selected as the new root port, Rapid PVST+ forces all the other ports to synchronize.

If the received BPDU is a Rapid PVST+ BPDU with the proposal flag set, the switch sends an agreement message after all of the other ports are synchronized. The new root port transitions to the forwarding state as soon as the previous port reaches the blocking state.

If the superior information received on the port causes the port to become a backup port or an alternate port, Rapid PVST+ sets the port to the blocking state and sends an agreement message. The designated port continues sending BPDUs with the proposal flag set until the forward-delay timer expires. At that time, the port transitions to the forwarding state.

Processing Inferior BPDU Information

An inferior BPDU is a BPDU with root information (such as a higher switch ID or higher path cost) that is inferior to what is currently stored for the port.

If a designated port receives an inferior BPDU, it immediately replies with its own information.

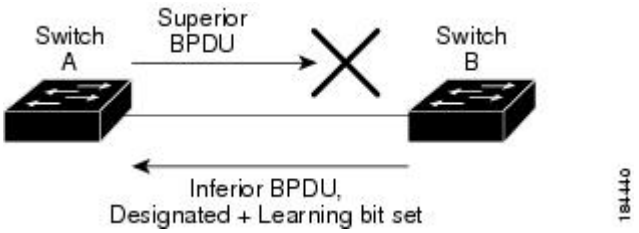
Spanning-Tree Dispute Mechanism

The software checks the consistency of the port role and state in the received BPDUs to detect unidirectional link failures that could cause bridging loops.

When a designated port detects a conflict, it keeps its role, but reverts to a discarding state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

The following figure illustrates a unidirectional link failure that typically creates a bridging loop. Switch A is the root bridge, and its BPDUs are lost on the link leading to Switch B. The 802.1w-standard BPDUs include the role and state of the sending port. With this information, Switch A can detect that Switch B does not react to the superior BPDUs it sends and that Switch B is the designated, not root port. As a result, Switch A blocks (or keeps blocking) its port, which prevents the bridging loop. The block is shown as an STP dispute.

Figure 15: Detecting Unidirectional Link Failure



Port Cost

Note Rapid PVST+ uses the short (16-bit) path-cost method to calculate the cost by default. With the short path-cost method, you can assign any value in the range of 1 to 65535. However, you can configure the switch to use the long (32-bit) path-cost method, which allows you to assign any value in the range of 1 to 200,000,000. You configure the path-cost calculation method globally.

The STP port path-cost default value is determined from the media speed and path-cost calculation method of a LAN interface. If a loop occurs, STP considers the port cost when selecting a LAN interface to put into the forwarding state.

Table 8: Default Port Cost

Bandwidth	Short Path-Cost Method of Port Cost	Long Path-Cost Method of Port Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gigabit Ethernet	4	20,000
10 Gigabit Ethernet	2	2,000

You can assign lower cost values to LAN interfaces that you want STP to select first and higher cost values to LAN interfaces that you want STP to select last. If all LAN interfaces have the same cost value, STP puts the LAN interface with the lowest LAN interface number in the forwarding state and blocks other LAN interfaces.

On access ports, you assign the port cost by the port. On trunk ports, you assign the port cost by the VLAN; you can configure the same port cost to all the VLANs on a trunk port.

Port Priority

If a loop occurs and multiple ports have the same path cost, Rapid PVST+ considers the port priority when selecting which LAN port to put into the forwarding state. You can assign lower priority values to LAN ports that you want Rapid PVST+ to select first and higher priority values to LAN ports that you want Rapid PVST+ to select last.

If all LAN ports have the same priority value, Rapid PVST+ puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports. The possible priority range is from 0 through 224 (the default is128), configurable in increments of 32. The software uses the port priority value when the LAN port is configured as an access port and uses the VLAN port priority values when the LAN port is configured as a trunk port.

Rapid PVST+ and IEEE 802.1Q Trunks

In a network of Cisco switches connected through 802.1Q trunks, the switches maintain one instance of STP for each VLAN allowed on the trunks. However, non-Cisco 802.1Q switches maintain only one instance of STP for all VLANs allowed on the trunks.

When you connect a Cisco switch to a non-Cisco switch through an 802.1Q trunk, the Cisco switch combines the STP instance of the 802.1Q VLAN of the trunk with the STP instance of the non-Cisco 802.1Q switch. However, all per-VLAN STP information that is maintained by Cisco switches is separated by a cloud of non-Cisco 802.1Q switches. The non-Cisco 802.1Q cloud that separates the Cisco switches is treated as a single trunk link between the switches.

Rapid PVST+ Interoperation with Legacy 802.1D STP

Rapid PVST+ can interoperate with switches that are running the legacy 802.1D protocol. The switch knows that it is interoperating with equipment running 802.1D when it receives a BPDU version 0. The BPDUs for Rapid PVST+ are version 2. If the BPDU received is an 802.1w BPDU version 2 with the proposal flag set, the switch sends an agreement message after all of the other ports are synchronized. If the BPDU is an 802.1D BPDU version 0, the switch does not set the proposal flag and starts the forward-delay timer for the port. The new root port requires twice the forward-delay time to transition to the forwarding state.

The switch interoperates with legacy 802.1D switches as follows:

- Notification—Unlike 802.1D BPDUs, 802.1w does not use TCN BPDUs. However, for interoperability with 802.1D switches, Cisco NX-OS processes and generates TCN BPDUs.
- Acknowledgement—When an 802.1w switch receives a TCN message on a designated port from an 802.1D switch, it replies with an 802.1D configuration BPDU with the TCA bit set. However, if the TC-while timer (the same as the TC timer in 802.1D) is active on a root port connected to an 802.1D switch and a configuration BPDU with the TCA set is received, the TC-while timer is reset.

This method of operation is required only for 802.1D switches. The 802.1w BPDUs do not have the TCA bit set.

- Protocol migration—For backward compatibility with 802.1D switches, 802.1w selectively sends 802.1D configuration BPDUs and TCN BPDUs on a per-port basis.

When a port is initialized, the migrate-delay timer is started (specifies the minimum time during which 802.1w BPDUs are sent), and 802.1w BPDUs are sent. While this timer is active, the switch processes all BPDUs received on that port and ignores the protocol type.

If the switch receives an 802.1D BPDU after the port migration-delay timer has expired, it assumes that it is connected to an 802.1D switch and starts using only 802.1D BPDUs. However, if the 802.1w switch is using 802.1D BPDUs on a port and receives an 802.1w BPDU after the timer has expired, it restarts the timer and starts using 802.1w BPDUs on that port.

Note If you want all switches to renegotiate the protocol, you must restart Rapid PVST+.

Rapid PVST+ Interoperation with 802.1s MST

Rapid PVST+ interoperates seamlessly with the IEEE 802.1s Multiple Spanning Tree (MST) standard. No user configuration is needed.

Configuring Rapid PVST+

Rapid PVST+, which has the 802.1w standard applied to the Rapid PVST+ protocol, is the default STP setting in the software.

You enable Rapid PVST+ on a per-VLAN basis. The software maintains a separate instance of STP for each VLAN (except on those VLANs on which you disable STP). By default, Rapid PVST+ is enabled on the default VLAN and on each VLAN that you create.

Enabling Rapid PVST+

Once you enable Rapid PVST+ on the switch, you must enable Rapid PVST+ on the specified VLANs.

Rapid PVST+ is the default STP mode. You cannot simultaneously run MST and Rapid PVST+.

Note Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mode rapid-pvst**	Enables Rapid PVST+ on the switch. Rapid PVST+ is the default spanning tree mode. **Note** Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

Example

This example shows how to enable Rapid PVST+ on the switch:

```
switch# configure terminal
switch(config)# spanning-tree mode rapid-pvst
```

Note Because STP is enabled by default, entering the **show running-config** command to view the resulting configuration does not display the command that you entered to enable Rapid PVST+.

Enabling Rapid PVST+ per VLAN

You can enable or disable Rapid PVST+ on each VLAN.

Note Rapid PVST+ is enabled by default on the default VLAN and on all VLANs that you create.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree** *vlan-range*	Enables Rapid PVST+ (default STP) on a per VLAN basis. The *vlan-range* value can be 2 through 4094 (except reserved VLAN values).
Step 3	(Optional) switch(config)# **no spanning-tree** *vlan-range*	Disables Rapid PVST+ on the specified VLAN.

	Command or Action	Purpose
		Caution Do not disable spanning tree on a VLAN unless all switches and bridges in the VLAN have spanning tree disabled. You cannot disable spanning tree on some of the switches and bridges in a VLAN and leave it enabled on other switches and bridges. This action can have unexpected results because switches and bridges with spanning tree enabled will have incomplete information regarding the physical topology of the network. Do not disable spanning tree in a VLAN without ensuring that there are no physical loops present in the VLAN. Spanning tree serves as a safeguard against misconfigurations and cabling errors.

Example

This example shows how to enable STP on a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5
```

Configuring the Root Bridge ID

The software maintains a separate instance of STP for each active VLAN in Rapid PVST+. For each VLAN, the switch with the lowest bridge ID becomes the root bridge for that VLAN.

To configure a VLAN instance to become the root bridge, modify the bridge priority from the default value (32768) to a significantly lower value.

When you enter the **spanning-tree vlan** *vlan_ID* **root** command, the switch checks the bridge priority of the current root bridges for each VLAN. The switch sets the bridge priority for the specified VLANs to 24576 if this value will cause the switch to become the root for the specified VLANs. If any root bridge for the specified VLANs has a bridge priority lower than 24576, the switch sets the bridge priority for the specified VLANs to 4096 less than the lowest bridge priority.

Note The **spanning-tree vlan** *vlan_ID* **root** command fails if the value required to be the root bridge is less than 1.

Caution The root bridge for each instance of STP should be a backbone or distribution switch. Do not configure an access switch as the STP primary root.

Enter the **diameter** keyword to specify the network diameter (that is, the maximum number of bridge hops between any two end stations in the network). When you specify the network diameter, the software automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can enter the **hello-time** keyword to override the automatically calculated hello time.

Note With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the **spanning-tree mst hello-time**, **spanning-tree mst forward-time**, and **spanning-tree mst max-age** configuration commands.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **root primary** [**diameter** *dia* [**hello-time** *hello-time*]]	Configures a software switch as the primary root bridge. The *vlan-range* value can be 2 through 4094 (except reserved VLAN values.) The *dia* default is 7. The *hello-time* can be from 1 to 10 seconds, and the default value is 2 seconds.

Example

This example shows how to configure the switch as the root bridge for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 root primary diameter 4
```

Configuring a Secondary Root Bridge

When you configure a software switch as the secondary root, the STP bridge priority is modified from the default value (32768) so that the switch is likely to become the root bridge for the specified VLANs if the primary root bridge fails (assuming the other switches in the network use the default bridge priority of 32768). STP sets the bridge priority to 28672.

Enter the **diameter** keyword to specify the network diameter (that is, the maximum number of bridge hops between any two end stations in the network). When you specify the network diameter, the software automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can enter the **hello-time** keyword to override the automatically calculated hello time.

You configure more than one switch in this manner to have multiple backup root bridges. Enter the same network diameter and hello time values that you used when configuring the primary root bridge.

Note With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the **spanning-tree mst hello-time**, **spanning-tree mst forward-time**, and **spanning-tree mst max-age** global configuration commands.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **root secondary** [**diameter** *dia* [**hello-time** *hello-time*]]	Configures a software switch as the secondary root bridge. The *vlan-range* value can be 2 through 4094 (except reserved VLAN values). The *dia* default is 7. The *hello-time* can be from 1 to 10 seconds, and the default value is 2 seconds.

Example

This example shows how to configure the switch as the secondary root bridge for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 root secondary diameter 4
```

Configuring the Rapid PVST+ Port Priority

You can assign lower priority values to LAN ports that you want Rapid PVST+ to select first and higher priority values to LAN ports that you want Rapid PVST+ to select last. If all LAN ports have the same priority value, Rapid PVST+ puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports.

The software uses the port priority value when the LAN port is configured as an access port and uses VLAN port priority values when the LAN port is configured as a trunk port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **spanning-tree** [**vlan** *vlan-list*] **port-priority** *priority*	Configures the port priority for the LAN interface. The *priority* value can be from 0 to

	Command or Action	Purpose
		224. The lower the value indicates the higher the priority. The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. All other values are rejected. The default value is 128.

Example

This example shows how to configure the access port priority of an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port-priority 160
```

You can only apply this command to a physical Ethernet interface.

Configuring the Rapid PVST+ Path-Cost Method and Port Cost

On access ports, you assign port cost by the port. On trunk ports, you assign the port cost by VLAN; you can configure the same port cost on all the VLANs on a trunk.

Note In Rapid PVST+ mode, you can use either the short or long path-cost method, and you can configure the method in either the interface or configuration submode.The default path-cost method is short.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree pathcost method** {**long** | **short**}	Selects the method used for Rapid PVST+ path-cost calculations. The default method is the short method.
Step 3	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters interface configuration mode.
Step 4	switch(config-if)# **spanning-tree** [**vlan** *vlan-id*] **cost** [*value* | **auto**]	Configures the port cost for the LAN interface. The cost value, depending on the path-cost calculation method, can be as follows: • short—1 to 65535 • long—1 to 200000000 **Note** You configure this parameter per interface on access ports and per VLAN on trunk ports.

	Command or Action	Purpose
		The default is **auto** , which sets the port cost on both the path-cost calculation method and the media speed.

Example

This example shows how to configure the access port cost of an Ethernet interface:

```
switch# configure terminal
switch (config)# spanning-tree pathcost method long
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree cost 1000
```

You can only apply this command to a physical Ethernet interface.

Configuring the Rapid PVST+ Bridge Priority of a VLAN

You can configure the Rapid PVST+ bridge priority of a VLAN.

Note Be careful when using this configuration. For most situations, we recommend that you configure the primary root and secondary root to modify the bridge priority.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **priority** *value*	Configures the bridge priority of a VLAN. Valid values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. All other values are rejected. The default value is 32768.

Example

This example shows how to configure the bridge priority of a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 priority 8192
```

Configuring the Rapid PVST+ Hello Time for a VLAN

You can configure the Rapid PVST+ hello time for a VLAN.

Note Be careful when using this configuration. For most situations, we recommend that you configure the primary root and secondary root to modify the hello time.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **hello-time** *hello-time*	Configures the hello time of a VLAN. The hello time value can be from 1 to 10 seconds. The default is 2 seconds.

Example

This example shows how to configure the hello time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 hello-time 7
```

Configuring the Rapid PVST+ Forward Delay Time for a VLAN

You can configure the forward delay time per VLAN when using Rapid PVST+.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **forward-time** *forward-time*	Configures the forward delay time of a VLAN. The forward delay time value can be from 4 to 30 seconds, and the default is 15 seconds.

Example

This example shows how to configure the forward delay time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 forward-time 21
```

Configuring the Rapid PVST+ Maximum Age Time for a VLAN

You can configure the maximum age time per VLAN when using Rapid PVST+.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **max-age** *max-age*	Configures the maximum aging time of a VLAN. The maximum aging time value can be from 6 to 40 seconds, and the default is 20 seconds.

Example

This example shows how to configure the maximum aging time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 max-age 36
```

Specifying the Link Type

Rapid connectivity (802.1w standard) is established only on point-to-point links. By default, the link type is controlled from the duplex mode of the interface. A full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection.

If you have a half-duplex link physically connected point-to-point to a single port on a remote switch, you can override the default setting on the link type and enable rapid transitions.

If you set the link to shared, STP moves back to 802.1D.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **spanning-tree link-type** {**auto** | **point-to-point** | **shared**}	Configures the link type to be either a point-to-point link or shared link. The system reads the default value from the switch connection, as follows: half duplex links are shared and full-duplex links are point-to-point. If the link type is shared, the STP reverts to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface.

Example

This example shows how to configure the link type as a point-to-point link:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
```

You can only apply this command to a physical Ethernet interface.

Restarting the Protocol

A bridge running Rapid PVST+ can send 802.1D BPDUs on one of its ports when it is connected to a legacy bridge. However, the STP protocol migration cannot determine whether the legacy switch has been removed from the link unless the legacy switch is the designated switch. You can restart the protocol negotiation (force the renegotiation with neighboring switches) on the entire switch or on specified interfaces.

Command	Purpose
switch# **clear spanning-tree detected-protocol** [**interface** *interface* [*interface-num* | *port-channel*]]	Restarts Rapid PVST+ on all interfaces on the switch or specified interfaces.

This example shows how to restart Rapid PVST+ on an Ethernet interface:

```
switch# clear spanning-tree detected-protocol interface ethernet 1/8
```

Verifying the Rapid PVST+ Configuration

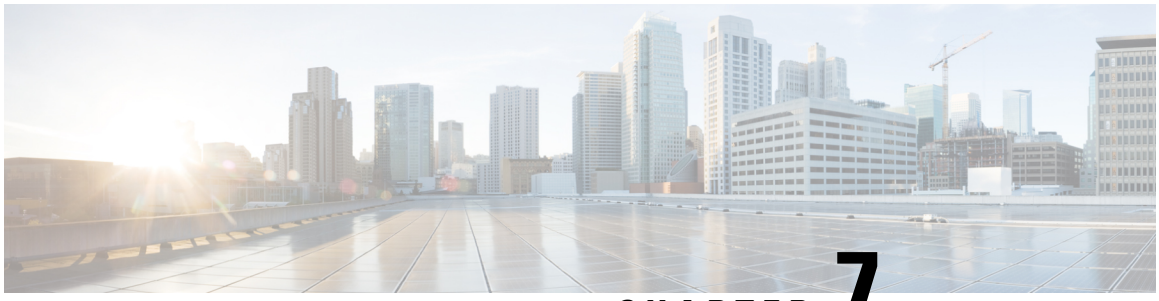
Use the following commands to display Rapid PVST+ configuration information.

Command	Purpose
show running-config spanning-tree [**all**]	Displays the current spanning tree configuration.
show spanning-tree [*options*]	Displays selected detailed information for the current spanning tree configuration.

This example shows how to display spanning tree status:

```
switch# show spanning-tree brief

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32768
             Address     001c.b05a.5447
             Cost        2
             Port        131 (Ethernet1/3)
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     000d.ec6d.7841
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
```

```
Eth1/3           Root FWD 2         128.131  P2p Peer(STP)
```



CHAPTER 7

Configuring Multiple Spanning Tree

This chapter contains the following sections:

Information About MST

MST Overview

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

MST maps multiple VLANs into a spanning tree instance with each instance having a spanning tree topology independent of other spanning tree instances. This architecture provides multiple forwarding paths for data traffic, enables load balancing, and reduces the number of STP instances required to support a large number of VLANs.

MST provides rapid convergence through explicit handshaking as each MST instance uses the IEEE 802.1w standard, which eliminates the 802.1D forwarding delay and quickly transitions root bridge ports and designated ports to the forwarding state.

MAC address reduction is always enabled while you are using MST. You cannot disable this feature.

MST improves spanning tree operation and maintains backward compatibility with these STP versions:

- Original 802.1D spanning tree
- Rapid per-VLAN spanning tree (Rapid PVST+)

  IEEE 802.1w defined the Rapid Spanning Tree Protocol (RSTP) and was incorporated into IEEE 802.1D.
- IEEE 802.1s defined MST and was incorporated into IEEE 802.1Q.

Note You must enable MST; Rapid PVST+ is the default spanning tree mode.

MST Regions

To allow switches to participate in MST instances, you must consistently configure the switches with the same MST configuration information.

A collection of interconnected switches that have the same MST configuration is an MST region. An MST region is a linked group of MST bridges with the same MST configuration.

The MST configuration controls the MST region to which each switch belongs. The configuration includes the name of the region, the revision number, and the MST VLAN-to-instance assignment map.

A region can have one or multiple members with the same MST configuration. Each member must be capable of processing 802.1w bridge protocol data units (BPDUs). There is no limit to the number of MST regions in a network.

Each region can support up to 65 MST instances (MSTIs). Instances are identified by any number in the range from 1 to 4094. The system reserves Instance 0 for a special instance, which is the IST. You can assign a VLAN to only one MST instance at a time.

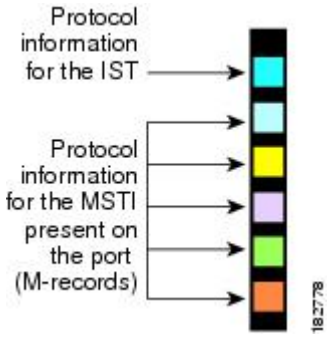
The MST region appears as a single bridge to adjacent MST regions and to other Rapid PVST+ regions and 802.1D spanning tree protocols.

Note We recommend that you do not partition the network into a large number of regions.

MST BPDUs

Each region has only one MST BPDU, and that BPDU carries an M-record for each MSTI within the region (see the following figure). Only the IST sends BPDUs for the MST region; all M-records are encapsulated in that one BPDU that the IST sends. Because the MST BPDU carries information for all instances, the number of BPDUs that need to be processed to support MSTIs is significantly reduced.

Figure 16: MST BPDU with M-Records for MSTIs



MST Configuration Information

The MST configuration that must be identical on all switches within a single MST region is configured by the user.

You can configure the following three parameters of the MST configuration:

- Name—32-character string, null padded and null terminated, identifying the MST region
- Revision number—Unsigned 16-bit number that identifies the revision of the current MST configuration

Note You must set the revision number when required as part of the MST configuration. The revision number is *not* incremented automatically each time that the MST configuration is committed.

- MST configuration table—4096-element table that associates each of the potential 4094 VLANs supported to a given instance with the first (0) and last element (4095) set to 0. The value of element number X represents the instance to which VLAN X is mapped.

Caution When you change the VLAN-to-MSTI mapping, the system restarts MST.

MST BPDUs contain these three configuration parameters. An MST bridge accepts an MST BPDU into its own region only if these three configuration parameters match exactly. If one configuration attribute differs, the MST bridge considers the BPDU to be from another MST region.

IST, CIST, and CST

IST, CIST, and CST Overview

Unlike Rapid PVST+, in which all the STP instances are independent, MST establishes and maintains IST, CIST, and CST spanning trees, as follows:

- An IST is the spanning tree that runs in an MST region.

MST establishes and maintains additional spanning trees within each MST region; these spanning trees are called multiple spanning tree instances (MSTIs).

Instance 0 is a special instance for a region, known as the IST. The IST always exists on all ports; you cannot delete the IST, or instance 0. By default, all VLANs are assigned to the IST. All other MST instances are numbered from 1 to 4094.

The IST is the only STP instance that sends and receives BPDUs. All of the other MSTI information is contained in MST records (M-records), which are encapsulated within MST BPDUs.

All MSTIs within the same region share the same protocol timers, but each MSTI has its own topology parameters, such as the root bridge ID, the root path cost, and so forth.

An MSTI is local to the region; for example, MSTI 9 in region A is independent of MSTI 9 in region B, even if regions A and B are interconnected.

- The CST interconnects the MST regions and any instance of 802.1D and 802.1w STP that may be running on the network. The CST is the one STP instance for the entire bridged network and encompasses all MST regions and 802.1w and 802.1D instances.
- A CIST is a collection of the ISTs in each MST region. The CIST is the same as an IST inside an MST region, and the same as a CST outside an MST region.

The spanning tree computed in an MST region appears as a subtree in the CST that encompasses the entire switched domain. The CIST is formed by the spanning tree algorithm running among switches that support the 802.1w, 802.1s, and 802.1D standards. The CIST inside an MST region is the same as the CST outside a region.

Spanning Tree Operation Within an MST Region

The IST connects all the MST switches in a region. When the IST converges, the root of the IST becomes the CIST regional root. The CIST regional root is also the CIST root if there is only one region in the network. If the CIST root is outside the region, the protocol selects one of the MST switches at the boundary of the region as the CIST regional root.

When an MST switch initializes, it sends BPDUs that identify itself as the root of the CIST and the CIST regional root, with both the path costs to the CIST root and to the CIST regional root set to zero. The switch also initializes all of its MSTIs and claims to be the root for all of them. If the switch receives superior MST root information (lower switch ID, lower path cost, and so forth) than the information that is currently stored for the port, it relinquishes its claim as the CIST regional root.

During initialization, an MST region might have many subregions, each with its own CIST regional root. As switches receive superior IST information from a neighbor in the same region, they leave their old subregions and join the new subregion that contains the true CIST regional root. This action causes all subregions to shrink except for the subregion that contains the true CIST regional root.

All switches in the MST region must agree on the same CIST regional root. Any two switches in the region will only synchronize their port roles for an MSTI if they converge to a common CIST regional root.

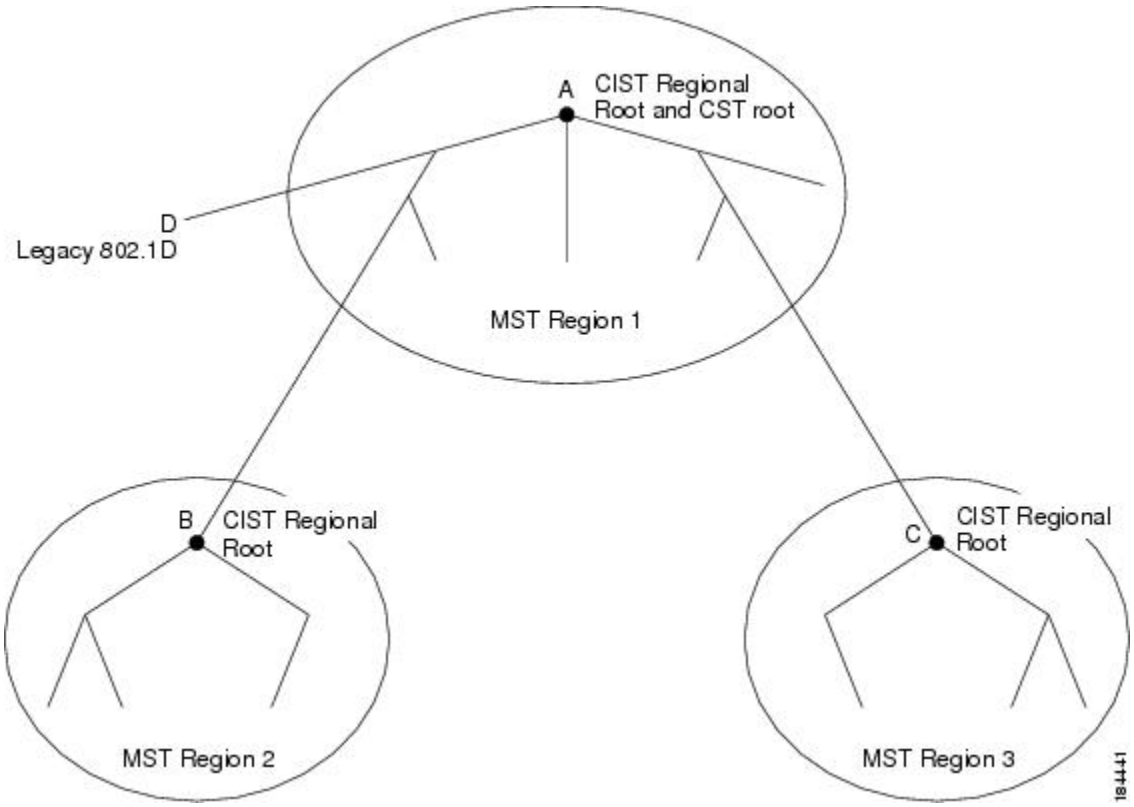
Spanning Tree Operations Between MST Regions

If you have multiple regions or 802.1 w or 802.1D STP instances within a network, MST establishes and maintains the CST, which includes all MST regions and all 802.1w and 802.1D STP switches in the network. The MSTIs combine with the IST at the boundary of the region to become the CST.

The IST connects all the MST switches in the region and appears as a subtree in the CIST that encompasses the entire switched domain. The root of the subtree is the CIST regional root. The MST region appears as a virtual switch to adjacent STP switches and MST regions.

The following figure shows a network with three MST regions and an 802.1D switch (D). The CIST regional root for region 1 (A) is also the CIST root. The CIST regional root for region 2 (B) and the CIST regional root for region 3 (C) are the roots for their respective subtrees within the CIST.

Figure 17: MST Regions, CIST Regional Roots, and CST Root



Only the CST instance sends and receives BPDUs. MSTIs add their spanning tree information into the BPDUs (as M-records) to interact with neighboring switches and compute the final spanning tree topology. Because of this process, the spanning tree parameters related to the BPDU transmission (for example, hello time, forward time, max-age, and max-hops) are configured only on the CST instance but affect all MSTIs. You can configure the parameters related to the spanning tree topology (for example, the switch priority, the port VLAN cost, and the port VLAN priority) on both the CST instance and the MSTI.

MST switches use Version 3 BPDUs or 802.1D STP BPDUs to communicate with 802.1D-only switches. MST switches use MST BPDUs to communicate with MST switches.

MST Terminology

MST naming conventions include identification of some internal or regional parameters. These parameters are used only within an MST region, compared to external parameters that are used throughout the whole network. Because the CIST is the only spanning tree instance that spans the whole network, only the CIST parameters require the external qualifiers and not the internal or regional qualifiers. The MST terminology is as follows:

- The CIST root is the root bridge for the CIST, which is the unique instance that spans the whole network.
- The CIST external root path cost is the cost to the CIST root. This cost is left unchanged within an MST region. An MST region looks like a single switch to the CIST. The CIST external root path cost is the root path cost calculated between these virtual switches and switches that do not belong to any region.

- If the CIST root is in the region, the CIST regional root is the CIST root. Otherwise, the CIST regional root is the closest switch to the CIST root in the region. The CIST regional root acts as a root bridge for the IST.
- The CIST internal root path cost is the cost to the CIST regional root in a region. This cost is only relevant to the IST, instance 0.

Hop Count

MST does not use the message-age and maximum-age information in the configuration BPDU to compute the STP topology inside the MST region. Instead, the protocol uses the path cost to the root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism.

By using the **spanning-tree mst max-hops** global configuration command, you can configure the maximum hops inside the region and apply it to the IST and all MST instances in that region.

The hop count achieves the same result as the message-age information (triggers a reconfiguration). The root bridge of the instance always sends a BPDU (or M-record) with a cost of 0 and the hop count set to the maximum value. When a switch receives this BPDU, it decrements the received remaining hop count by one and propagates this value as the remaining hop count in the BPDUs that it generates. When the count reaches zero, the switch discards the BPDU and ages the information held for the port.

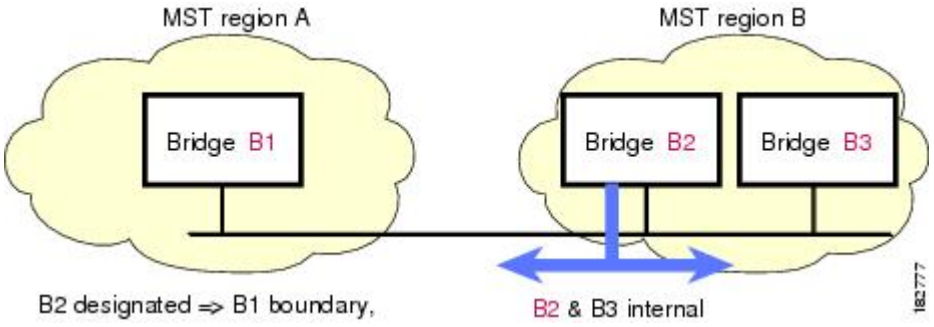
The message-age and maximum-age information in the 802.1w portion of the BPDU remain the same throughout the region (only on the IST), and the same values are propagated by the region-designated ports at the boundary.

You configure a maximum aging time as the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration.

Boundary Ports

A boundary port is a port that connects one region to another. A designated port knows that it is on the boundary if it detects an STP bridge or receives an agreement proposal from an MST bridge with a different configuration or a Rapid PVST+ bridge. This definition allows two ports that are internal to a region to share a segment with a port that belongs to a different region, creating the possibility of receiving both internal and external messages on a port (see the following figure).

Figure 18: MST Boundary Ports



At the boundary, the roles of MST ports do not matter; the system forces their state to be the same as the IST port state. If the boundary flag is set for the port, the MST port-role selection process assigns a port role to the boundary and assigns the same state as the state of the IST port. The IST port at the boundary can take up any port role except a backup port role.

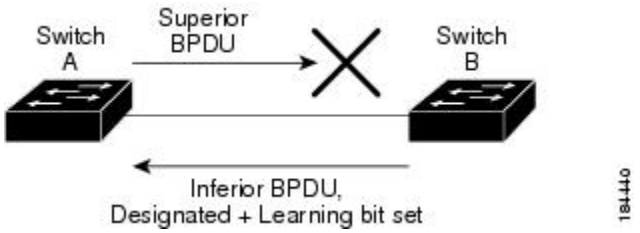
Spanning-Tree Dispute Mechanism

Currently, this feature is not present in the IEEE MST standard, but it is included in the standard-compliant implementation. The software checks the consistency of the port role and state in the received BPDUs to detect unidirectional link failures that could cause bridging loops.

When a designated port detects a conflict, it keeps its role but reverts to a discarding state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

The following figure shows a unidirectional link failure that typically creates a bridging loop. Switch A is the root bridge, and its BPDUs are lost on the link leading to Switch B. Rapid PVST+ (802.1w) and MST BPDUs include the role and state of the sending port. With this information, Switch A can detect that Switch B does not react to the superior BPDUs that it sends and that Switch B is the designated, not root port. As a result, Switch A blocks (or keeps blocking) its port, which prevents the bridging loop. The block is shown as an STP dispute.

Figure 19: Detecting a Unidirectional Link Failure



Port Cost and Port Priority

Spanning tree uses port costs to break a tie for the designated port. Lower values indicate lower port costs, and spanning tree chooses the least costly path. Default port costs are taken from the bandwidth of the interface, as follows:

- 10 Mbps—2,000,000
- 100 Mbps—200,000
- 1 Gigabit Ethernet—20,000
- 10 Gigabit Ethernet—2,000

You can configure the port costs in order to influence which port is chosen.

Note MST always uses the long path-cost calculation method, so the range of valid values is between 1 and 200,000,000.

The system uses port priorities to break ties among ports with the same cost. A lower number indicates a higher priority. The default port priority is 128. You can configure the priority to values between 0 and 224, in increments of 32.

Interoperability with IEEE 802.1D

A switch that runs MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D STP switches. If this switch receives an 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. In addition, an MST switch can detect that a port is at the boundary of a region when it receives an 802.1D BPDU, an MST BPDU (Version 3) associated with a different region, or an 802.1w BPDU (Version 2).

However, the switch does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D switch has been removed from the link unless the 802.1D switch is the designated switch. A switch might also continue to assign a boundary role to a port when the switch to which this switch is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches), enter the **clear spanning-tree detected-protocols** command.

All Rapid PVST+ switches (and all 8021.D STP switches) on the link can process MST BPDUs as if they are 802.1w BPDUs. MST switches can send either Version 0 configuration and topology change notification (TCN) BPDUs or Version 3 MST BPDUs on a boundary port. A boundary port connects to a LAN, the designated switch of which is either a single spanning tree switch or a switch with a different MST configuration.

Note MST interoperates with the Cisco prestandard Multiple Spanning Tree Protocol (MSTP) whenever it receives prestandard MSTP on an MST port; no explicit configuration is necessary.

Interoperability with Rapid PVST+: Understanding PVST Simulation

MST interoperates with Rapid PVST+ with no need for user configuration. The PVST simulation feature enables this seamless interoperability.

Note PVST simulation is enabled by default. That is, by default, all interfaces on the switch interoperate between MST and Rapid PVST+.

However, you may want to control the connection between MST and Rapid PVST+ to protect against accidentally connecting an MST-enabled port to a Rapid PVST+-enabled port. Because Rapid PVST+ is the default STP mode, you may encounter many Rapid PVST+-enabled connections.

Disabling Rapid PVST+ simulation, which can be done per port or globally for the entire switch, moves the MST-enabled port to the blocking state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving Rapid PVST+/SSTP BPDUs, and then the port resumes the normal STP transition process.

Configuring MST

MST Configuration Guidelines

When configuring MST, follow these guidelines:

- When you work with private VLANs, enter the **private-vlan synchronize** command to map the secondary VLANs to the same MST instance as the primary VLAN.
- When you are in the MST configuration mode, the following guidelines apply:
  - Each command reference line creates its pending regional configuration.
  - The pending region configuration starts with the current region configuration.
  - To leave the MST configuration mode without committing any changes, enter the **abort** command.
  - To leave the MST configuration mode and commit all the changes that you made before you left the mode, enter the **exit** command.

Enabling MST

You must enable MST; Rapid PVST+ is the default.

Caution Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode. Also, having two different spanning-tree modes on Virtual Port Channel (vPC) peer switches is an inconsistency, so this operation is disruptive.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch# **configure terminal**	Enters configuration mode.
Step 3	switch(config)# **spanning-tree mode mst**	Enables MST on the switch.
Step 4	(Optional) switch(config)# **no spanning-tree mode mst**	Disables MST on the switch and returns you to Rapid PVST+.

Example

This example shows how to enable MST on the switch:

```
switch# configure terminal
switch(config)# spanning-tree mode mst
```

Note Because STP is enabled by default, entering a **show running-config** command to view the resulting configuration does not display the command that you entered to enable STP.

Entering MST Configuration Mode

You enter MST configuration mode to configure the MST name, VLAN-to-instance mapping, and MST revision number on the switch.

For two or more switches to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

Note Each command reference line creates its pending regional configuration in MST configuration mode. In addition, the pending region configuration starts with the current region configuration.

When you are working in MST configuration mode, note the difference between the **exit** and **abort** commands.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration mode on the system. You must be in the MST configuration mode to assign the MST configuration parameters, as follows: • MST name • Instance-to-VLAN mapping • MST revision number • Synchronize primary and secondary VLANs in private VLANs
Step 3	switch(config-mst)# **exit** or switch(config-mst)# **abort**	Exits or aborts. • The **exit** command commits all the changes and exits MST configuration mode. • The **abort** command exits the MST configuration mode without committing any of the changes.

	Command or Action	Purpose
Step 4	(Optional) switch(config)# **no spanning-tree mst configuration**	Returns the MST region configuration to the following default values: • The region name is an empty string. • No VLANs are mapped to any MST instance (all VLANs are mapped to the CIST instance). • The revision number is 0.

Specifying the MST Name

You configure a region name on the bridge. For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration submode.
Step 3	switch(config-mst)# **name** *name*	Specifies the name for MST region. The *name* string has a maximum length of 32 case-sensitive characters. The default is an empty string.

Example

This example shows how to set the name of the MST region:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# name accounting
```

Specifying the MST Configuration Revision Number

You configure the revision number on the bridge. For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration submode.
Step 3	switch(config-mst)# **revision** *version*	Specifies the revision number for the MST region. The range is from 0 to 65535, and the default value is 0.

Example

This example shows how to configure the revision number of the MSTI region for 5:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# revision 5
```

Specifying the Configuration on an MST Region

For two or more switches to be in the same MST region, they must have the same VLAN-to-instance mapping, the same configuration revision number, and the same MST name.

A region can have one member or multiple members with the same MST configuration; each member must be capable of processing IEEE 802.1w RSTP BPDUs. There is no limit to the number of MST regions in a network, but each region can support only up to 65 MST instances. You can assign a VLAN to only one MST instance at a time.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration submode.
Step 3	switch(config-mst)# **instance** *instance-id* **vlan** *vlan-range*	Maps VLANs to an MST instance as follows: • For *instance-id* , the range is from 1 to 4094. • For **vlan** *vlan-range* , the range is from 1 to 4094. When you map VLANs to an MST instance, the mapping is incremental, and the VLANs specified in the command are added to or

	Command or Action	Purpose
		removed from the VLANs that were previously mapped. To specify a VLAN range, enter a hyphen; for example, enter the **instance 1 vlan 1-63** command to map VLANs 1 through 63 to MST instance 1. To specify a VLAN series, enter a comma; for example, enter the **instance 1 vlan 10, 20, 30** command to map VLANs 10, 20, and 30 to MST instance 1.
Step 4	switch(config-mst)# **name** *name*	Specifies the instance name. The *name* string has a maximum length of 32 case-sensitive characters.
Step 5	switch(config-mst)# **revision** *version*	Specifies the configuration revision number. The range is from 0 to 65535.

Example

To return to defaults, do the following:

- To return to the default MST region configuration settings, enter the **no spanning-tree mst configuration** configuration command.
- To return to the default VLAN-to-instance map, enter the **no instance** *instance-id* **vlan** *vlan-range* MST configuration command.
- To return to the default name, enter the **no name** MST configuration command.
- To return to the default revision number, enter the **no revision** MST configuration command.
- To reenable Rapid PVST+, enter the **no spanning-tree mode** or the **spanning-tree mode rapid-pvst** global configuration command.

This example shows how to enter MST configuration mode, map VLANs 10 to 20 to MST instance 1, name the region region1, set the configuration revision to 1, display the pending configuration, apply the changes, and return to global configuration mode:

```
switch(config)# spanning-tree mst configuration
switch(config-mst)# instance 1 vlan 10-20
switch(config-mst)# name region1
switch(config-mst)# revision 1
switch(config-mst)# show pending
Pending MST configuration
Name      [region1]
Revision  1
Instances configured 2
```

```
Instance  Vlans Mapped
--------  ---------------------
0         1-9,21-4094
1         10-20
------------------------------
```

Mapping and Unmapping VLANs to MST Instances

Caution When you change the VLAN-to-MSTI mapping, the system restarts MST.

Note You cannot disable an MSTI.

For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration submode.
Step 3	switch(config-mst)# **instance** *instance-id* **vlan** *vlan-range*	Maps VLANs to an MST instance, as follows: • For *instance-id* the range is from 1 to 4094. Instance 0 is reserved for the IST for each MST region. • For *vlan-range* the range is from 1 to 4094. When you map VLANs to an MSTI, the mapping is incremental, and the VLANs specified in the command are added to or removed from the VLANs that were previously mapped.
Step 4	switch(config-mst)# **no instance** *instance-id* **vlan** *vlan-range*	Deletes the specified instance and returns the VLANs to the default MSTI, which is the CIST.

Example

This example shows how to map VLAN 200 to MSTI 3:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# instance 3 vlan 200
```

Mapping Secondary VLANs to Same MSTI as Primary VLANs for Private VLANs

When you are working with private VLANs on the system, all secondary VLANs must be in the same MSTI and their associated primary VLAN.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration submode.
Step 3	switch(config-mst)# **private-vlan synchronize**	Automatically maps all secondary VLANs to the same MSTI as their associated primary VLAN in all private VLANs.

Example

This example shows how to automatically map all the secondary VLANs to the same MSTI as their associated primary VLANs in all private VLANs:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# private-vlan synchronize
```

Configuring the Root Bridge

You can configure the switch to become the root bridge.

Note The root bridge for each MSTI should be a backbone or distribution switch. Do not configure an access switch as the spanning tree primary root bridge.

Enter the **diameter** keyword, which is available only for MSTI 0 (or the IST), to specify the network diameter (that is, the maximum number of hops between any two end stations in the network). When you specify the network diameter, the switch automatically sets an optimal hello time, forward-delay time, and maximum-age

time for a network of that diameter, which can significantly reduce the convergence time. You can enter the **hello** keyword to override the automatically calculated hello time.

Note With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the **spanning-tree mst hello-time**, **spanning-tree mst forward-time**, and **spanning-tree mst max-age** global configuration commands.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]]	Configures a switch as the root bridge as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094. • For diameter *net-diameter*, specify the maximum number of hops between any two end stations. The default is 7. This keyword is available only for MST instance 0. • For hello-time *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds.
Step 3	(Optional) switch(config)# **no spanning-tree mst** *instance-id* **root**	Returns the switch priority, diameter, and hello time to default values.

Example

This example shows how to configure the switch as the root switch for MSTI 5:

```
switch# configure terminal
switch(config)# spanning-tree mst 5 root primary
```

Configuring a Secondary Root Bridge

You can execute this command on more than one switch to configure multiple backup root bridges. Enter the same network diameter and hello-time values that you used when you configured the primary root bridge with the **spanning-tree mst root primary** configuration command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]]	Configures a switch as the secondary root bridge as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094. • For diameter *net-diameter*, specify the maximum number of hops between any two end stations. The default is 7. This keyword is available only for MST instance 0. • For hello-time *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds.
Step 3	(Optional) switch(config)# **no spanning-tree mst** *instance-id* **root**	Returns the switch priority, diameter, and hello-time to default values.

Example

This example shows how to configure the switch as the secondary root switch for MSTI 5:

```
switch# configure terminal
switch(config)# spanning-tree mst 5 root secondary
```

Configuring the Port Priority

If a loop occurs, MST uses the port priority when selecting an interface to put into the forwarding state. You can assign lower priority values to interfaces that you want selected first and higher priority values to the interface that you want selected last. If all interfaces have the same priority value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.

	Command or Action	Purpose
Step 3	switch(config-if)# **spanning-tree mst** *instance-id* **port-priority** *priority*	Configures the port priority as follows: • For *instance-id*, you can specify a single MSTI, a range of MSTIs separated by a hyphen, or a series of MSTIs separated by a comma. The range is from 1 to 4094. • For *priority*, the range is 0 to 224 in increments of 32. The default is 128. A lower number indicates a higher priority. The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. The system rejects all other values.

Example

This example shows how to set the MST interface port priority for MSTI 3 on Ethernet port 3/1 to 64:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# spanning-tree mst 3 port-priority 64
```

You can only apply this command to a physical Ethernet interface.

Configuring the Port Cost

The MST path-cost default value is derived from the media speed of an interface. If a loop occurs, MST uses the cost when selecting an interface to put in the forwarding state. You can assign lower cost values to interfaces that you want selected first and higher cost to interfaces values that you want selected last. If all interfaces have the same cost value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Note MST uses the long path-cost calculation method.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.

	Command or Action	Purpose
Step 3	switch(config-if)# **spanning-tree mst** *instance-id* **cost** [*cost* | **auto**]	Configures the cost. If a loop occurs, MST uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094. • For *cost*, the range is from 1 to 200000000. The default value is auto, which is derived from the media speed of the interface.

Example

This example shows how to set the MST interface port cost on Ethernet 3/1 for MSTI 4:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# spanning-tree mst 4 cost 17031970
```

Configuring the Switch Priority

You can configure the switch priority for an MST instance so that it is more likely that the specified switch is chosen as the root bridge.

Note Exercise care when using this command. For most situations, we recommend that you enter the **spanning-tree mst root primary** and the **spanning-tree mst root secondary** global configuration commands to modify the switch priority.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst** *instance-id* **priority** *priority-value*	Configures a switch priority as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094.

	Command or Action	Purpose
		• For priority, the range is from 0 to 61440 in increments of 4096; the default is 32768. A lower number indicates that the switch will most likely be chosen as the root bridge. Priority values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. The system rejects all other values.

Example

This example shows how to configure the priority of the bridge to 4096 for MSTI 5:

```
switch# configure terminal
switch(config)# spanning-tree mst 5 priority 4096
```

Configuring the Hello Time

You can configure the interval between the generation of configuration messages by the root bridge for all instances on the switch by changing the hello time.

Note Exercise care when using this command. For most situations, we recommend that you enter the **spanning-tree mst** *instance-id* **root primary** and the **spanning-tree mst** *instance-id* **root secondary** configuration commands to modify the hello time.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst hello-time** *seconds*	Configures the hello time for all MST instances. The hello time is the interval between the generation of configuration messages by the root bridge. These messages mean that the switch is alive. For *seconds*, the range is from 1 to 10, and the default is 2 seconds.

Example

This example shows how to configure the hello time of the switch to 1 second:

```
switch# configure terminal
```

```
switch(config)# spanning-tree mst hello-time 1
```

Configuring the Forwarding-Delay Time

You can set the forward delay timer for all MST instances on the switch with one command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst forward-time** *seconds*	Configures the forward time for all MST instances. The forward delay is the number of seconds that a port waits before changing from its spanning tree blocking and learning states to the forwarding state. For *seconds*, the range is from 4 to 30, and the default is 15 seconds.

Example

This example shows how to configure the forward-delay time of the switch to 10 seconds:

```
switch# configure terminal
switch(config)# spanning-tree mst forward-time 10
```

Configuring the Maximum-Aging Time

The maximum-aging timer is the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration.

You set the maximum-aging timer for all MST instances on the switch with one command (the maximum age time only applies to the IST).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst max-age** *seconds*	Configures the maximum-aging time for all MST instances. The maximum-aging time is the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration. For *seconds*, the range is from 6 to 40, and the default is 20 seconds.

Example

This example shows how to configure the maximum-aging timer of the switch to 40 seconds:

```
switch# configure terminal
switch(config)# spanning-tree mst max-age 40
```

Configuring the Maximum-Hop Count

MST uses the path cost to the IST regional root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism. You configure the maximum hops inside the region and apply it to the IST and all MST instances in that region. The hop count achieves the same result as the message-age information (triggers a reconfiguration).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree mst max-hops** *hop-count*	Specifies the number of hops in a region before the BPDU is discarded, and the information held for a port is aged. For *hop-count*, the range is from 1 to 255, and the default value is 20 hops.

Example

This example shows how to set the maximum hops to 40:

```
switch# configure terminal
switch(config)# spanning-tree mst max-hops 40
```

Configuring PVST Simulation Globally

You can block this automatic feature either globally or per port. You can enter the global command and change the PVST simulation setting for the entire switch while you are in interface command mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no spanning-tree mst simulate pvst global**	Disables all interfaces on the switch from automatically interoperating with connected switch that is running in Rapid PVST+ mode.

	Command or Action	Purpose
		By default, all interfaces on the switch operate seamlessly between Rapid PVST+ and MST.

Example

This example shows how to prevent the switch from automatically interoperating with a connecting switch that is running Rapid PVST+:

```
switch# configure terminal
switch(config)# no spanning-tree mst simulate pvst global
```

Configuring PVST Simulation Per Port

MST interoperates seamlessly with Rapid PVST+. However, to prevent an accidental connection to a switch that does not run MST as the default STP mode, you may want to disable this automatic feature. If you disable PVST simulation, the MST-enabled port moves to the blocking state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving BPDUs, and then the port resumes the normal STP transition process.

You can block this automatic feature either globally or per port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **spanning-tree mst simulate pvst disable**	Disables specified interfaces from automatically interoperating with a connected switch that is running in Rapid PVST+ mode. By default, all interfaces on the switch operate seamlessly between Rapid PVST+ and MST.
Step 4	switch(config-if)# **spanning-tree mst simulate pvst**	Reenables the seamless operation between MST and Rapid PVST+ on specified interfaces.
Step 5	switch(config-if)# **no spanning-tree mst simulate pvst**	Sets the interface to the switch-wide MST and Rapid PVST+ interoperation that you configured using the **spanning-tree mst simulate pvst global** command.

Example

This example shows how to prevent the specified interfaces from automatically interoperating with a connecting switch that is not running MST:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree mst simulate pvst disable
```

Specifying the Link Type

Rapid connectivity (802.1w standard) is established only on point-to-point links. By default, the link type is controlled from the duplex mode of the interface. A full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection.

If you have a half-duplex link physically connected point-to-point to a single port on a remote switch, you can override the default setting on the link type and enable rapid transitions.

If you set the link to shared, STP reverts to 802.1D.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **spanning-tree link-type** {**auto** | **point-to-point** | **shared**}	Configures the link type to be either point to point or shared. The system reads the default value from the switch connection. Half-duplex links are shared and full-duplex links are point to point. If the link type is shared, the STP reverts to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface.

Example

This example shows how to configure the link type as point to point:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
```

Restarting the Protocol

An MST bridge can detect that a port is at the boundary of a region when it receives a legacy BPDU or an MST BPDU that is associated with a different region. However, the STP protocol migration cannot determine whether the legacy switch, which is a switch that runs only IEEE 802.1D, has been removed from the link unless the legacy switch is the designated switch. Enter this command to restart the protocol negotiation (force the renegotiation with neighboring switches) on the entire switch or on specified interfaces.

Procedure

	Command or Action	Purpose
Step 1	switch# **clear spanning-tree detected-protocol** [**interface** *interface* [*interface-num* | *port-channel*]]	Restarts MST on the entire switch or specified interfaces.

Example

This example shows how to restart MST on the Ethernet interface on slot 2, port 8:

```
switch# clear spanning-tree detected-protocol interface ethernet 2/8
```

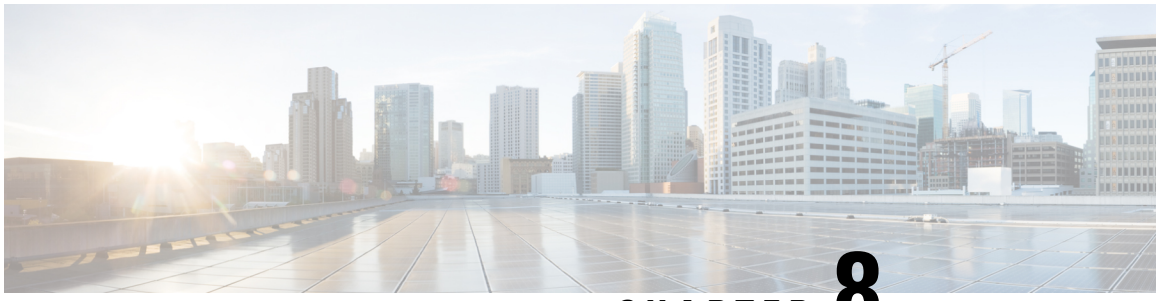
Verifying the MST Configuration

Use the following commands to display MST configuration information.

Command	Purpose
show running-config spanning-tree [**all**]	Displays the current spanning tree configuration.
show spanning-tree mst [*options*]	Displays detailed information for the current MST configuration.

This example shows how to display the current MST configuration:

```
switch# show spanning-tree mst configuration
% Switch is not in mst mode
Name      [mist-attempt]
Revision  1     Instances configured 2
Instance  Vlans mapped
--------  ---------------------------------------------------------------------
0         1-12,14-41,43-4094
1         13,42
```



CHAPTER 8

Configuring STP Extensions

This chapter contains the following sections:

Overview

Cisco has added extensions to Spanning Tree Protocol (STP) that make convergence more efficient. In some cases, even though similar functionality may be incorporated into the IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) standard, we recommend using these extensions. All of these extensions can be used with both RPVST+ and Multiple Spanning Tree Protocol (MST).

The available extensions are spanning tree port types, Bridge Assurance, BPDU Guard, BPDU Filtering, Loop Guard, and Root Guard. Many of these features can be applied either globally or on specified interfaces.

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

Information About STP Extensions

Understanding STP Port Types

You can configure a spanning tree port as an edge port, a network port, or a normal port. A port can be in only one of these states at a given time. The default spanning tree port type is normal. Depending on the type of device to which the interface is connected, you can configure a spanning tree port as one of these port types.

Spanning Tree Edge Ports

Edge ports, which are connected to hosts, can be either an access port or a trunk port. The edge port interface immediately transitions to the forwarding state, without moving through the blocking or learning states. (This immediate transition was previously configured as the Cisco-proprietary feature PortFast.)

Interfaces that are connected to hosts should not receive STP bridge protocol data units (BPDUs).

Note If you configure a port connected to another switch as an edge port, you might create a bridging loop.

Spanning Tree Network Ports

Network ports are connected only to switches or bridges. Configuring a port as a network port while Bridge Assurance is enabled globally, enables Bridge Assurance on that port.

Note If you mistakenly configure ports that are connected to hosts or other edge devices as spanning tree network ports, those ports automatically move into the blocking state.

Spanning Tree Normal Ports

Normal ports can be connected to either hosts, switches, or bridges. These ports function as normal spanning tree ports.

The default spanning tree interface is a normal port.

Understanding Bridge Assurance

You can use Bridge Assurance to protect against certain problems that can cause bridging loops in the network. Specifically, you use Bridge Assurance to protect against a unidirectional link failure and a device that continues to forward data traffic when it is no longer running the spanning tree algorithm.

Note Bridge Assurance is supported only by Rapid PVST+ and MST. Legacy 802.1D spanning tree does not support Bridge Assurance.

Bridge Assurance is enabled by default and can only be disabled globally. Also, Bridge Assurance can be enabled only on spanning tree network ports that are point-to-point links. Finally, both ends of the link must have Bridge Assurance enabled.

With Bridge Assurance enabled, BPDUs are sent out on all operational network ports, including alternate and backup ports, for each hello time period. If the port does not receive a BPDU for a specified period, the port moves into the blocking state and is not used in the root port calculation. Once that port receives a BPDU, it resumes the normal spanning tree transitions.

Understanding BPDU Guard

Enabling BPDU Guard shuts down that interface if a BPDU is received.

You can configure BPDU Guard at the interface level. When configured at the interface level, BPDU Guard shuts the port down as soon as the port receives a BPDU, regardless of the port type configuration.

When you configure BPDU Guard globally, it is effective only on operational spanning tree edge ports. In a valid configuration, LAN edge interfaces do not receive BPDUs. A BPDU that is received by an edge LAN interface signals an invalid configuration, such as the connection of an unauthorized host or switch. BPDU Guard, when enabled globally, shuts down all spanning tree edge ports when they receive a BPDU.

Note On the edge trunk interface level, if the remote side of the disabled VLAN is configured as an access port then the BPDUs will be ignored.

BPDU Guard provides a secure response to invalid configurations, because you must manually put the LAN interface back in service after an invalid configuration.

Note When enabled globally, BPDU Guard applies to all operational spanning tree edge interfaces.

Understanding BPDU Filtering

You can use BPDU Filtering to prevent the switch from sending or even receiving BPDUs on specified ports.

When configured globally, BPDU Filtering applies to all operational spanning tree edge ports. You should connect edge ports only to hosts, which typically drop BPDUs. If an operational spanning tree edge port receives a BPDU, it immediately returns to a normal spanning tree port type and moves through the regular transitions. In that case, BPDU Filtering is disabled on this port, and spanning tree resumes sending BPDUs on this port.

In addition, you can configure BPDU Filtering by the individual interface. When you explicitly configure BPDU Filtering on a port, that port does not send any BPDUs and drops all BPDUs that it receives. You can effectively override the global BPDU Filtering setting on individual ports by configuring the specific interface. This BPDU Filtering command on the interface applies to the entire interface, whether the interface is trunking or not.

Caution Use care when configuring BPDU Filtering per interface. If you explicitly configuring BPDU Filtering on a port that is not connected to a host, it can result in bridging loops because the port ignores any BPDU that it receives and goes to forwarding.

If the port configuration is not set to default BPDU Filtering, the edge configuration does not affect BPDU Filtering. The following table lists all the BPDU Filtering combinations.

Table 9: BPDU Filtering Configurations

BPDU Filtering Per Port Configuration	BPDU Filtering Global Configuration	STP Edge Port Configuration	BPDU Filtering State
Default	Enabled	Enabled	EnabledThe port transmits at least 10 BPDUs. If this port receives any BPDUs, the port returns to the spanning tree normal port state and BPDU Filtering is disabled.
Default	Enabled	Disabled	Disabled
Default	Disabled	Enabled/Disabled	Disabled
Disable	Enabled/Disabled	Enabled/Disabled	Disabled

BPDU Filtering Per Port Configuration	BPDU Filtering Global Configuration	STP Edge Port Configuration	BPDU Filtering State
Enabled	Enabled/Disabled	Enabled/Disabled	Enabled **Caution** BPDUs are never sent and if received, they do not trigger the regular STP behavior - use with caution.

Understanding Loop Guard

Loop Guard protects networks from loops that are caused by the following:

- Network interfaces that malfunction
- Busy CPUs
- Anything that prevents the normal forwarding of BPDUs

An STP loop occurs when a blocking port in a redundant topology erroneously transitions to the forwarding state. This transition usually happens because one of the ports in a physically redundant topology (not necessarily the blocking port) stops receiving BPDUs.

Loop Guard is useful only in switched networks where devices are connected by point-to-point links. On a point-to-point link, a designated bridge cannot disappear unless it sends an inferior BPDU or brings the link down.

Note Loop Guard can be enabled only on network and normal spanning tree port types.

You can use Loop Guard to determine if a root port or an alternate/backup root port receives BPDUs. If the port does not receive BPDUs, Loop Guard puts the port into an inconsistent state (blocking) until the port starts to receive BPDUs again. A port in the inconsistent state does not transmit BPDUs. If the port receives BPDUs again, the protocol removes its loop-inconsistent condition, and the STP determines the port state because such recovery is automatic.

Loop Guard isolates the failure and allows STP to converge to a stable topology without the failed link or bridge. Disabling Loop Guard moves all loop-inconsistent ports to the listening state.

You can enable Loop Guard on a per-port basis. When you enable Loop Guard on a port, it is automatically applied to all of the active instances or VLANs to which that port belongs. When you disable Loop Guard, it is disabled for the specified ports.

Understanding Root Guard

When you enable Root Guard on a port, Root Guard does not allow that port to become a root port. If a received BPDU triggers an STP convergence that makes that designated port become a root port, that port is put into a root-inconsistent (blocked) state. After the port stops sending superior BPDUs, the port is unblocked again. Through STP, the port moves to the forwarding state. Recovery is automatic.

Root Guard enabled on an interface applies this functionality to all VLANs to which that interface belongs.

You can use Root Guard to enforce the root bridge placement in the network. Root Guard ensures that the port on which Root Guard is enabled is the designated port. Normally, root bridge ports are all designated ports, unless two or more of the ports of the root bridge are connected. If the bridge receives superior BPDUs on a Root Guard-enabled port, the bridge moves this port to a root-inconsistent STP state. In this way, Root Guard enforces the position of the root bridge.

You cannot configure Root Guard globally.

Note You can enable Root Guard on all spanning tree port types: normal, edge, and network ports.

Configuring STP Extensions

STP Extensions Configuration Guidelines

When configuring STP extensions, follow these guidelines:

- Configure all access and trunk ports connected to hosts as edge ports.
- Bridge Assurance runs only on point-to-point spanning tree network ports. You must configure each side of the link for this feature.
- Loop Guard does not run on spanning tree edge ports.
- Enabling Loop Guard on ports that are not connected to a point-to-point link will not work.
- You cannot enable Loop Guard if Root Guard is enabled.

Configuring Spanning Tree Port Types Globally

The spanning tree port type designation depends on the type of device the port is connected to, as follows:

- Edge—Edge ports are connected to hosts and can be either an access port or a trunk port.
- Network—Network ports are connected only to switches or bridges.
- Normal—Normal ports are neither edge ports nor network ports; they are normal spanning tree ports. These ports can be connected to any type of device.

You can configure the port type either globally or per interface. By default, the spanning tree port type is normal.

Before you begin

Ensure that STP is configured.

Ensure that you are configuring the ports correctly for the type of device to which the interface is connected.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **spanning-tree port type edge default**	Configures all interfaces as edge ports. Using this command assumes all ports are connected to hosts/servers. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types.
Step 3	switch(config)# **spanning-tree port type network default**	Configures all interfaces as spanning tree network ports. Using this command assumes all ports are connected to switches and bridges. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types. **Note** If you configure interfaces connected to hosts as network ports, those ports automatically move into the blocking state.

Example

This example shows how to configure all access and trunk ports connected to hosts as spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge default
```

This example shows how to configure all ports connected to switches or bridges as spanning tree network ports:

```
switch# configure terminal
switch(config)# spanning-tree port type network default
```

Configuring Spanning Tree Edge Ports on Specified Interfaces

You can configure spanning tree edge ports on specified interfaces. Interfaces configured as spanning tree edge ports immediately transition to the forwarding state without passing through the blocking or learning states on linkup.

This command has four states:

- **spanning-tree port type edge**—This command explicitly enables edge behavior on the access port.
- **spanning-tree port type edge trunk**—This command explicitly enables edge behavior on the trunk port.

Note If you enter the **spanning-tree port type edge trunk** command, the port is configured as an edge port even in the access mode.

- **spanning-tree port type normal**—This command explicitly configures the port as a normal spanning tree port and the immediate transition to the forwarding state is not enabled.
- **no spanning-tree port type**—This command implicitly enables edge behavior if you define the **spanning-tree port type edge default** command in global configuration mode. If you do not configure the edge ports globally, the **no spanning-tree port type** command is equivalent to the **spanning-tree port type disable** command.

Before you begin

Ensure that STP is configured.

Ensure that the interface is connected to hosts.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **spanning-tree port type edge**	Configures the specified access interfaces to be spanning edge ports. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types.

Example

This example shows how to configure the Ethernet access interface 1/4 to be a spanning tree edge port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type edge
```

Configuring Spanning Tree Network Ports on Specified Interfaces

You can configure spanning tree network ports on specified interfaces.

Bridge Assurance runs only on spanning tree network ports.

This command has three states:

- **spanning-tree port type network**—This command explicitly configures the port as a network port. If you enable Bridge Assurance globally, it automatically runs on a spanning tree network port.
- **spanning-tree port type normal**—This command explicitly configures the port as a normal spanning tree port and Bridge Assurance cannot run on this interface.
- **no spanning-tree port type**—This command implicitly enables the port as a spanning tree network port if you define the **spanning-tree port type network default** command in global configuration mode. If you enable Bridge Assurance globally, it automatically runs on this port.

Note A port connected to a host that is configured as a network port automatically moves into the blocking state.

Before you begin

Ensure that STP is configured.

Ensure that the interface is connected to switches or routers.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** type slot/port	Specifies the interface to configure, and enters the interface configuration mode. The interface can be a physical Ethernet port.
Step 3	switch(config-if)# **spanning-tree port type network**	Configures the specified interfaces to be spanning network ports. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types.

Example

This example shows how to configure the Ethernet interface 1/4 to be a spanning tree network port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type network
```

Enabling BPDU Guard Globally

You can enable BPDU Guard globally by default. In this condition, the system shuts down an edge port that receives a BPDU.

Note We recommend that you enable BPDU Guard on all edge ports.

Before you begin

Ensure that STP is configured.

Ensure that you have configured some spanning tree edge ports.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree port type edge bpduguard default**	Enables BPDU Guard by default on all spanning tree edge ports. By default, global BPDU Guard is disabled.

Example

This example shows how to enable BPDU Guard on all spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge bpduguard default
```

Enabling BPDU Guard on Specified Interfaces

You can enable BPDU Guard on specified interfaces. Enabling BPDU Guard shuts down the port if it receives a BPDU.

You can configure BPDU Guard on specified interfaces as follows:

- **spanning-tree bpduguard enable**—Unconditionally enables BPDU Guard on the interface.
- **spanning-tree bpduguard disable**—Unconditionally disables BPDU Guard on the interface.
- **no spanning-tree bpduguard**—Enables BPDU Guard on the interface if it is an operational edge port and if the **spanning-tree port type edge bpduguard default** command is configured.

Before you begin

Ensure that STP is configured.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **interface** *type slot/port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **spanning-tree bpduguard** {**enable** | **disable**}	Enables or disables BPDU Guard for the specified spanning tree edge interface. By default, BPDU Guard is disabled on physical Ethernet interfaces.
Step 4	(Optional) switch(config-if)# **no spanning-tree bpduguard**	Disables BPDU Guard on the interface. **Note** Enables BPDU Guard on the interface if it is an operational edge port and if you enter the **spanning-tree port type edge bpduguard default** command.

Example

This example shows how to explicitly enable BPDU Guard on the Ethernet edge port 1/4:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree bpduguard enable
switch(config-if)# no spanning-tree bpduguard
```

Enabling BPDU Filtering Globally

You can enable BPDU Filtering globally by default on spanning tree edge ports.

If an edge port with BPDU Filtering enabled receives a BPDU, it loses its operation status and as edge port and resumes the regular STP transitions. However, this port maintains it configuration as an edge port.

Caution Be careful when using this command: using it incorrectly can cause bridging loops.

Note When enabled globally, BPDU Filtering is applied *only* on ports that are operational edge ports. Ports send a few BPDUs at linkup before they effectively filter outbound BPDUs. If a BPDU is received on an edge port, it immediately loses its operational edge port status and BPDU Filtering is disabled.

Before you begin

Ensure that STP is configured.

Ensure that you have configured some spanning tree edge ports.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree port type edge bpdufilter default**	Enables BPDU Filtering by default on all operational spanning tree edge ports. Global BPDU Filtering is disabled by default.

Example

This example shows how to enable BPDU Filtering on all operational spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge bpdufilter default
```

Enabling BPDU Filtering on Specified Interfaces

You can apply BPDU Filtering to specified interfaces. When enabled on an interface, that interface does not send any BPDUs and drops all BPDUs that it receives. This BPDU Filtering functionality applies to the entire interface, whether trunking or not.

Caution Be careful when you enter the **spanning-tree bpdufilter enable** command on specified interfaces. Explicitly configuring BPDU Filtering on a port that is not connected to a host can result in bridging loops because the port ignores any BPDU it receives and goes to forwarding.

You can enter this command to override the port configuration on specified interfaces.

This command has three states:

- **spanning-tree bpdufilter enable**—Unconditionally enables BPDU Filtering on the interface.
- **spanning-tree bpdufilter disable**—Unconditionally disables BPDU Filtering on the interface.
- **no spanning-tree bpdufilter**—Enables BPDU Filtering on the interface if the interface is an operational edge port and if you configure the **spanning-tree port type edge bpdufilter default** command.

Note When you enable BPDU Filtering locally on a port, this feature prevents the device from receiving or sending BPDUs on this port.

Before you begin

Ensure that STP is configured.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **spanning-tree bpdufilter** {**enable** | **disable**}	Enables or disables BPDU Filtering for the specified spanning tree edge interface. By default, BPDU Filtering is disabled.
Step 4	(Optional) switch(config-if)# **no spanning-tree bpdufilter**	Disables BPDU Filtering on the interface. **Note** Enables BPDU Filtering on the interface if the interface is an operational spanning tree edge port and if you enter the **spanning-tree port type edge bpdufilter default** command.

Example

This example shows how to explicitly enable BPDU Filtering on the Ethernet spanning tree edge port 1/4:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree bpdufilter enable
```

Enabling Loop Guard Globally

You can enable Loop Guard globally by default on all point-to-point spanning tree normal and network ports. Loop Guard does not run on edge ports.

Loop Guard provides additional security in the bridge network. Loop Guard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link.

Note Entering the Loop Guard command for the specified interface overrides the global Loop Guard command.

Before you begin

Ensure that STP is configured.

Ensure that you have spanning tree normal ports or have configured some network ports.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree loopguard default**	Enables Loop Guard by default on all spanning tree normal and network ports. By default, global Loop Guard is disabled.

Example

This example shows how to enable Loop Guard on all spanning tree normal or network ports:

```
switch# configure terminal
switch(config)# spanning-tree loopguard default
```

Enabling Loop Guard or Root Guard on Specified Interfaces

You can enable either Loop Guard or Root Guard on specified interfaces.

Enabling Root Guard on a port means that port cannot become a root port, and LoopGuard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link.

Both Loop Guard and Root Guard enabled on an interface apply to all VLANs to which that interface belongs.

Note Entering the Loop Guard command for the specified interface overrides the global Loop Guard command.

Before you begin

Ensure that STP is configured.

Ensure that you are configuring Loop Guard on spanning tree normal or network ports.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **spanning-tree guard** {**loop** | **root** | **none**}	Enables or disables either Loop Guard or Root Guard for the specified interface. By default, Root Guard is disabled by default, and Loop Guard on specified ports is also disabled. **Note** Loop Guard runs only on spanning tree normal and network interfaces.

Example

This example shows how to enable Root Guard on Ethernet port 1/4:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree guard root
```

Configuring FEX Port Spanning Tree BPDU Transmit Interval

You can configure the number of seconds between generation of the config Bridge Protocol Data Units (BPDUs) for FEX ports when they are connected to Cisco Nexus devices.

Before you begin

Enter the **spanning-tree bpdufilter disable** command if the FEX ports are connected to a Cisco Nexus device.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-id* **fex-hello-time** *fex-hello-time-value*	Configures the number of seconds between generation of config BPDUs for FEX ports. The fex-hello-time-value range is from 2 to 12. **Note** If the global **hello-time** is two seconds (default) and the **fex-hello-time** is two seconds, FEX port BPDU is not sent.

Example

The following examples show how to configure the number of seconds between generation of config BPDUs to 5 for VLAN 10..

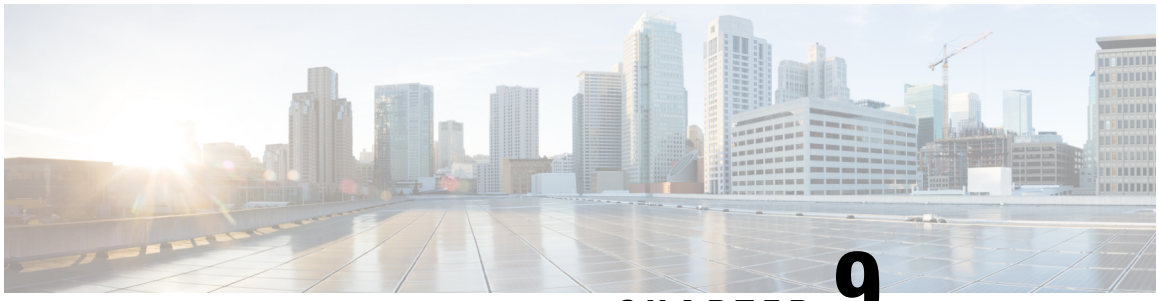
```
switch# configure terminal
switch(config)# spanning-tree vlan 10 fex-hello-time 5
```

Verifying the STP Extension Configuration

Use the following commands to display the configuration information for the STP extensions.

Command	Purpose
show running-config spanning-tree [**all**]	Displays the current status of spanning tree on the switch.

Command	Purpose
show spanning-tree [*options*]	Displays selected detailed information for the current spanning tree configuration.



CHAPTER 9

Configuring LLDP

This chapter contains the following sections:

Configuring LLDP

Before you begin

Ensure that the Link Layer Discovery Protocol (LLDP) feature is enabled on the switch.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **lldp** {**holdtime** *seconds* | **reinit** *seconds* | **timer** *seconds* | **tlv-select** {**dcbxp** | **management-address** | **port-description** | **port-vlan** | **system-capabilities** | **system-description** | **system-name**}}	Configures LLDP options. Use the **holdtime** option to set the length of time (10 to 255 seconds) that a device should save LLDP information received before discarding it. The default value is 120 seconds. Use the **reinit** option to set the length of time (1 to 10 seconds) to wait before performing LLDP initialization on any interface. The default value is 2 seconds. Use the **timer** option to set the rate (5 to 254 seconds) at which LLDP packets are sent. The default value is 30 seconds. Use the **tlv-select** option to specify the type length value (TLV). The default is enabled to send and receive all TLVs.

	Command or Action	Purpose
		Use the **dcbxp** option to specify the Data Center Ethernet Parameter Exchange (DCBXP) TLV messages. Use the **management-address** option to specify the management address TLV messages. Use the **port-description** option to specify the port description TLV messages. Use the **port-vlan** option to specify the port VLAN ID TLV messages. Use the **system-capabilities** option to specify the system capabilities TLV messages. Use the **system-description** option to specify the system description TLV messages. Use the **system-name** option to specify the system name TLV messages.
Step 3	switch(config)# **no lldp** {**holdtime** | **reinit** | **timer**}	Resets the LLDP values to their defaults.
Step 4	(Optional)switch# **show lldp**	Displays LLDP configurations.

Example

This example shows how to configure the global LLDP hold time to 200 seconds:

```
switch# configure terminal
switch(config)# lldp holdtime 200
switch(config)#
```

This example shows how to enable LLDP to send or receive the management address TLVs:

```
switch# configure terminal
switch(config)# lldp tlv-select management-address
switch(config)#
```

Configuring Interface LLDP

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Selects the interface to change.

	Command or Action	Purpose
Step 3	switch(config-if)# [**no**] **lldp** {**receive** | **transmit**}	Sets the selected interface to either receive or transmit. The **no** form of the command disables the LLDP transmit or receive.
Step 4	(Optional) switch# **show lldp** {**interface** | **neighbors** [**detail** | **interface** | **system-detail**] | **timers** | **traffic**}	Displays LLDP configurations.

Example

This example shows how to set an interface to transmit LLDP packets:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# lldp transmit
```

This example shows how to configure an interface to disable LLDP:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
```

This example shows how to display LLDP interface information:

```
switch# show lldp interface ethernet 1/2
tx_enabled: TRUE
rx_enabled: TRUE
dcbx_enabled: TRUE
Port MAC address:    00:0d:ec:a3:5f:48
Remote Peers Information
No remote peers exist
```

This example shows how to display LLDP neighbor information:

```
switch# show lldp neighbors
LLDP Neighbors
Remote Peers Information on interface Eth1/40
Remote peer's MSAP: length 12 Bytes:
00    c0    dd    0e    5f    3a    00    c0    dd    0e    5f    3a
LLDP TLV's
LLDP TLV type:Chassis ID  LLDP TLV Length: 7
LLDP TLV type:Port ID  LLDP TLV Length: 7
LLDP TLV type:Time to Live  LLDP TLV Length: 2
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 55
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 5
LLDP TLV type:END of LLDPDU  LLDP TLV Length: 0
Remote Peers Information on interface Eth1/34
```

```
Remote peer's MSAP: length 12 Bytes:
00    0d    ec    a3    27    40    00    0d    ec    a3    27    69
LLDP TLV's
LLDP TLV type:Chassis ID  LLDP TLV Length: 7
LLDP TLV type:Port ID  LLDP TLV Length: 7
LLDP TLV type:Time to Live  LLDP TLV Length: 2
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 55
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 5
LLDP TLV type:END of LLDPDU  LLDP TLV Length: 0
Remote Peers Information on interface Eth1/33
Remote peer's MSAP: length 12 Bytes:
00    0d    ec    a3    27    40    00    0d    ec    a3    27    68
LLDP TLV's
LLDP TLV type:Chassis ID  LLDP TLV Length: 7
LLDP TLV type:Port ID  LLDP TLV Length: 7
LLDP TLV type:Time to Live  LLDP TLV Length: 2
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 55
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 5
LLDP TLV type:END of LLDPDU  LLDP TLV Length: 0
```

This example shows how to display the system details about LLDP neighbors:

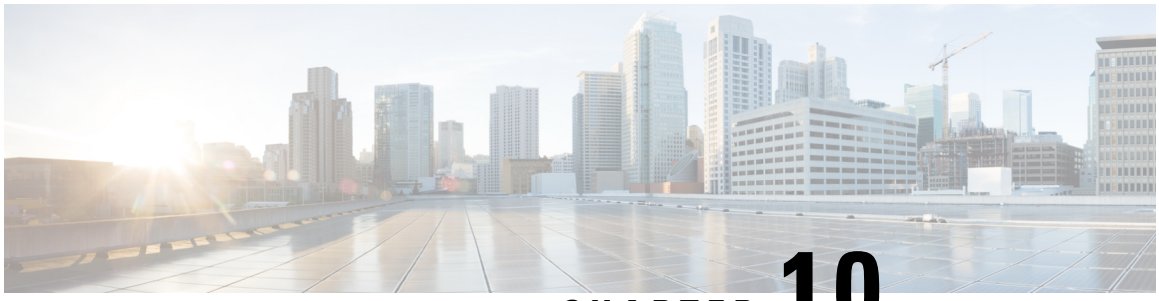
```
switch# sh lldp neighbors system-detail
Capability codes:
 (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
 (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID Local Intf Chassis ID PortID Hold-time Capability

switch-2  Eth1/7  0005.73b7.37ce Eth1/7  120 B
switch-3  Eth/9   0005.73b7.37d0 Eth1/9  120 B
switch-4  Eth1/10 0005.73b7.37d1 Eth1/10 120 B
Total entries displayed: 3
```

This example shows how to display LLDP timer information:

```
switch# show lldp timers

LLDP Timers

holdtime 120 seconds

reinit 2 seconds

msg_tx_interval 30 seconds
```

This example shows how to display information about LLDP counters:

```
switch# show lldp traffic

LLDP traffic statistics:


    Total frames out: 8464

    Total Entries aged: 6

    Total frames in: 6342

    Total frames received in error: 2

    Total frames discarded: 2

    Total TLVs unrecognized: 0
```



CHAPTER 10

Configuring MAC Address Tables

This chapter contains the following sections:

Information About MAC Addresses

To switch frames between LAN ports, the switch maintains an address table. When the switch receives a frame, it associates the media access control (MAC) address of the sending network device with the LAN port on which it was received.

The switch dynamically builds the address table by using the MAC source address of the frames received. When the switch receives a frame for a MAC destination address not listed in its address table, it floods the frame to all LAN ports of the same VLAN except the port that received the frame. When the destination station replies, the switch adds its relevant MAC source address and port ID to the address table. The switch then forwards subsequent frames to a single LAN port without flooding all LAN ports.

You can also enter a MAC address, which is termed a static MAC address, into the table. These static MAC entries are retained across a reboot of the switch.

RMAC Learning

Starting with Cisco NX-OS Release 7.2(0)N1(1), the RMAC Learning feature is supported on Cisco Nexus 5600 and 6000 series switches. This feature allows the default MAC address (RMAC) of a VLAN interface to be dynamically learned on another VLAN over a bridged interface on the switch. For example, consider two VLANs—VLAN X and VLAN Y—bridged over an external device. If a customer has a VLAN interface configured on VLAN Y, the MAC address of the interface will be dynamically learned on VLAN X.

Configuring MAC Addresses

Configuring Static MAC Addresses

You can configure static MAC addresses for the switch. These addresses can be configured in interface configuration mode or in VLAN configuration mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config) # **mac address-table static** *mac_address* **vlan** *vlan-id* {**drop** | **interface** {*type slot*/*port*} | **port-channel** *number*} [**auto-learn**]	Specifies a static address to add to the MAC address table. If you enable the **auto-learn** option, the switch will update the entry if the same MAC address is seen on a different port.
Step 3	(Optional) switch(config)# **no mac address-table static** *mac_address* **vlan** *vlan-id*	Deletes the static entry from the MAC address table. Use the **mac address-table static** command to assign a static MAC address to a virtual interface.

Example

This example shows how to put a static entry in the MAC address table:

```
switch# configure terminal
switch(config) # mac address-table static 12ab.47dd.ff89 vlan 3 interface ethernet 1/4
switch(config) #
```

Configuring the Aging Time for the MAC Table

You can configure the amount of time that an entry (the packet source MAC address and port that packet ingresses) remains in the MAC table. MAC aging time can be configured in either interface configuration mode or in VLAN configuration mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac address-table aging-time** *seconds* [**vlan** *vlan_id*]	Specifies the time before an entry ages out and is discarded from the MAC address table.

	Command or Action	Purpose
		The *seconds* range is from 0 to 1000000. The default is 300 seconds for Cisco NX-OS 5500 and 1800 for Cisco NX-OS 5600 and 6000 series. Entering the value 0 disables the MAC aging. If a VLAN is not specified, the aging specification applies to all VLANs.

Example

This example shows how to set the aging time for entries in the MAC address table to 300 seconds:

```
switch# configure terminal
switch(config) # mac address-table aging-time 300
switch(config) #
```

Configuring MAC Move Loop Detection

When the number of MAC address moves between two ports exceeds a threshold, it forms a loop. From Cisco NX-OS release 6.0(2)N2(1), you can configure the action of bringing down the port with the lower interface index when such a loop is detected by using the **mac address-table loop-detect port-down** command. To revert to the default action of disabling MAC learning, use the **no** form of this command.

Note If only the loop-detect port-down configuration is enabled, the last port on which MAC loop is detected is err-disabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] mac address-table loop-detect port-down**	Specifies the port-down action for MAC move loop detection. The **no** form of this command reverts to the default action of disabling MAC learning for 180 seconds.
Step 3	switch(config)# **mac address-table loop-detect port-down edge-port**	Enables the err-disabled detection for the edge-port on the MAC move loop detection.

Example

This example shows how to configure port-down as the action for MAC move loop detection.

```
switch# configure terminal
switch(config)# mac address-table loop-detect port-down
```

This example shows how to enable the err-disabled detection for the edge-port on the MAC move loop detection.

```
switch# configure terminal
switch(config)# mac address-table loop-detect port-down edge-port
```

Clearing Dynamic Addresses from the MAC Table

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **clear mac address-table dynamic** {**address** *mac-addr*} {**interface** [*type slot*/*port* | **port-channel** *number*} {**vlan** *vlan-id*}	Clears the dynamic address entries from the MAC address table.

Enabling RMAC Learning Feature

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] mac address-table router-mac learn-enable**	Enables/disables the RMAC learning feature. • You can use the **clear mac address-table dynamic** command to clear the learned MAC addresses.

Verifying the MAC Address Configuration

Use one of the following commands to verify the configuration:

Table 10: MAC Address Configuration Verification Commands

Command	Purpose
show mac address-table aging-time	Displays the MAC address aging time for all VLANs defined in the switch.
show mac address-table	Displays the contents of the MAC address table. **Note** IGMP snooping learned MAC addresses are not displayed.
show mac address-table loop-detect	Displays the currently configured action.

This example shows how to display the MAC address table:

```
switch# show mac address-table
VLAN      MAC Address       Type    Age       Port
---------+-----------------+-------+---------+------------------------------
1         0018.b967.3cd0    dynamic 10        Eth1/3
1         001c.b05a.5380    dynamic 200       Eth1/3
Total MAC Addresses: 2
```

This example shows how to display the current aging time:

```
switch# show mac address-table aging-time
Vlan  Aging Time
----- ----------
1     300
13    300
42    300
```

This example shows how to display the currently configured action:

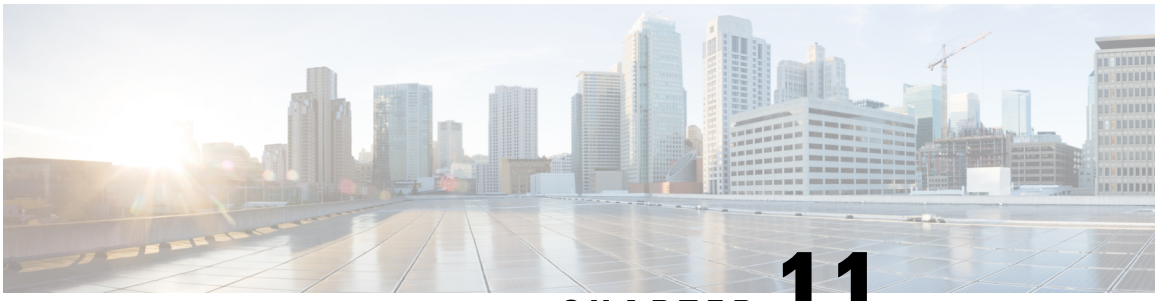
```
switch# configure terminal
switch(config)# show mac address-table loop-detect
Port Down Action Mac Loop Detect : enabled
```

```
switch# configure terminal
switch(config)# no mac address-table loop-detect port-down
switch(config)# show mac address-table loop-detect
Port Down Action Mac Loop Detect : disabled
```

Verifying RMAC Learning Feature

Use the **show mac address-table interface** *type slot/port* **vlan** *vlan_id* command to display the information about the MAC address table. In the sample output given below, RMAC is learned on Ethernet 1/33.

```
switch# show mac address-table interface ethernet 1/33 vlan 2

Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link
   VLAN     MAC Address      Type      age     Secure NTFY   Ports/SWID.SSID.LID
---------+-----------------+--------+---------+------+----+------------------
* 2        002a.6aca.b6bc    dynamic   20         F    F  Eth1/33
```



CHAPTER 11

Configuring IGMP Snooping

This chapter contains the following sections:

Information About IGMP Snooping

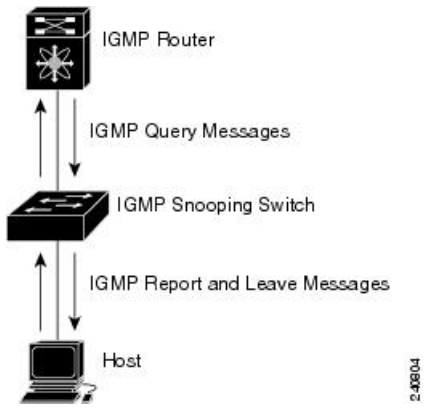
The IGMP snooping software examines IGMP protocol messages within a VLAN to discover which interfaces are connected to hosts or other devices interested in receiving this traffic. Using the interface information, IGMP snooping can reduce bandwidth consumption in a multiaccess LAN environment to avoid flooding the entire VLAN. The IGMP snooping feature tracks which ports are attached to multicast-capable routers to help it manage the forwarding of IGMP membership reports. The IGMP snooping software responds to topology change notifications.

Note IGMP snooping is supported on all Ethernet interfaces. The term *snooping* is used because Layer 3 control plane packets are intercepted and influence Layer 2 forwarding decisions.

Cisco NX-OS supports IGMPv2 and IGMPv3. IGMPv2 supports IGMPv1, and IGMPv3 supports IGMPv2. Although not all features of an earlier version of IGMP are supported, the features related to membership query and membership report messages are supported for all IGMP versions.

The following figure shows an IGMP snooping switch that is located between the host and the IGMP router. The IGMP snooping switch snoops the IGMP membership reports and leave messages and forwards them only when necessary to the connected IGMP routers.

Figure 20: IGMP Snooping Switch



Note The switch supports IGMPv3 snooping based only on the destination multicast MAC address. It does not support snooping based on the source MAC address or on proxy reports.

The Cisco NX-OS IGMP snooping software supports optimized multicast flooding (OMF) that forwards unknown traffic to routers only and performs no data driven state creation. For more information about IGMP snooping, see http://tools.ietf.org/wg/magma/draft-ietf-magma-snoop/rfc4541.txt.

IGMPv1 and IGMPv2

Both IGMPv1 and IGMPv2 support membership report suppression, which means that if two hosts on the same subnet want to receive multicast data for the same group, the host that receives a member report from the other host suppresses sending its report. Membership report suppression occurs for hosts that share a port.

If no more than one host is attached to each VLAN switch port, you can configure the fast leave feature in IGMPv2. The fast leave feature does not send last member query messages to hosts. As soon as the software receives an IGMP leave message, the software stops forwarding multicast data to that port.

IGMPv1 does not provide an explicit IGMP leave message, so the software must rely on the membership message timeout to indicate that no hosts remain that want to receive multicast data for a particular group.

Note Cisco NX-OS ignores the configuration of the last member query interval when you enable the fast leave feature because it does not check for remaining hosts.

IGMPv3

The IGMPv3 snooping implementation on the switch forwards IGMPv3 reports to allow the upstream multicast router to do source-based filtering.

By default, the software tracks hosts on each VLAN port. The explicit tracking feature provides a fast leave mechanism. Because every IGMPv3 host sends membership reports, a report suppression feature limits the amount of traffic the switch sends to other multicast-capable routers. When report suppression is enabled, and no IGMPv1 or IGMPv2 hosts request the same group, the software provides proxy reporting. The proxy

feature builds the group state from membership reports from the downstream hosts and generates membership reports in response to queries from upstream queriers.

Even though the IGMPv3 membership reports provide a full accounting of group members on a LAN segment, when the last host leaves, the software sends a membership query. You can configure the parameter last member query interval. If no host responds before the timeout, the software removes the group state.

IGMP Snooping Querier

When there is no multicast router in the VLAN to originate the queries, you must configure an IGMP snooping querier to send membership queries.

When an IGMP snooping querier is enabled, it sends out periodic IGMP queries that trigger IGMP report messages from hosts that want to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

Currently, you can configure the same SVI IP address for the switch querier and the IGMP snooping querier. Both queriers will then be active at the same time, and both queriers will send general queries to the VLAN periodically. To prevent this from happening, ensure that you use different IP addresses for the IGMP snooping querier and the switch querier.

IGMP Forwarding

The Cisco Nexus device supports snooping based on (S,G)/(*,G) IP addresses. Multicast MAC aliasing does not apply for Cisco Nexus devices and the snooped entries are programmed in the FIB tables and not in MAC table.

When a host connected to the switch wants to join an IP multicast group, it sends an unsolicited IGMP join message, specifying the IP multicast group to join. Alternatively, when the switch receives a general query from a connected router, it forwards the query to all interfaces, physical and virtual, in the VLAN. Hosts that want to join the multicast group respond by sending a join message to the switch. The switch CPU creates a multicast forwarding table entry for the group if it is not already present. The CPU also adds the interface where the join message was received to the forwarding table entry. The host associated with that interface receives multicast traffic for that multicast group.

The router sends periodic multicast general queries and the switch forwards these queries through all ports in the VLAN. Interested hosts respond to the queries. If at least one host in the VLAN wants to receive multicast traffic, the router continues forwarding the multicast traffic to the VLAN. The switch forwards multicast group traffic to only those hosts listed in the forwarding table for that multicast group.

When hosts want to leave a multicast group, they can either silently leave, or they can send a leave message. When the switch receives a leave message from a host, it sends a group-specific query to determine if any other devices connected to that interface are interested in traffic for the specific multicast group. The switch then updates the forwarding table for that (S,G) or (*,G) group so that only those hosts interested in receiving multicast traffic for the group are listed in the forwarding table. If the router receives no reports from a VLAN, it removes the group for the VLAN from its IGMP cache.

Guidelines and Limitations for IGMP Snooping

Consider the following configuration limitations with other features when configuring IGMP Snooping:

- IGMP snooping runs only on the primary VLAN and uses the configuration of the primary VLAN for all secondary VLANs.

  Any IGMP snooping join request in the secondary VLAN is treated as if it is received in the primary VLAN.

Configuring IGMP Snooping Parameters

To manage the operation of the IGMP snooping process, you can configure the optional IGMP snooping parameters described in the following table.

Table 11: IGMP Snooping Parameters

Parameter	Description
IGMP snooping	Enables IGMP snooping on a per-VLAN basis. The default is enabled. **Note** If the global setting is disabled, all VLANs are treated as disabled, whether they are enabled or not.
Explicit tracking	Tracks IGMPv2 and IPMPv3 membership reports from individual hosts for each port on a per-VLAN basis. The default is enabled.
Fast leave	Enables the software to remove the group state when it receives an IGMP Leave report without sending an IGMP query message. This parameter is used for IGMPv2 hosts when no more than one host is present on each VLAN port. The default is disabled.
Last member query interval	Sets the interval that the software waits after sending an IGMP query to verify that no hosts that want to receive a particular multicast group remain on a network segment. If no hosts respond before the last member query interval expires, the software removes the group from the associated VLAN port. Values range from 1 to 25 seconds. The default is 1 second.
Snooping querier	Configures a snooping querier on an interface when there is no multicast router in the VLAN to generate queries. The default is disabled.
Report suppression	Limits the membership report traffic sent to multicast-capable routers. When you disable report suppression, all IGMP reports are sent as is to multicast-capable routers. The default is enabled.
Multicast router	Configures a static connection to a multicast router. The interface to the router must be in the selected VLAN.

Parameter	Description
Multicast router vpc-peer-link	Configures a static connection to a virtual port channel (vPC) peer link. By default, the vPC peer link is considered a multicast router port and the multicast packet is sent to the peer link for each receiver VLAN. To send the multicast traffic over a vPC peer link to each receiver VLAN that has orphan ports, use the **no ip igmp snooping mrouter vpc-peer-link** command. If you use the **no ip igmp snooping mrouter vpc-peer-link** command, the multicast traffic is not sent over to a peer link for the source VLAN and receiver VLAN unless there is an orphan port in the VLAN. The IGMP snooping mrouter VPC peer link should also be globally disabled on the peer VPC switch.
Static group	Configures an interface that belongs to a VLAN as a static member of a multicast group.

You can disable IGMP snooping either globally or for a specific VLAN.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip igmp snooping**	Globally enables IGMP snooping. The default is enabled. **Note** If the global setting is disabled, all VLANs are treated as disabled, whether they are enabled or not.
Step 3	switch(config)# **vlan configuration** *vlan-id*	Enters VLAN configuration mode.
Step 4	switch(config-vlan)# **ip igmp snooping**	Enables IGMP snooping for the current VLAN. The default is enabled. **Note** If IGMP snooping is enabled globally, this command is not required.
Step 5	switch(config-vlan)# **ip igmp snooping explicit-tracking**	Tracks IGMPv2 and IGMPv3 membership reports from individual hosts for each port on a per-VLAN basis. The default is enabled on all VLANs.
Step 6	switch(config-vlan)# **ip igmp snooping fast-leave**	Supports IGMPv2 hosts that cannot be explicitly tracked because of the host report suppression mechanism of the IGMPv2 protocol. When you enable fast leave, the IGMP software assumes that no more than one host is present on each VLAN port. The default is disabled for all VLANs.

	Command or Action	Purpose
Step 7	switch(config-vlan)# **ip igmp snooping last-member-query-interval** *seconds*	Removes the group from the associated VLAN port if no hosts respond to an IGMP query message before the last member query interval expires. Values range from 1 to 25 seconds. The default is 1 second.
Step 8	switch(config-vlan)# **ip igmp snooping querier** *IP-address*	Configures a snooping querier when you do not enable PIM because multicast traffic does not need to be routed. The IP address is used as the source in messages. The default is disabled.
Step 9	switch(config-vlan)# **ip igmp snooping report-suppression**	Limits the membership report traffic sent to multicast-capable routers. When you disable report suppression, all IGMP reports are sent as is to multicast-capable routers. The default is enabled.
Step 10	switch(config-vlan)# **ip igmp snooping mrouter interface** *interface*	Configures a static connection to a multicast router. The interface to the router must be in the selected VLAN. You can specify the interface by type and number.
Step 11	switch(config-vlan)# **ip igmp snooping mrouter vpc-peer-link**	Configures a static connection to a virtual port channel (vPC) peer link. By default, the vPC peer link is considered as a multicast router port and the multicast packet is sent to the peer link for each receiver VLAN. To send the multicast traffic over a vPC peer link to each receiver VLAN that has orphan ports, use the **no ip igmp snooping mrouter vpc-peer-link** command. The IGMP snooping mrouter VPC peer link should also be globally disabled on the peer VPC switch.
Step 12	switch(config-vlan)# **ip igmp snooping static-group** *group-ip-addr* [**source** *source-ip-addr*] **interface** *interface*	Configures an interface belonging to a VLAN as a static member of a multicast group. You can specify the interface by type and number.

Example

This example shows how to configure IGMP snooping parameters for a VLAN:

```
switch# configure terminal
switch(config)# vlan configuration 5
switch(config-vlan)# ip igmp snooping last-member-query-interval 3
switch(config-vlan)# ip igmp snooping querier 172.20.52.106
switch(config-vlan)# ip igmp snooping explicit-tracking
switch(config-vlan)# ip igmp snooping fast-leave
switch(config-vlan)# ip igmp snooping report-suppression
switch(config-vlan)# ip igmp snooping mrouter interface ethernet 1/10
```

```
switch(config-vlan)# ip igmp snooping static-group 230.0.0.1 interface ethernet 1/10
switch(config-vlan)# end
```

Verifying the IGMP Snooping Configuration

Use the following commands to verify the IGMP snooping configuration.

Command	Description
show ip igmp snooping [[**vlan**] *vlan-id*]	Displays IGMP snooping configuration by VLAN.
show ip igmp snooping groups [[**vlan**] *vlan-id*] [**detail**]	Displays IGMP snooping information about groups by VLAN.
show ip igmp snooping querier [[**vlan**] *vlan-id*]	Displays IGMP snooping queriers by VLAN.
show ip igmp snooping mrouter [[**vlan**] *vlan-id*]	Displays multicast router ports by VLAN.
show ip igmp snooping explicit-tracking vlan *vlan-id*	Displays IGMP snooping explicit tracking information by VLAN.

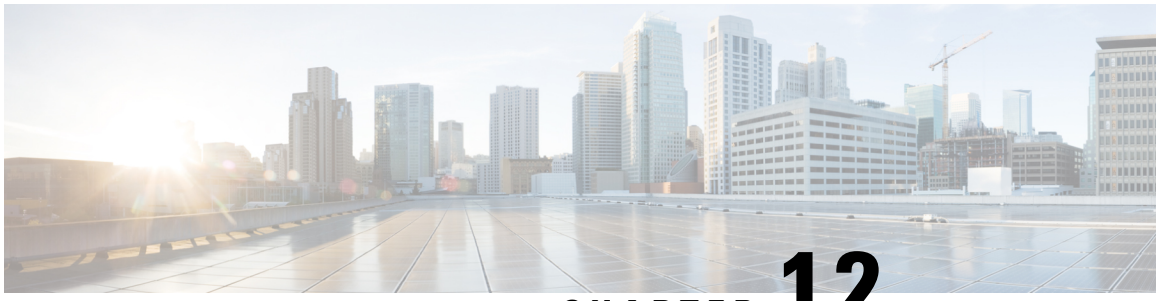
Note **VPC behavior for v2 EHT**: In a VPC scenario, the explicit host tracking is not synced to the VPC peer. However in a VPC peer, the EHT is also learned by cfs sync and is displayed by using the detail option.

This example shows how to verify the IGMP snooping parameters:

```
switch# show ip igmp snooping
Global IGMP Snooping Information:
  IGMP Snooping enabled
IGMP Snooping information for vlan 1
  IGMP snooping enabled
  IGMP querier none
  Switch-querier disabled
  Explicit tracking enabled
  Fast leave disabled
  Report suppression enabled
  Router port detection using PIM Hellos, IGMP Queries
  Number of router-ports: 0
  Number of groups: 0
IGMP Snooping information for vlan 5
IGMP snooping enabled
  IGMP querier present, address: 192.0.2.1, version: 3
  Querier interval: 125 secs
  Querier last member query interval: 10 secs
  Querier robustness: 2
  Switch-querier enabled, address 192.0.2.1, currently running
  Explicit tracking enabled
  Fast leave enabled
  Report suppression enabled
  Router port detection using PIM Hellos, IGMP Queries
  Number of router-ports: 1
  Number of groups: 1
```

This example shows how to display the IGMP snooping configuration for explicit tracking on an IGMPv2 host:

```
switch# show ip igmp snooping explicit tracking
IGMP Snooping Explicit-tracking information
Vlan Source/Group
     Intf           Reporter      Uptime    Last-Join Expires   Ver  Reports
100  */225.1.1.69
     Eth1/43        10.1.1.2      00:00:02  00:00:02  00:04:17  v2   1
100  */225.1.1.70
     Eth1/43        10.1.1.2      00:00:02  00:00:02  00:04:17  v2   1
100  */225.1.1.71
     Eth1/43        10.1.1.2      00:00:02  00:00:02  00:04:17  v2   1
100  */225.1.1.72
     Eth1/43        10.1.1.2      00:00:02  00:00:02  00:04:17  v2   1
100  */225.1.1.73
     Eth1/43        10.1.1.2      00:00:02  00:00:02  00:04:17  v2   1
100  */225.1.1.74
     Eth1/43        10.1.1.2      00:00:02  00:00:02  00:04:17  v2   1
100  */225.1.1.75
     Eth1/43        10.1.1.2      00:00:02  00:00:02  00:04:17  v2   1
100  */225.1.1.76
     Eth1/43        10.1.1.2      00:00:02  00:00:02  00:04:17  v2   1
100  */225.1.1.77
     Eth1/43        10.1.1.2      00:00:02  00:00:02  00:04:17  v2   1
100  */225.1.1.78
     Eth1/43        10.1.1.2      00:00:02  00:00:02  00:04:17  v2   1
switch#:
```



CHAPTER 12

Configuring MVR

This chapter contains the following sections:

Information About MVR

MVR Overview

In a typical Layer 2 multi-VLAN network, subscribers to a multicast group can be on multiple VLANs. To maintain data isolation between these VLANs, the multicast stream on the source VLAN must be passed to a router, which replicates the stream on all subscriber VLANs, wasting upstream bandwidth.

Multicast VLAN Registration (MVR) allows a Layer 2 switch to forward the multicast data from a source on a common assigned VLAN to the subscriber VLANs, conserving upstream bandwidth by bypassing the router. The switch forwards multicast data for MVR IP multicast streams only to MVR ports on which hosts have joined, either by IGMP reports or by MVR static configuration. The switch forwards IGMP reports received from MVR hosts only to the source port. For other traffic, VLAN isolation is preserved.

MVR requires at least one VLAN to be designated as the common VLAN to carry the multicast stream from the source. More than one such multicast VLAN (MVR VLAN) can be configured in the system, and you can configure a global default MVR VLAN as well as interface-specific default MVR VLANs. Each multicast group using MVR is assigned to an MVR VLAN.

MVR allows a subscriber on a port to subscribe and unsubscribe to a multicast stream on the MVR VLAN by sending IGMP join and leave messages. IGMP leave messages from an MVR group are handled according to the IGMP configuration of the VLAN on which the leave message is received. If IGMP fast leave is enabled on the VLAN, the port is removed immediately; otherwise, an IGMP query is sent to the group to determine whether other hosts are present on the port.

MVR Interoperation with Other Features

MVR and IGMP Snooping

Although MVR operates on the underlying mechanism of IGMP snooping, the two features operate independently of each other. One feature can be enabled or disabled without affecting the operation of the other feature. If IGMP snooping is disabled globally or on a VLAN, and if MVR is enabled on the VLAN, IGMP snooping is internally enabled on the VLAN. Joins received for MVR groups on non-MVR receiver ports, or joins received for non-MVR groups on MVR receiver ports, are processed by IGMP snooping.

MVR and vPC

- As with IGMP snooping, IGMP control messages received by virtual port channel (vPC) peer switches are exchanged between the peers, allowing synchronization of MVR group information.
- MVR configuration must be consistent between the peers.
- The **no ip igmp snooping mrouter vpc-peer-link** command applies to MVR. With this command, multicast traffic is not sent over to a peer link for the source VLAN and receiver VLAN unless there is an orphan port in the VLAN.
- The **show mvr member** command shows the multicast group on the vPC peer switch. However, the vPC peer switch does not show the multicast groups if it does not receive the IGMP membership report of the groups.

Licensing Requirements for MVR

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	This feature does not require a license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations for MVR

When configuring MVR, follow these guidelines:

- MVR is supported only on Layer 2 Ethernet ports, such as individual ports, port channels, and virtual Ethernet (vEth) ports.
- MVR receiver ports can only be access ports; they cannot be trunk ports. MVR source ports can be either access or trunk ports.
- MVR configuration on Flex Link ports is not supported.
- Priority tagging is not supported on MVR receiver ports.
- When using private VLANs (PVLANs), you cannot configure a secondary VLAN as the MVR VLAN.

- The total number of MVR VLANs cannot exceed 250.

Note During and in-service software upgrade (ISSU), MVR IGMP membership for the MVR receiver ports may timeout because the joins are not forwarded to the upstream router. In order to avoid a timeout, the querier timer on the upstream router or the network querier should be increased to accommodate an ISSU.

Default MVR Settings

Parameter	Default
MVR	Disabled globally and per interface
Global MVR VLAN	None configured
Interface (per port) default	Neither a receiver nor a source port

Configuring MVR

Configuring MVR Global Parameters

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **mvr**	Globally enables MVR. The default is disabled. Use the **no** form of the command to disable MVR.
Step 3	switch(config)# [**no**] **mvr-vlan** *vlan-id*	Specifies the global default MVR VLAN. The MVR VLAN is the source of the multicast message that subsequent receivers subscribe to. The range is 1 to 4094. Use the **no** form of the command to clear the MVR VLAN.
Step 4	switch(config)# [**no**] **mvr-group** *addr*[/*mask*] [**count** *groups*] [**vlan** *vlan-id*]	Adds a multicast group at the specified IPv4 address and (optional) netmask length to the global default MVR VLAN. You can repeat this command to add additional groups to the MVR VLAN.

	Command or Action	Purpose
		The IP address is entered in the format *a.b.c.d/m*, where *m* is the number of bits in the netmask, from 1 to 31. (Optional) You can specify a number of MVR groups using contiguous multicast IP addresses starting with the specified IP address. Use the **count** keyword followed by a number from 1 to 64. (Optional) You can explicitly specify an MVR VLAN for the group by using the **vlan** keyword; otherwise, the group is assigned to the default MVR VLAN. Use the **no** form of the command to clear the group configuration.
Step 5	(Optional) switch(config)# **end**	Returns to privileged EXEC mode.
Step 6	(Optional) switch# **clear mvr counters** [**source-ports** | **receiver-ports**]	Clears MVR IGMP packet counters.
Step 7	(Optional) switch# **show mvr**	Displays the global MVR configuration.
Step 8	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to globally enable MVR and configure the global parameters:

```
switch# configure terminal
switch(config)# mvr
switch(config-mvr)# mvr-vlan 100
switch(config-mvr)# mvr-group 230.1.1.1 count 4
switch(config-mvr)# mvr-group 228.1.2.240/28 vlan 101
switch(config-mvr)# mvr-group 235.1.1.6 vlan 340
switch(config-mvr)# end
switch# show mvr
MVR Status           : enabled
Global MVR VLAN      : 100
Number of MVR VLANs  : 3
switch# copy running-config startup-config
```

Configuring MVR Interfaces

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**mvr**	Globally enables MVR. The default is disabled. **Note** If MVR is enabled globally, then this command is not required.
Step 3	**interface** {**ethernet** *type slot/port* | **port-channel** *channel-number* | **vethernet** *number*}	Specifies the Layer 2 port to configure, and enters interface configuration mode.
Step 4	[**no**] **mvr-type** {**source** | **receiver**}	Configures an MVR port as one of these types of ports: • **source**—An uplink port that sends and receives multicast data is configured as an MVR source. The port automatically becomes a static receiver of MVR multicast groups. A source port should be a member of the MVR VLAN. • **receiver**— An access port that is connected to a host that wants to subscribe to an MVR multicast group is configured as an MVR receiver. A receiver port receives data only when it becomes a member of the multicast group by using IGMP leave and join messages. If you attempt to configure a non-MVR port with MVR characteristics, the configuration is cached and does not take effect until the port becomes an MVR port. The default port mode is non-MVR.
Step 5	(Optional) [**no**] **mvr-vlan** *vlan-id*	Specifies an interface default MVR VLAN that overrides the global default MVR VLAN for joins received on the interface. The MVR VLAN is the source of the multicast message that subsequent receivers subscribe to. The range is 1 to 4094.
Step 6	(Optional) [**no**] **mvr-group** *addr*[/*mask*] [**vlan** *vlan-id*]	Adds a multicast group at the specified IPv4 address and (optional) netmask length to the interface MVR VLAN, overriding the global MVR group configuration. You can repeat this

	Command or Action	Purpose
		command to add additional groups to the MVR VLAN The IP address is entered in the format *a.b.c.d/m*, where *m* is the number of bits in the netmask, from 1 to 31. (Optional) You can explicitly specify an MVR VLAN for the group by using the **vlan** keyword; otherwise, the group is assigned to the interface default (if specified) or global default MVR VLAN. Use the **no** form of the command to clear the IPv4 address and netmask.
Step 7	(Optional) **end**	Return to privileged EXEC mode.
Step 8	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure an Ethernet port as an MVR receiver port:

```
switch# configure terminal
switch(config)# mvr
switch(config-mvr)# interface ethernet 1/10
switch(config-if)# mvr-group 225.1.3.1 vlan 100
switch(config-if)# mvr-type receiver
switch(config-if)# end
switch# copy running-config startup-config
switch#
```

Verifying the MVR Configuration

Use the following commands to verify the MVR configuration:

Command	Description
show mvr	Displays the MVR subsystem configuration and status.
show mvr groups	Displays the MVR group configuration.
show ip igmp snooping [**vlan** *vlan-id*]	Displays information about IGMP snooping on the specified VLAN.
show mvr interface {**ethernet** *type slot/port* | **port-channel** *number*}	Displays the MVR configuration on the specified interface.

Command	Description
show mvr members [**count**]	Displays the number and details of all MVR receiver members.
show mvr members interface {**ethernet** *type slot/port* | **port-channel** *number*}	Displays details of MVR members on the specified interface.
show mvr members vlan *vlan-id*	Displays details of MVR members on the specified VLAN.
show mvr receiver-ports [**ethernet** *type slot/port* | **port-channel** *number*]	Displays all MVR receiver ports on all interfaces or on the specified interface.
show mvr source-ports [**ethernet** *type slot/port* | **port-channel** *number*]	Displays all MVR source ports on all interfaces or on the specified interface.

This example shows how to verify the MVR parameters:

```
switch# show mvr
MVR Status          : enabled
Global MVR VLAN     : 100
Number of MVR VLANs : 4
```

This example shows how to verify the MVR group configuration:

```
switch# show mvr groups
* - Global default MVR VLAN.

Group start    Group end       Count  MVR-VLAN Interface
                               Mask
-------------  --------------- ------ -------- -----------
228.1.2.240    228.1.2.255     /28     101
230.1.1.1      230.1.1.4        4     *100
235.1.1.6      235.1.1.6        1      340
225.1.3.1      225.1.3.1        1     *100     Eth1/10
```

This example shows how to verify the MVR interface configuration and status:

```
switch# show mvr interface
Port         VLAN Type      Status    MVR-VLAN
----         ---- ----      ------    --------
Po10         100  SOURCE    ACTIVE    100-101
Po201        201  RECEIVER  ACTIVE    100-101,340
Po202        202  RECEIVER  ACTIVE    100-101,340
Po203        203  RECEIVER  ACTIVE    100-101,340
Po204        204  RECEIVER  INACTIVE  100-101,340
Po205        205  RECEIVER  ACTIVE    100-101,340
Po206        206  RECEIVER  ACTIVE    100-101,340
Po207        207  RECEIVER  ACTIVE    100-101,340
Po208        208  RECEIVER  ACTIVE    2000-2001
Eth1/9       340  SOURCE    ACTIVE    340
Eth1/10      20   RECEIVER  ACTIVE    100-101,340
Eth2/2       20   RECEIVER  ACTIVE    100-101,340
Eth102/1/1   102  RECEIVER  ACTIVE    100-101,340
Eth102/1/2   102  RECEIVER  INACTIVE  100-101,340
Eth103/1/1   103  RECEIVER  ACTIVE    100-101,340
Eth103/1/2   103  RECEIVER  ACTIVE    100-101,340

Status INVALID indicates one of the following misconfiguration:
```

```
a) Interface is not a switchport.
b) MVR receiver is not in access, pvlan host or pvlan promiscuous mode.
c) MVR source is in fex-fabric mode.
```

This example shows how to display all MVR members:

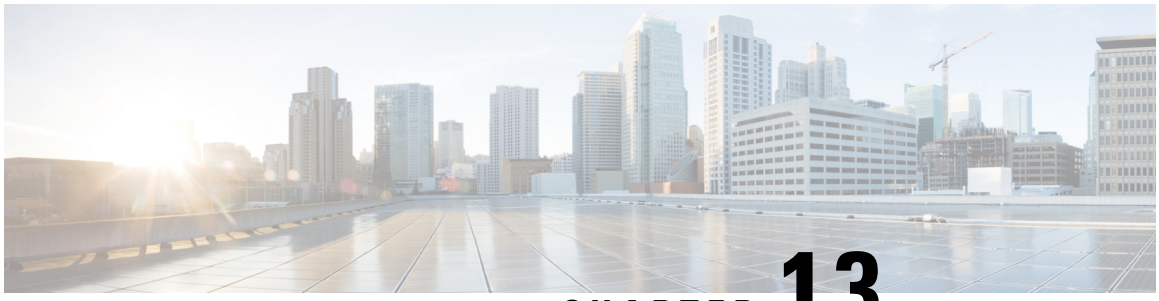
```
switch# show mvr members
MVR-VLAN  Group Address  Status   Members
--------  -------------  -------  -------
100       230.1.1.1      ACTIVE    Po201 Po202 Po203 Po205 Po206
100       230.1.1.2      ACTIVE    Po205 Po206 Po207 Po208
340       235.1.1.6      ACTIVE    Eth102/1/1
101       225.1.3.1      ACTIVE    Eth1/10 Eth2/2
101       228.1.2.241    ACTIVE    Eth103/1/1 Eth103/1/2
```

This example shows how to display all MVR receiver ports on all interfaces:

```
switch# show mvr receiver-ports
Port          MVR-VLAN  Status   Joins         Leaves
                                  (v1,v2,v3)
------------  --------  -------- ------------  ------------
Po201         100       ACTIVE   8             2
Po202         100       ACTIVE   8             2
Po203         100       ACTIVE   8             2
Po204         100       INACTIVE 0             0
Po205         100       ACTIVE   10            6
Po206         100       ACTIVE   10            6
Po207         100       ACTIVE   5             0
Po208         100       ACTIVE   6             0
Eth1/10       101       ACTIVE   12            2
Eth2/2        101       ACTIVE   12            2
Eth102/1/1    340       ACTIVE   16            15
Eth102/1/2    340       INACTIVE 16            16
Eth103/1/1    101       ACTIVE   33            0
Eth103/1/2    101       ACTIVE   33            0
```

This example shows how to display all MVR source ports on all interfaces:

```
switch# show mvr source-ports
Port          MVR-VLAN  Status
------------  --------  --------
Po10          100       ACTIVE
Eth1/9        340       ACTIVE
```



CHAPTER 13

Configuring VTP V3

This chapter contains the following sections:

Configuring VTP V3

From Cisco NX-OS Release 7.2(0)N1(1), VLAN Trunk Protocol (VTP) V3 supports PVLAN integration, 4K VLAN integration, generic database transport mechanism, and VTP authentication mechanism.

VTP V3 Overview

VTP V3 allows each router or LAN device to transmit advertisements in frames on its trunk ports. These frames are sent to a multicast address where they can be received by all neighboring devices. They are not forwarded by normal bridging procedures. An advertisement lists the sending device's VTP management domain, its configuration revision number, the VLANs which it knows about, and certain parameters for each known VLAN. By hearing these advertisements, all devices in the same management domain learn about any new VLANs that are configured in the transmitting device. This process allows you to create and configure a new VLAN only on one device in the management domain, and then that information is automatically learned by all the other devices in the same management domain.

Once a device learns about a VLAN, the device receives all frames on that VLAN from any trunk port by default, and if appropriate, forwards them to each of its other trunk ports, if any. This process prevents unnecessary VLAN traffic from being sent to a device. An extension of VTP called VTP pruning has been defined to limit the scope of broadcast traffic and save bandwidth. Beginning with Release 5.1(1), the Cisco NX-OS software supports VTP pruning.

VTP also publishes information about the domain and the mode in a shared local database that can be read by other processes such as Cisco Discovery Protocol (CDP).

VTP V3 Modes

From Cisco NX-OS Release 7.2(0)N1(1), VTP V3 supports the following modes:

- Transparent—Allows you to relay all VTP protocol packets that it receives on a trunk port to all other trunk ports. When you create or modify a VLAN that is in VTP transparent mode, those VLAN changes affect only the local device. A VTP transparent network device does not advertise its VLAN configuration

and does not synchronize its VLAN configuration based on received advertisements. You cannot configure VLANs 1002 to 1005 in VTP client/server mode because these VLANs are reserved for Token Ring.

- Server— Allows you to create, remove, and modify VLANs over the entire network. You can set other configuration options like the VTP version and also turn on or off VTP pruning for the entire VTP domain. VTP servers advertise their VLAN configuration to other switches in the same VTP domain and synchronize their VLAN configuration with other switches based on messages received over trunk links. Beginning with Release 5.1(1), the server mode is the default mode. The VLAN information is stored on the bootflash and is not erased after a reboot.
- Client— Allows you to create, change, and delete VLANs on the local device. In VTP client mode, a switch stores the last known VTP information including the configuration revision number, on the bootflash. A VTP client might or might not start with a new configuration when it powers up.
- Off— Behaves similarly to the transparent mode but does not forward any VTP packets. The off mode allows you to monitor VLANs by using the CISCO-VTP-MIB without having to run VTP. On Cisco Nexus 7000 Series devices, because VTP is a conditional service, its MIB is loaded only when the corresponding feature is enabled. The CISCO-VTP-MIB does not follow this convention. It is loaded by the VLAN manager and will always return the correct values whether the VTP process is enabled or disabled.

Note VTP client will move to transparent mode if there is any failure during updating VLAN database received from server. Following syslog message is displayed on console. `"VTP-2-VTP_MODE_TRANSPARENT_CREATE_SEQ_FAILED: VTP Mode changed to transparent since VTP vlan create/update failed"`. User need to change back the VTP mode to client to get latest database from server.

VTP V3 Pruning

The VLAN architecture requires all flooded traffic for a VLAN to be sent across a trunk port even if it leads to switches that have no devices that are active in the VLAN. This method leads to wasted network bandwidth.

VTP V3 Pruning optimizes the usage of network bandwidth by restricting the flooded traffic to only those trunk ports that can reach all the active network devices. When this protocol is in use, a trunk port does not receive the flooded traffic that is meant for a certain VLAN unless an appropriate join message is received.

A join message is defined as a new message type in addition to the ones already supported by version 1 of the VTP V3 protocol. A VTP V3 implementation indicates that it supports this extension by appending a special TLV at the end of the summary advertisement messages that it generates. In VTP V3 transparent mode, VTP relays all VTP packets, and pruning requires that the switch processes TLVs in the VTP V3 summary packets.

VTP V3 Per Interface

VTP allows you to enable or disable the VTP protocol on a per-port basis to control the VTP traffic. When a trunk is connected to a switch or end device, it drops incoming VTP packets and prevents VTP advertisements on this particular trunk. By default, VTP is enabled on all the switch ports.

VTP V3 Pruning and Spanning Tree Protocol

VTP maintains a list of trunk ports in the Spanning Tree Protocol (STP) forwarding state by querying STP at bootup and listening to the notifications that are generated by STP.

VTP sets a trunk port into the pruned or joined state by interacting with STP. STP notifies VTP V3 when a trunk port goes to the blocking or forwarding state. VTP V3 notifies STP when a trunk port becomes pruned or joined.

Configuring VTP V3

Note

- VLAN 1 is required on all trunk ports used for switch interconnects if VTP V3 is used in transparent mode in the network. Disabling VLAN 1 from any of these ports prevents VTP from functioning properly in transparent mode.
- The overlapping of system reserved VLANs between Cisco Nexus 6000 Series Switches and Cisco Nexus 5000 Series Switches causes interoperability issues. When a Cisco Catalyst 6000 Switch sends a VLAN reserved in the Cisco Nexus switch, it causes the Cisco Nexus 5000 and 6000 Series Switches to move to VTP Transparent Mode.

  Check the generated syslog messages in the Cisco Nexus Switches for information on the VLAN that caused the interoperability issue.

Before you begin

Ensure that you are in the correct virtual device context (VDC) (or enter the **switchto vdc** command). VLAN names and IDs can be repeated in different VDCs, so you must confirm which VDC that you are working in.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature vtp**	Enables VTP on the device. The default is disabled.
Step 3	switch(config)# **vtp domain** *domain-name*	Specifies the name of the VTP domain that you want this device to join. The default is blank.
Step 4	switch(config)# **vtp version** {**1** | **2** | **3**}	Sets the VTP version that you want to use. The default is version 1.
Step 5	Required: switch(config)# **vtp mode** {**client** | **server** | **transparent** | **off**} [**vlan** | **mst** | **unknown**]	Sets the VTP mode to client, server, transparent, or off. The default server mode is for vlan instance and transparent is for mst instance.

	Command or Action	Purpose
Step 6	switch(config)# **vtp interface** *interface-name* [**only**]	Configures the interface name used by the VTP updater for this device.
Step 7	switch(config)# **vtp file** *file-name*	Specifies the ASCII filename of the IFS file system file where the VTP configuration is stored.

Step 8	switch(config)# **vtp password** *password-value* [**hidden** | **secret**]
	Example:
	For Hidden:
	`Device(config)# vtp password helping hidden`
	`Generating the secret associated to the password.`
	`Device# exit`
	`Device# show vtp password`
	`VTP Password:`
	`89914640C8D90868B6A0D8103847A733`
	Example:
	For Secret:
	`Device(config)# vtp password 89914640C8D90868B6A0D8103847A733 secret`
	`Device# exit`
	`Device# show vtp password`
	`VTP Password:`
	`89914640C8D90868B6A0D8103847A733`
		Specifies the password for the VTP administrative domain. Default value is taken from vlan.dat.
		The following options are applicable only on VTP V3:
		• Hidden–Password is not saved as clear text in vlan.data file. Instead, a hexadecimal secret key generated from the password is saved. This is displayed as the output of the **show vtp password**.
		• Secret–Use this keyword to directly configure the 32-character hexadecimalsecret key. System administrators can distribute this secret key instead of the clear text password.
		Note This command is applicable for VTP version 3 only.

Step 9	switch(config)# **exit**	Exits the configuration submode.

Step 10	switch# **vtp primary** [*feature*] [**force**]
	Example:
	`Device# vtp primary vlan`
	`Enter VTP password:`
	`This switch is becoming Primary server for vlan feature in the VTP domain`
	`VTP Database Conf Switch ID Primary Server Revision System Name`
	`------------ ---- -------------- -------------- --------`
	`VLANDB Yes 00d0.00b8.1400=00d0.00b8.1400 1 stp7`
	`Do you want to continue (y/n) [n]? y`
		This command changes the operational state of a secondary server to primary and advertises the information to the entire VTP domain. If the password is configured as hidden, the user is prompted to re-enter the password after this command.
		Before the device takes over the role of primary, it attempts to discover servers that conflict this information and follows another primary server. If conflicting servers are discovered, the user must reconfirm the takeover of operational state and the subsequent overwriting of configuration.
		• feature–Configures the device as primary server for a specific feature database. For example, the MST database. Possible values are MST and VLAN. By default, the VLAN database is chosen.

	Command or Action	Purpose
		Note This command is applicable for VTPv3 only.
Step 11	(Optional) switch# **show vtp status**	Displays information about the VTP configuration on the device, such as the version, mode, and revision number.
Step 12	(Optional) switch# **show vtp counters**	Displays information about VTP advertisement statistics on the device.
Step 13	(Optional) switch# **show vtp interface**	Displays the list of VTP-enabled interfaces.
Step 14	(Optional) switch# **show vtp password**	Displays the password for the management VTP domain.

Step 15	(Optional) switch# **show vtp devices [conflict]**
	Example:
	`Device# show vtp devices`
	
	`Gathering information from the domain, please wait.`
	`VTP Database Conf switch ID Primary Server Revision System Name lict`
	`------------ ---- -------------- ---------- ----------`
	`VLAN Yes 00b0.8e50.d000 000c.0412.6300 12354 main.cisco.com`
	`MST No 00b0.8e50.d000 0004.AB45.6000 24 main.cisco.com`
	`VLAN Yes 000c.0412.6300=000c.0412.6300 67 qwerty.cisco.com`
		This is a VTP version 3 command that displays information about neighbor switches. The information is not learned from the summary packet used for regular VTP packets. This command sends out a separate packet to collect information regarding neighbor switches running VTP version 3.

Step 16	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure VTP in transparent mode for the device:

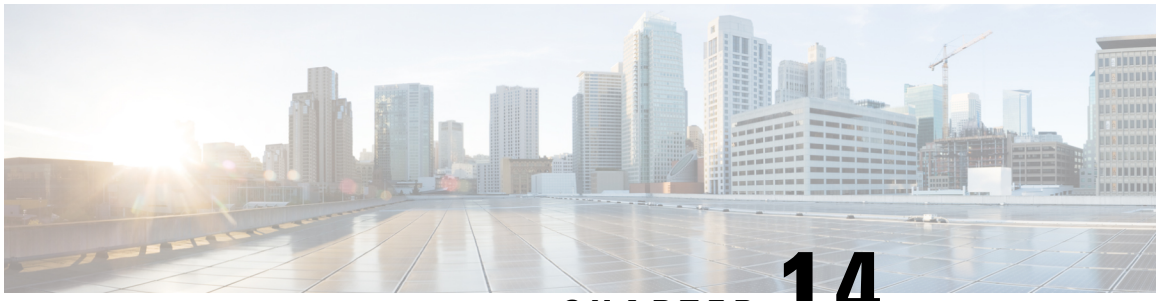
```
switch# configure terminal
switch(config)# feature vtp
switch(config)# vtp domain accounting
switch(config)# vtp version 2
switch(config)# vtp mode transparent
switch(config)# exit
switch#
```

Configuring VTP V3 Pruning

Follow the steps given below to configure VTP V3 Pruning.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vtp pruning**	Enables VTP pruning on the device. The default is disabled.
Step 3	(Optional) switch(config)# **no vtp pruning**	Disables VTP pruning on the device. The default is disabled.
Step 4	(Optional) switch(config)# **show interface** *interface-identifier* **switchport**	Displays the VTP pruning eligibility of the trunk port. The default is that all the VLANs from 2 to 1001 are pruning eligible.
Step 5	switch(config)# **interface port-channel** *channel-number*	Creates a port-channel interface and enter interface configuration mode.
Step 6	Required: switch(config-if)# **switchport trunk pruning vlan** [**add** | **remove** | **except** | **none** | **all**] *VLAN-IDs*	Sets the specified VLANs to be VTP pruning eligible.
Step 7	switch(config-if)# **end**	Returns to privileged EXEC mode.
Step 8	(Optional) switch# **show vtp counters**	Displays VTP pruning information and counters.
Step 9	(Optional) switch# **clear vtp counters**	Resets all the VTP pruning counter values.



CHAPTER 14

Configuring Traffic Storm Control

This chapter contains the following sections:

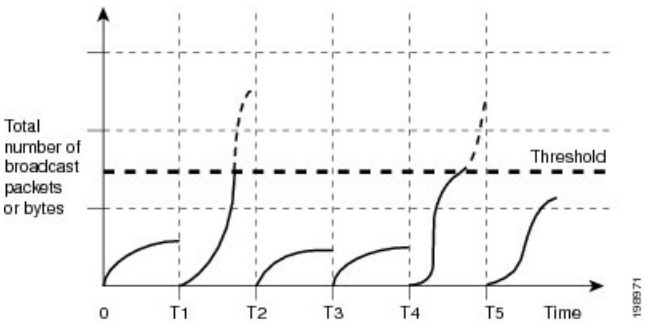
Information About Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. You can use the traffic storm control feature to prevent disruptions on Ethernet interfaces by a broadcast, multicast, or unknown unicast traffic storm.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming broadcast, multicast, or unknown unicast traffic over a 10-microsecond interval. During this interval, the traffic level, which is a percentage of the total available bandwidth of the port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends.

The following figure shows the broadcast traffic patterns on an Ethernet interface during a specified time interval. In this example, traffic storm control occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

Figure 21: Broadcast Suppression



The traffic storm control threshold numbers and the time interval allow the traffic storm control algorithm to work with different levels of packet granularity. For example, a higher threshold allows more packets to pass through.

Traffic storm control is implemented in the hardware. The traffic storm control circuitry monitors packets that pass from an Ethernet interface to the switching bus. Using the Individual/Group bit in the packet destination address, the circuitry determines if the packet is unicast or broadcast, tracks the current count of packets within the 10-microsecond interval, and filters out subsequent packets when a threshold is reached.

Traffic storm control uses a bandwidth-based method to measure traffic. You set the percentage of total available bandwidth that the controlled traffic can use. Because packets do not arrive at uniform intervals, the 10-microsecond interval can affect the operation of traffic storm control.

The following are examples of how traffic storm control operation is affected:

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding broadcast traffic until the end of the interval.
- If you enable multicast traffic storm control, and the multicast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding multicast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding broadcast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding multicast traffic until the end of the interval.

By default, Cisco NX-OS takes no corrective action when traffic exceeds the configured level.

Guidelines and Limitations for Traffic Storm Control

When configuring the traffic storm control level, follow these guidelines and limitations:

- You can configure traffic storm control on a port-channel interface or on any physical interface.
- Specify the level as a percentage of the total interface bandwidth:
  - The level can be from 0 to 100.
  - The optional fraction of a level can be from 0 to 99.
  - 100 percent means no traffic storm control.
  - 0.0 percent suppresses all traffic.
- There are local link and hardware limitations that prevent storm-control drops from being counted separately. Instead, storm-control drops are counted with other drops in the discards counter.
- Because of hardware limitations and the method by which packets of different sizes are counted, the level percentage is an approximation. Depending on the sizes of the frames that make up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points.

- Applying storm control over a HIF range is not recommended. The configuration might fail for one or more interfaces in the range depending on the hardware resource availability. The result of the command is partial success in some cases.
- In the Cisco Nexus 5000 switch, storm-control does not distinguish between IP, non-IP, registered, or unregistered multicast traffic. All multicast traffic is subject to a single-multicast storm control policer when configured.
- In the Cisco Nexus 5500 switch, storm-control is applied only to unregistered or unknown multicast MAC address.
- The link-level control protocols (LACP, LLDP and so on) are not affected in case of a traffic storm. The storm control is applied to data plane traffic only.
- The burst size values are:
  - For a 10G port, 48.68 Mbytes/390Mbits
  - For a 1G port, 25 Mbytes/200Mbits

Configuring Traffic Storm Control

You can set the percentage of total available bandwidth that the controlled traffic can use.

Note Traffic storm control uses a 10-microsecond interval that can affect the operation of traffic storm control.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {**ethernet** *slot*/*port* | **port-channel** *number*}	Enters interface configuration mode.
Step 3	switch(config-if)# **storm-control** [**broadcast** | **multicast**] **level** *percentage*[.*fraction*]	Configures traffic storm control for traffic on the interface. The default state is disabled.

Example

This example shows how to configure traffic storm control for port channels 122 and 123:

```
switch# configure terminal
switch(config)# interface port-channel 122, port-channel 123
switch(config-if-range)# storm-control multicast level 66.75
switch(config-if-range)# storm-control broadcast level 66.75
switch(config-if-range)#
```

Verifying the Traffic Storm Control Configuration

Use the following commands to display traffic storm control configuration information:

Command	Purpose
show interface [**ethernet** *slot*/*port* | **port-channel** *number*] **counters storm-control**	Displays the traffic storm control configuration for the interfaces. **Note** Traffic storm control uses a 10-microsecond interval that can affect the operation of traffic storm control.
show running-config interface	Displays the traffic storm control configuration.

Note When a storm event occurs on a port and the packets are dropped due to storm control configuration, a syslog message is generated to indicate that the storm event has started. An additional syslog message is generated when the storm event ends and the packet are no longer dropped.

Traffic Storm Control Example Configuration

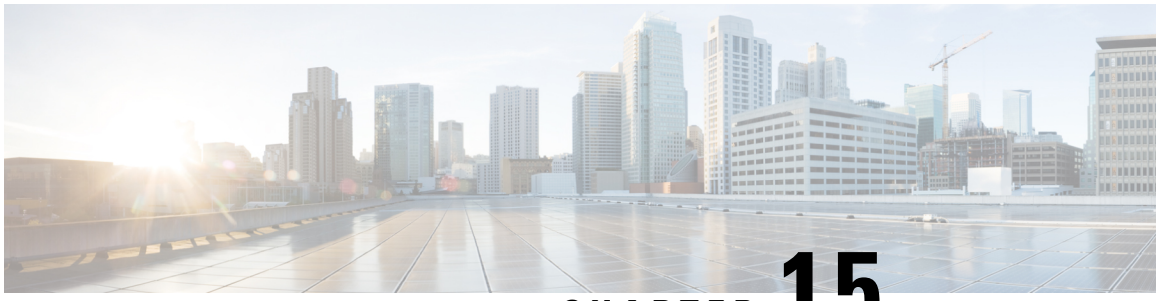
This example shows how to configure traffic storm control:

Default Settings for Traffic Storm Control

The following table lists the default settings for traffic storm control parameters.

Table 12: Default Traffic Storm Control Parameters

Parameters	Default
Traffic storm control	Disabled
Threshold percentage	100



CHAPTER 15

Configuring the Fabric Extender

This chapter contains the following sections:

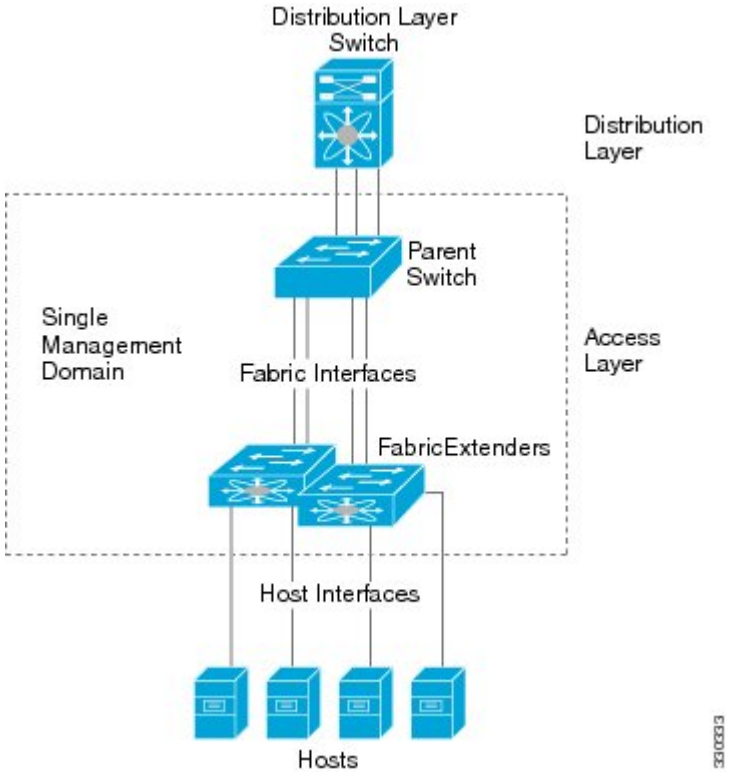
Information About the Cisco Nexus 2000 Series Fabric Extender

The Cisco Nexus 2000 Series Fabric Extender, also known as FEX, is a highly scalable and flexible server networking solution that works with Cisco Nexus Series devices to provide high-density, low-cost connectivity for server aggregation. Scaling across 1-Gigabit Ethernet, 10-Gigabit Ethernet, unified fabric, rack, and blade server environments, the Fabric Extender is designed to simplify data center architecture and operations.

The Fabric Extender integrates with its parent switch, which is a Cisco Nexus Series device, to allow automatic provisioning and configuration taken from the settings on the parent device. This integration allows large numbers of servers and hosts to be supported by using the same feature set as the parent device, including security and quality-of-service (QoS) configuration parameters, with a single management domain. The Fabric

Extender and its parent switch enable a large multipath, loop-free, active-active data center topology without the use of the Spanning Tree Protocol (STP).

Figure 22: Single Management Domain



The Cisco Nexus 2000 Series Fabric Extender forwards all traffic to its parent Cisco Nexus Series device over 10-Gigabit Ethernet fabric uplinks, which allows all traffic to be inspected by policies established on the Cisco Nexus Series device.

No software is included with the Fabric Extender. The software is automatically downloaded and upgraded from its parent device.

Fabric Extender Terminology

Some terms used in this document are as follows:

- Fabric interface—A 10-Gigabit Ethernet uplink port that is designated for connection from the Fabric Extender to its parent switch. A fabric interface cannot be used for any other purpose. It must be directly connected to the parent switch.

Note A fabric interface includes the corresponding interface on the parent switch. This interface is enabled when you enter the **switchport mode fex-fabric** command.

- Port channel fabric interface—A port channel uplink connection from the Fabric Extender to its parent switch. This connection consists of fabric interfaces that are bundled into a single logical channel.

- Host interface—An Ethernet host interface for connection to a server or host system.

Note Do not connect a bridge or switch to a host interface. These interfaces are designed to provide end host or server connectivity.

Note On Cisco Nexus 2348TQ and Nexus 2348UPQ FEX, if a port channel is used to connect a parent switch with a Fabric Extender device, the port channels can have maximum of 8 ports.

The Nexus 2348 FEX devices have a total of 6 * 40 Gigabit Ethernet uplink ports towards the parent switch. If these are used with native 40G uplinks port on a parent switch, then there is no limitation. All 6 ports can be used in either single homed or dual homed configuration. You can also use 40 Gigabit Ethernet uplink ports on the N2348 Fabric Extender device with 10 Gigabit Ethernet ports on the parent switch when used with the appropriate cabling. A maximum of 8 ports can be added to the port channel between the parent switch and Fabric Extender device. If it is a dual homed setup, VPC to the Fabric Extender device, only 4 ports per switch are allowed in the port channel.

- Port channel host interface—A port channel host interface for connection to a server or host system.

Fabric Extender Features

The Cisco Nexus 2000 Series Fabric Extender allows a single switch—and a single consistent set of switch features—to be supported across a large number of hosts and servers. By supporting a large server-domain under a single management entity, policies can be enforced more efficiently.

Some of the features of the parent switch cannot be extended onto the Fabric Extender.

Layer 2 Host Interfaces

Host Port Channel

The following fabric extenders support port channel host interface configurations. Up to eight interfaces can be combined in a port channel. The port channel can be configured with or without Link Aggregation Control Protocol (LACP).

- Cisco Nexus 2248TP
- Cisco Nexus 2348UPQ
- Cisco Nexus 2348TQ
- Cisco Nexus 2232PP
- Cisco Nexus 2332TQ

- Cisco Nexus 2224TP
- Cisco Nexus 2248PQ
- Cisco Nexus B22 Fabric Extender for Fujitsu (N2K-B22FTS-P)
- Cisco Nexus B22 Fabric Extender for Dell (N2K-B22DELL-P)
- Cisco Nexus B22 Fabric Extender for HP (N2K-B22HP-P)
- Cisco Nexus B22 Fabric Extender for IBM (N2K-B22IBM-P)

VLANs and Private VLANs

The Fabric Extender supports Layer 2 VLAN trunks and IEEE 802.1Q VLAN encapsulation. Host interfaces can be members of private VLANs with the following restrictions:

- You can configure a host interface as an isolated or community access port only.
- You cannot configure a host interface as a promiscuous port.
- You cannot configure a host interface as a private VLAN trunk port.

For more information about VLANs, see the chapter in this guide on Configuring VLANs.

Virtual Port Channels

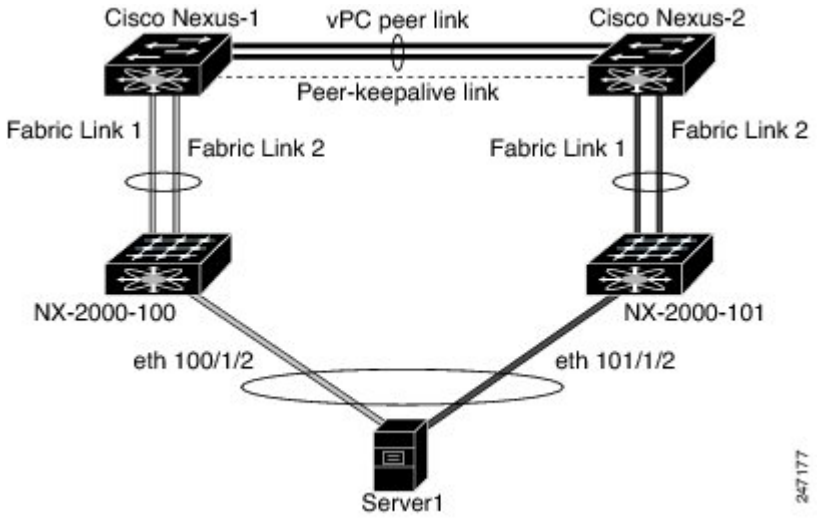
With a virtual port channel (vPC), you can configure topologies where a Cisco Nexus Fabric Extender is connected to a pair of parent switches or a pair of Fabric Extenders are connected to a single parent switch. The vPC can provide multipath connections, which allow you to create redundancy between the nodes on your network.

Note A port channel between two FEXs that are connected to the same Cisco Nexus device is not supported. Virtual port channels (vPCs) cannot span two different FEXs when connected to the same Cisco Nexus device.

The following vPC topologies are possible with the Fabric Extender:

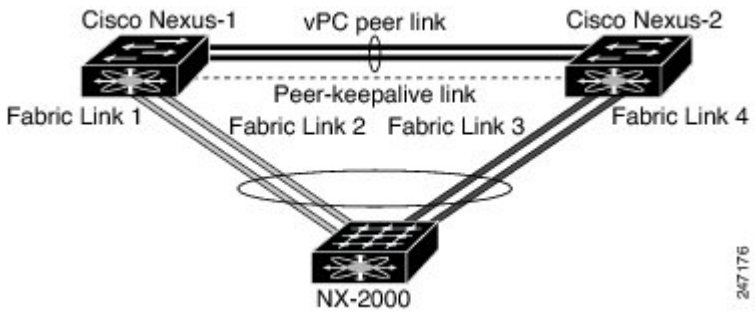
- The parent switches are connected single homed to Fabric Extenders that are subsequently connected to servers with dual interfaces (see the following figure).

Figure 23: Single Homed Fabric Extender vPC Topology



- The Fabric Extender is connected dual homed to two upstream parent switches and connected downstream to single homed servers (see the following figure).

Figure 24: Dual Homed Fabric Extender vPC Topology



This configuration is also called an Active-Active topology.

Note Port channels between two Fabric Extenders connected to the same Cisco Nexus device is not supported vPCs cannot span two different Fabric Extenders that are connected to the same physical Cisco Nexus device.

Fibre Channel over Ethernet Support

The Cisco Nexus 2232PP and Cisco Nexus 2248PQ support Fibre Channel over Ethernet (FCoE) with the following restrictions:

- Only FCoE Initialization Protocol (FIP) enabled converged network adapters (CNAs) are supported on the Fabric Extender.
- Binding to a port channel is limited to only one member in the port channel.

For configuration details, see the Fibre Channel over Ethernet Configuration Guide for the Nexus software release that you are using. The available versions of this document can be found at the following URL: http://www.cisco.com/en/US/products/ps9670/products_installation_and_configuration_guides_list.html.

Protocol Offload

To reduce the load on the control plane of the Cisco Nexus Series device, Cisco NX-OS allows you to offload link-level protocol processing to the Fabric Extender CPU. The following protocols are supported:

- Link Layer Discovery Protocol (LLDP)
- Cisco Discovery Protocol (CDP)
- Link Aggregation Control Protocol (LACP)

Quality of Service

Access Control Lists

The Fabric Extender supports the full range of ingress access control lists (ACLs) that are available on its parent Cisco Nexus Series device.

IGMP Snooping

Switched Port Analyzer

Fabric Interface Features

Oversubscription

Management Model

The Cisco Nexus 2000 Series Fabric Extender is managed by its parent switch over the fabric interfaces through a zero-touch configuration model. The switch discovers the Fabric Extender by detecting the fabric interfaces of the Fabric Extender.

After discovery, if the Fabric Extender has been correctly associated with the parent switch, the following operations are performed:

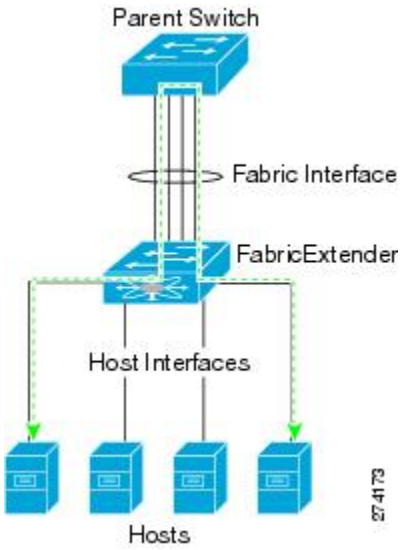
1. The switch checks the software image compatibility and upgrades the Fabric Extender if necessary.
2. The switch and Fabric Extender establish in-band IP connectivity with each other.
3. The switch pushes the configuration data to the Fabric Extender. The Fabric Extender does not store any configuration locally.

4. The Fabric Extender updates the switch with its operational status. All Fabric Extender information is displayed using the switch commands for monitoring and troubleshooting.

Forwarding Model

The Cisco Nexus 2000 Series Fabric Extender does not perform any local switching. All traffic is sent to the parent switch that provides central forwarding and policy enforcement, including host-to-host communications between two systems that are connected to the same Fabric Extender as shown in the following figure.

Figure 25: Forwarding Model



The forwarding model facilitates feature consistency between the Fabric Extender and its parent Cisco Nexus Series device.

Note The Fabric Extender provides end-host connectivity into the network fabric. As a result, BPDU Guard is enabled on all its host interfaces. If you connect a bridge or switch to a host interface, that interface is placed in an error-disabled state when a BPDU is received.

You cannot disable BPDU Guard on the host interfaces of the Fabric Extender.

The Fabric Extender supports egress multicast replication from the network to the host. Packets that are sent from the parent switch for multicast addresses attached to the Fabric Extender are replicated by the Fabric Extender ASICs and are then sent to corresponding hosts.

Connection Model

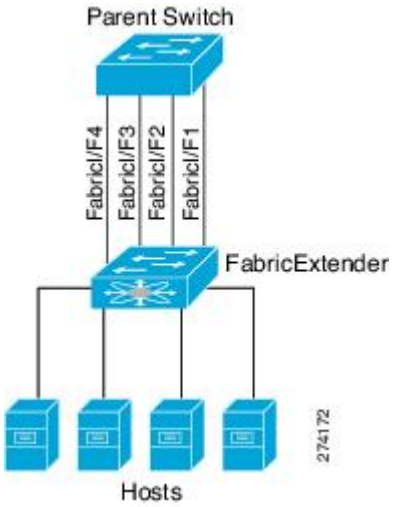
Two methods (the static pinning fabric interface connection and the Port Channel fabric interface connection) allow the traffic from an end host to the parent switch to be distributed when going through the Cisco Nexus 2000 Series Fabric Extender.

Note The Cisco Nexus 2248PQ Fabric Extender does not support the static pinning fabric interface connection.

Static Pinning Fabric Interface Connection

To provide a deterministic relationship between the host interfaces and the parent switch, you can configure the Fabric Extender to use individual fabric interface connections. This configuration connects the 10-Gigabit Ethernet fabric interfaces as shown in the following figure. You can use any number of fabric interfaces up to the maximum available on the model of the Fabric Extender.

Figure 26: Static Pinning Fabric Interface Connections



When the Fabric Extender is brought up, its host interfaces are distributed equally among the available fabric interfaces. As a result, the bandwidth that is dedicated to each end host toward the parent switch is never changed by the switch but instead is always specified by you.

Note If a fabric interface fails, all its associated host interfaces are brought down and remain down until the fabric interface is restored.

You must use the **pinning max-links** command to create a number of pinned fabric interface connections so that the parent switch can determine a distribution of host interfaces. The host interfaces are divided by the number of the max-links and distributed accordingly. The default value is max-links 1.

Caution Changing the value of the **max-links** is disruptive; all the host interfaces on the Fabric Extender are brought down and back up as the parent switch reassigns its static pinning.

The pinning order of the host interfaces is initially determined by the order in which the fabric interfaces were configured. When the parent switch is restarted, the configured fabric interfaces are pinned to the host interfaces in an ascending order by the port number of the fabric interface.

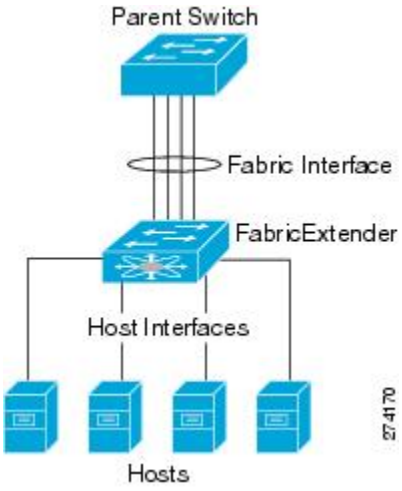
To guarantee a deterministic and sticky association across a reboot, you can manually redistribute the pinning.

Note The redistribution of the host interfaces will always be in an ascending order by the port number of the fabric interface.

Port Channel Fabric Interface Connection

To provide load balancing between the host interfaces and the parent switch, you can configure the Fabric Extender to use a port channel fabric interface connection. This connection bundles 10-Gigabit Ethernet fabric interfaces into a single logical channel as shown in the following figure.

Figure 27: Port Channel Fabric Interface Connection



When you configure the Fabric Extender to use a port channel fabric interface connection to its parent switch, the switch load balances the traffic from the hosts that are connected to the host interface ports by using the following load-balancing criteria to select the link:

- For a Layer 2 frame, the switch uses the source and destination MAC addresses.
- For a Layer 3 frame, the switch uses the source and destination MAC addresses and the source and destination IP addresses.

Note A fabric interface that fails in the port channel does not trigger a change to the host interfaces. Traffic is automatically redistributed across the remaining links in the port channel fabric interface. If all links in the fabric port channel go down, all host interfaces on the FEX are set to the down state.

Port Numbering Convention

Fabric Extender Image Management

No software ships with the Cisco Nexus 2000 Series Fabric Extender. The Fabric Extender image is bundled into the system image of the parent switch. The image is automatically verified and updated (if required) during the association process between the parent switch and the Fabric Extender.

When you enter the **install all** command, it upgrades the software on the parent Cisco Nexus Series switch and also upgrades the software on any attached Fabric Extender. To minimize downtime as much as possible, the Fabric Extender remains online while the installation process loads its new software image. Once the software image has successfully loaded, the parent switch and the Fabric Extender both automatically reboot.

This process is required to maintain version compatibility between the parent switch and the Fabric Extender.

Fabric Extender Hardware

The Cisco Nexus 2000 Series Fabric Extender architecture allows hardware configurations with various host interface counts and speeds.

Chassis

The Cisco Nexus 2000 Series Fabric Extender is a 1 RU chassis that is designed for rack mounting. The chassis supports redundant hot-swappable fans and power supplies.

Ethernet Interfaces

There are 8 models of the Cisco Nexus 2000 Series Fabric Extender:

- The Cisco Nexus 2148T has 48 1000BASE-T Ethernet host interfaces for its downlink connection to servers or hosts and 4 10-Gigabit Ethernet fabric interfaces with SFP+ interface adapters for its uplink connection to the parent switch.
- The Cisco Nexus 2224TP has 24 100BASE-T/1000Base-T Ethernet host interfaces for its downlink connection to servers or hosts and 2 10-Gigabit Ethernet fabric interfaces with SFP+ interface adapters for its uplink connection to the parent switch.
- The Cisco Nexus 2248PQ has 48 10-Gigabit Ethernet host interfaces with SFP+ interface adapters and 16 10-Gigabit Ethernet fabric interfaces corresponding to 4 QSFP interface adapters for its uplink connection to the parent switch.
- The Cisco Nexus 2232PP has 32 10-Gigabit Ethernet host interfaces with SFP+ interface adapters and 8 10-Gigabit Ethernet fabric interfaces with SFP+ interface adapters for its uplink connection to the parent switch.
- Cisco Nexus 2348UPQ—FEX for QSA (FET-10G, SFP-10G-SR, SFP-10G-ER). From Cisco NX-OS Release 7.3(2)N1(1), Cisco Nexus 2348UPQ 10GE Fabric Extender for ports using 1G Cu SFP GLC-T supports the **speed 100** command.

Note From Cisco NX-OS Release 7.3(2)N1(1), Cisco Nexus 2248PQ 10GE Fabric Extender, Cisco Nexus 2232PP 10GE Fabric Extender, and Cisco Nexus 2348UPQ 10GE Fabric Extender with 1G-based SFP support the **no negotiate auto** command.

- Cisco Nexus N2332TQ—FEX supporting 32 10GBaseT host ports and 4 QSFP+ network ports.
- Cisco Nexus 2348TQ FEX (N2K-C2348TQ-10GE)
- The Cisco Nexus 2248TP has 48 100BASE-T/1000Base-T Ethernet host interfaces for its downlink connection to servers or hosts and 4 10-Gigabit Ethernet fabric interfaces with SFP+ interface adapters for its uplink connection to the parent switch.

  The Cisco Nexus 2248TP-E has all the features of the Cisco Nexus 2248TP with these additional features:

  - A larger buffer to absorb large bursts.
  - Support for an ingress and egress queue-limit per port.
  - Support for debug counters.
  - Support for pause no-drop behavior over a cable distance of 3000 meters between the Fabric Extender and switch.
  - Support for a user configurable shared-buffer.
  - Support for the **speed auto 100** command from Cisco NX-OS Release 7.3(2)N1(1).

- The Cisco Nexus B22 Fabric Extender for HP (NB22HP) has 16 1G/10-Gigabit Ethernet host interfaces. All host interfaces use all of the available fabric interfaces.
- The Cisco Nexus B22 Fabric Extender for Fujitsu (NB22FTS) has 16 10-Gigabit Ethernet host interfaces. All host interfaces use all of the available fabric interfaces.
- The Cisco Nexus B22 Fabric Extender for Dell (NB22DELL) has 16 1G/10-Gigabit Ethernet host interfaces. All host interfaces use all of the available fabric interfaces.
- The Cisco Nexus B22 Fabric Extender for IBM (NB22IBM) has 14 1G/10-Gigabit Ethernet host interfaces. All host interfaces use all of the available fabric interfaces.

Speed and Duplex Mode

The table listed below shows the autonegotiation matrix for N2K-C2348TQ-10GE and N2K-C2332TQ-10GE fabric extenders.

Configuration N2K-C2348TQ-10GE and N2K-C2332TQ-10GE HIF (Speed/Duplex	Configuration NIC (Speed/Duplex)	Resulting HIF status (Speed/Duplex)	Resulting NIC status (Speed/Duplex)	Comments
Table for 100 Mbps				

Configuration N2K-C2348TQ-10GE and N2K-C2332TQ-10GE HIF (Speed/Duplex	Configuration NIC (Speed/Duplex)	Resulting HIF status (Speed/Duplex)	Resulting NIC status (Speed/Duplex)	Comments
AUTO	AUTO	UP, 100 Mbps, Full duplex	UP, 100 Mbps, Full duplex	Assuming maximum capability of NIC is 100 Mbps, Full duplex
100 Mbps, Full duplex (force mode)	100 Mbps, Full duplex	UP, 100 Mbps, Full duplex	UP, 100 Mbps, Full duplex	Link DOWN in some scenarios. [2]
100 Mbps, Full duplex (force mode)	100 Mbps, Half duplex	UP, 100 Mbps, Full duplex	UP, 100 Mbps, Half duplex	Duplex mismatch as per standard; results in collision errors. Hence, not a functional scenario.
100 Mbps, Full duplex (force mode)	AUTO	UP, 100 Mbps, Full duplex	UP, 100 Mbps, Half Duplex	Duplex mismatch as per standard; results in collision errors. Hence, not a functional scenario. (Link will be DOWN if NIC is not 100 Mbps, Half Duplex capable)
AUTO	100 Mbps, Full duplex	DOWN	DOWN	—
AUTO	100 Mbps, Half duplex	DOWN	DOWN	—
Table for 1G (1000 Mbps)				
AUTO	AUTO	UP, 1000 Mbps, Full duplex	UP, 1000 Mbps, Full duplex	Assuming maximum capability of NIC is 1000 Mbps, Full duplex
1000 Mbps, Full duplex	AUTO	UP, 1000 Mbps, Full duplex	UP, 1000 Mbps, Full duplex	Assuming maximum capability of NIC is 1000 Mbps, Full duplex
AUTO	1000 Mbps, Full duplex	UP, 1000 Mbps, Full duplex	UP, 1000 Mbps, Full duplex	—

Configuration N2K-C2348TQ-10GE and N2K-C2332TQ-10GE HIF (Speed/Duplex	Configuration NIC (Speed/Duplex)	Resulting HIF status (Speed/Duplex)	Resulting NIC status (Speed/Duplex)	Comments
1000 Mbps, Full duplex	1000 Mbps, Full duplex	UP, 1000 Mbps, Full duplex	UP, 1000 Mbps, Full duplex	—
100 Mbps, Full duplex	1000 Mbps, Full duplex	DOWN	DOWN	Speed Mismatch
1000 Mbps, Full duplex	100 Mbps, Full duplex	DOWN	DOWN	Speed Mismatch
1000 Mbps, Full duplex	100 Mbps, Half duplex	DOWN	DOWN	Speed Mismatch
10000 Mbps, Full duplex	1000 Mbps, Full duplex	DOWN	DOWN	Speed Mismatch
10000 Mbps, Full duplex	100 Mbps, Full duplex	DOWN	DOWN	Speed Mismatch
10000 Mbps, Full duplex	100 Mbps, Half duplex	DOWN	DOWN	Speed Mismatch
Table for 10G (10000 Mbps)				
AUTO	AUTO	UP, 10000 Mbps, Full duplex	UP, 10000 Mbps, Full duplex	Assuming maximum capability of NIC is 10000 Mbps, Full duplex
10000 Mbps, Full duplex	AUTO	UP, 10000 Mbps, Full duplex	UP, 10000 Mbps, Full duplex	Assuming maximum capability of NIC is 10000 Mbps, Full duplex
AUTO	10000 Mbps, Full duplex	UP, 10000 Mbps, Full duplex	UP, 10000 Mbps, Full duplex	—
10000 Mbps, Full duplex	10000 Mbps, Full duplex	UP, 10000 Mbps, Full duplex	UP, 10000 Mbps, Full duplex	—
100 Mbps, Full duplex	10000 Mbps, Full duplex	DOWN	DOWN	Speed Mismatch
1000 Mbps, Full duplex	10000 Mbps, Full duplex	DOWN	DOWN	Speed Mismatch

[2] Refer to CSCut35369 for more details.

Example: Configuring the Interface Speed Parameters

Configuration: AUTO

Configuring speed as AUTO (advertises all speeds and Full Duplex only)

```
switch(config)# interface ethernet 101/1/1
switch(config-if)# speed auto
```

Configuration: AUTO 100

Configuring speed as AUTO 100 Mbps (autonegotiates the advertised speed to 100 Mbps)

```
switch(config)# interface ethernet 101/1/1
switch(config-if)# speed auto 100
```

Configuration: 100 (Force Mode)

Configuring speed as 100 Mbps

```
switch(config)# interface ethernet 101/1/1
switch(config-if)# speed 100
```

Configuration: 1G

Configuring speed as 1000 Mbps (This has autoneg enabled with 1000 Mbps)

```
switch(config)# interface ethernet 101/1/1
switch(config-if)# speed 1000
```

Configuration: 10G

Configuring speed as 10000 Mbps (This has autoneg enabled with 10000 Mbps)

```
switch(config)# interface ethernet 101/1/1
switch(config-if)# speed 10000
```

Disabling Autonegotiation

The **negotiate auto** command enables the autonegotiation protocol to configure the speed, duplex, and automatic flow-control of the Gigabit Ethernet interface. Use the **no negotiate auto** command to disable autonegotiation. The **no negotiate auto** command is supported only in Cisco Nexus 2248PQ 10GE Fabric Extender, Cisco Nexus 2232PP 10GE Fabric Extender, and Cisco Nexus 2348UPQ 10GE Fabric Extender with 1G-based SFP.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Specify an Ethernet interface to configure:

switch(config)# **interface ethernet** *slot/port*

Step 3 Disable the autonegotiation:

switch(config-if)# **no negotiate auto**

Step 4 Exit interface and global configuration modes:

switch(config-if)# **exit**

switch(config)# **exit**

Disabling Autonegotiation

This example shows a running configuration to disable autonegotiation. Replace the *placeholders* with relevant values for your setup.

```
configure terminal
 interface ethernet <1>/<40>
  no negotiate auto
  exit
 exit
```

Associating a Fabric Extender to a Fabric Interface

Associating a Fabric Extender to an Ethernet Interface

Before you begin

Ensure that you have enabled the Fabric Extender feature.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 1/40`
	`switch(config)#`
		Specifies an Ethernet interface to configure.

	Command or Action	Purpose

Step 3	**switchport mode fex-fabric**
	Example:
	`switch(config-if)# switchport mode fex-fabric`
	`switch(config-if)#`
		Sets the interface to support an external Fabric Extender.

Step 4	**fex associate** *FEX-number*
	Example:
	`switch(config-if)# fex associate 101`
	`switch#`
		Associates the FEX number to the Fabric Extender unit attached to the interface. The range of the FEX number is from 100 to 199.

Step 5	(Optional) **show interface ethernet** *port*/*slot* **fex-intf**
	Example:
	`switch# show interface ethernet 1/40 fex-intf`
	`switch#`
		Displays the association of a Fabric Extender to an Ethernet interface.

Example

This example shows how to associate the Fabric Extender to an Ethernet interface on the parent device:

```
switch# configure terminal
switch(config)# interface ethernet 1/40
switch(config-if)# switchport mode fex-fabric
switch(config-if)# fex associate 101
switch(config)#
```

This example shows how to display the association of the Fabric Extender and the parent device:

```
switch# show interface ethernet 1/40 fex-intf
Fabric           FEX
Interface        Interfaces
---------------------------------------------------
Eth1/40          Eth101/1/48   Eth101/1/47   Eth101/1/46   Eth101/1/45
                 Eth101/1/44   Eth101/1/43   Eth101/1/42   Eth101/1/41
                 Eth101/1/40   Eth101/1/39   Eth101/1/38   Eth101/1/37
                 Eth101/1/36   Eth101/1/35   Eth101/1/34   Eth101/1/33
                 Eth101/1/32   Eth101/1/31   Eth101/1/30   Eth101/1/29
                 Eth101/1/28   Eth101/1/27   Eth101/1/26   Eth101/1/25
                 Eth101/1/24   Eth101/1/23   Eth101/1/22   Eth101/1/21
                 Eth101/1/20   Eth101/1/19   Eth101/1/18   Eth101/1/17
                 Eth101/1/16   Eth101/1/15   Eth101/1/14   Eth101/1/13
                 Eth101/1/12   Eth101/1/11   Eth101/1/10   Eth101/1/9
                 Eth101/1/8    Eth101/1/7    Eth101/1/6    Eth101/1/5
                 Eth101/1/4    Eth101/1/3    Eth101/1/2    Eth101/1/1
```

Associating a Fabric Extender to a Port Channel

Before you begin

Ensure that you have enabled the Fabric Extender feature.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface port-channel** *channel*
	Example:
	`switch(config)# interface port-channel 4`
	`switch(config-if)#`
		Specifies a port channel to configure.

Step 3	**switchport mode fex-fabric** **Example:** `switch(config-if)# switchport mode fex-fabric`	Sets the port channel to support an external Fabric Extender.
Step 4	**fex associate** *FEX-number* **Example:** `switch(config-if)# fex associate 101`	Associates a FEX number to the Fabric Extender unit attached to the interface. The range is from 100 to 199.
Step 5	(Optional) **show interface port-channel** *channel* **fex-intf** **Example:** `switch# show interface port-channel 4 fex-intf`	Displays the association of a Fabric Extender to a port channel interface.

Example

This example shows how to associate the Fabric Extender to a port channel interface on the parent device:

```
switch# configure terminal
switch(config)# interface ethernet 1/28
switch(config-if)# channel-group 4
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface ethernet 1/29
switch(config-if)# channel-group 4
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface ethernet 1/30
```

```
switch(config-if)# channel-group 4
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface ethernet 1/31
switch(config-if)# channel-group 4
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface port-channel 4
switch(config-if)# switchport
switch(config-if)# switchport mode fex-fabric
switch(config-if)# fex associate 101
```

Tip As a best practice, only enter the **fex associate** command from the port channel interface, not from the physical interface.

Note When adding physical interfaces to port channels, all configurations on the port channel and physical interface must match.

This example shows how to display the association of the Fabric Extender and the parent device:

```
switch# show interface port-channel 4 fex-intf
Fabric           FEX
Interface        Interfaces
---------------------------------------------------
Po4              Eth101/1/48   Eth101/1/47   Eth101/1/46   Eth101/1/45
                 Eth101/1/44   Eth101/1/43   Eth101/1/42   Eth101/1/41
                 Eth101/1/40   Eth101/1/39   Eth101/1/38   Eth101/1/37
                 Eth101/1/36   Eth101/1/35   Eth101/1/34   Eth101/1/33
                 Eth101/1/32   Eth101/1/31   Eth101/1/30   Eth101/1/29
                 Eth101/1/28   Eth101/1/27   Eth101/1/26   Eth101/1/25
                 Eth101/1/24   Eth101/1/23   Eth101/1/22   Eth101/1/21
                 Eth101/1/20   Eth101/1/19   Eth101/1/18   Eth101/1/17
                 Eth101/1/16   Eth101/1/15   Eth101/1/14   Eth101/1/13
                 Eth101/1/12   Eth101/1/11   Eth101/1/10   Eth101/1/9
                 Eth101/1/8    Eth101/1/7    Eth101/1/6    Eth101/1/5
                 Eth101/1/4    Eth101/1/3    Eth101/1/2    Eth101/1/1
```

Disassociating a Fabric Extender from an Interface

Before you begin

Ensure that you have enabled the Fabric Extender feature.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**interface** {**ethernet** *slot*/*port* | **port-channel** *channel*}
	Example:
	`switch(config)# interface port-channel 4`
	`switch(config-if)#`
		Specifies the interface to configure. The interface can be an Ethernet interface or a port channel.

Step 3	**no fex associate** **Example:** `switch(config-if)# no fex associate`	Disassociates the Fabric Extender unit attached to the interface.

Configuring Fabric Extender Global Features

You can configure global features on the Fabric Extender.

Before you begin

Ensure that you have enabled the Fabric Extender feature set.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**fex** *FEX-number*
	Example:
	`switch(config)# fex 101`
	`switch(config-fex)#`
		Enters FEX configuration mode for the specified Fabric Extender. The range of the *FEX-number* is from 100 to 199.

Step 3	(Optional) **description** *desc* **Example:** `switch(config-fex)# description Rack7A-N2K`	Specifies the description. The default is the string FEX*xxxx* where *xxxx* is the FEX number. If the FEX number is 123, the description is FEX0123.
Step 4	(Optional) **no description** **Example:** `switch(config-fex)# no description`	Deletes the description.
Step 5	(Optional) **no type** **Example:** `switch(config-fex)# no type`	Deletes the FEX type. When a Fabric Extender is connected to the fabric interfaces and does not match the configured type that is saved in the binary configuration on the parent switch,

	Command or Action	Purpose
		all configurations for all interfaces on the Fabric Extender are deleted.
Step 6	(Optional) **pinning max-links** *uplinks* **Example:** `switch(config-fex)# pinning max-links 2`	Defines the number of uplinks. The default is 1. The range is from 1 to 4. This command is only applicable if the Fabric Extender is connected to its parent switch using one or more statically pinned fabric interfaces. There can only be one port channel connection. **Caution** Changing the number of uplinks with the **pinning max-links** command disrupts all the host interface ports of the Fabric Extender.
Step 7	(Optional) **no pinning max-links** **Example:** `switch(config-fex)# no pinning max-links`	Resets the number of uplinks to the default. **Caution** Changing the number of uplinks with the **no pinning max-links** command disrupts all the host interface ports of the Fabric Extender.
Step 8	(Optional) **serial** *serial* **Example:** `switch(config-fex)# serial JAF1339BDSK`	Defines a serial number string. If this command is configured, a switch allows the corresponding chassis ID to associate (using the **fex associate** command) only if the Fabric Extender reports a matching serial number string. **Caution** Configuring a serial number that does not match the specified Fabric Extender forces the Fabric Extender offline.
Step 9	(Optional) **no serial** **Example:** `switch(config-fex)# no serial`	Deletes the serial number string.

Enabling the Fabric Extender Locator LED

The locator beacon LED on the Fabric Extender allows you to locate a specific Fabric Extender in a rack.

Procedure

	Command or Action	Purpose
Step 1	**locator-led fex** *FEX-number* **Example:**	Turns on the locator beacon LED for a specific Fabric Extender.

	Command or Action	Purpose
	`switch# locator-led fex 101`	
Step 2	(Optional) **no locator-led fex** *FEX-number* **Example:** `switch# no locator-led fex 101`	Turns off the locator beacon LED for a specific Fabric Extender.

Redistributing the Links

When you provision the Fabric Extender with statically pinned interfaces, the downlink host interfaces on the Fabric Extender are pinned to the fabric interfaces in the order that they were initially configured. If you want to maintain a specific relationship of host interfaces to fabric interface across reboots, you should repin the links.

You may want to perform this function in these two situations:

- A change in the max-links configuration.
- If you need to maintain the pinning order of host interfaces to fabric interfaces.

Note The Cisco Nexus 2248PQ Fabric Extender does not support the static pinning fabric interface connection.

Changing the Number of Links

If you initially configured a specific port on the parent switch, for example port 33, as your only fabric interface, all 48 host interfaces are pinned to this port. If you provision another port, for example 35, you must enter the **pinning max-links 2** command to redistribute the host interfaces. All host interfaces are brought down and host interfaces 1 to 24 are pinned to fabric interface 33 and host interfaces 25 to 48 are pinned to fabric interface 35.

Maintaining the Pinning Order

The pinning order of the host interfaces is initially determined by the order in which the fabric interfaces were configured. In this example, four fabric interfaces were configured in the following order:

```
switch# show interface ethernet 1/35 fex-intf
Fabric           FEX
Interface        Interfaces
---------------------------------------------------
 Eth1/35         Eth101/1/12   Eth101/1/11   Eth101/1/10   Eth101/1/9
                 Eth101/1/8    Eth101/1/7    Eth101/1/6    Eth101/1/5
                 Eth101/1/4    Eth101/1/3    Eth101/1/2    Eth101/1/1

switch# show interface ethernet 1/33 fex-intf
Fabric           FEX
Interface        Interfaces
---------------------------------------------------
 Eth1/33         Eth101/1/24   Eth101/1/23   Eth101/1/22   Eth101/1/21
                 Eth101/1/20   Eth101/1/19   Eth101/1/18   Eth101/1/17
```

```
                Eth101/1/16   Eth101/1/15   Eth101/1/14   Eth101/1/13

switch# show interface ethernet 1/38 fex-intf
Fabric          FEX
Interface       Interfaces
---------------------------------------------------
 Eth1/38        Eth101/1/36   Eth101/1/35   Eth101/1/34   Eth101/1/33
                Eth101/1/32   Eth101/1/31   Eth101/1/30   Eth101/1/29
                Eth101/1/28   Eth101/1/27   Eth101/1/26   Eth101/1/25

switch# show interface ethernet 1/40 fex-intf
Fabric          FEX
Interface       Interfaces
---------------------------------------------------
 Eth1/40        Eth101/1/48   Eth101/1/47   Eth101/1/46   Eth101/1/45
                Eth101/1/44   Eth101/1/43   Eth101/1/42   Eth101/1/41
                Eth101/1/40   Eth101/1/39   Eth101/1/38   Eth101/1/37
```

The next time that you reboot the Fabric Extender, the configured fabric interfaces are pinned to the host interfaces in an ascending order by port number of the fabric interface. If you want to configure the same fixed distribution of host interfaces without restarting the Fabric Extender, enter the **fex pinning redistribute** command.

Note It is a misconfiguration to have more fabric ports than pinning number even if the extra port is in DOWN state.

Redistributing Host Interfaces

Caution This command disrupts all the host interface ports of the Fabric Extender.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**fex pinning redistribute** *FEX-number*
	Example:
	`switch(config) # fex pinning redistribute 101`
	`switch(config) #`
		Redistributes the host connections. The range of the *FEX-number* is from 100 to 199.

Verifying the Fabric Extender Configuration

Use the following commands to display configuration information about the defined interfaces on a Fabric Extender:

Command or Action	Purpose
show fex [*FEX-number*] [**detail**]	Displays information about a specific Fabric Extender or all attached units.
show interface *type number* **fex-intf**	Displays the Fabric Extender ports that are pinned to a specific switch interface.
show interface fex-fabric	Displays the switch interfaces that have detected a Fabric Extender uplink.
show interface ethernet *number* **transceiver** [**fex-fabric**]	Displays the SFP+ transceiver and diagnostic optical monitoring (DOM) information for the Fabric Extender uplinks.
show feature-set	Displays the status of the feature sets on the device.

Configuration Examples for the Fabric Extender

This example shows how to display all the attached Fabric Extender units:

```
switch# show fex
  FEX         FEX           FEX                      FEX
Number    Description      State            Model            Serial
------------------------------------------------------------------------
100    FEX0100               Online     N2K-C2248TP-1GE   JAF1339BDSK
101    FEX0101               Online     N2K-C2232P-10GE   JAF1333ADDD
102    FEX0102               Online     N2K-C2232P-10GE   JAS12334ABC
```

This example shows how to display the detailed status of a specific Fabric Extender:

```
switch# show fex 100 detail
FEX: 100 Description: FEX0100   state: Online
  FEX version: 5.0(2)N1(1) [Switch version: 5.0(2)N1(1)]
  FEX Interim version: 5.0(2)N1(0.205)
  Switch Interim version: 5.0(2)N1(0.205)
  Extender Model: N2K-C2224TP-1GE,  Extender Serial: JAF1427BQLG
  Part No: 73-13373-01
  Card Id: 132, Mac Addr: 68:ef:bd:62:2a:42, Num Macs: 64
  Module Sw Gen: 21  [Switch Sw Gen: 21]
  post level: complete
 pinning-mode: static    Max-links: 1
  Fabric port for control traffic: Eth1/29
  Fabric interface state:
    Po100 - Interface Up. State: Active
    Eth1/29 - Interface Up. State: Active
    Eth1/30 - Interface Up. State: Active
  Fex Port        State  Fabric Port  Primary Fabric
       Eth100/1/1    Up       Po100       Po100
       Eth100/1/2    Up       Po100       Po100
       Eth100/1/3    Up       Po100       Po100
```

```
      Eth100/1/4    Up       Po100       Po100
      Eth100/1/5    Up       Po100       Po100
      Eth100/1/6    Up       Po100       Po100
      Eth100/1/7    Up       Po100       Po100
      Eth100/1/8    Up       Po100       Po100
      Eth100/1/9    Up       Po100       Po100
     Eth100/1/10    Up       Po100       Po100
     Eth100/1/11    Up       Po100       Po100
     Eth100/1/12    Up       Po100       Po100
     Eth100/1/13    Up       Po100       Po100
     Eth100/1/14    Up       Po100       Po100
     Eth100/1/15    Up       Po100       Po100
     Eth100/1/16    Up       Po100       Po100
     Eth100/1/17    Up       Po100       Po100
     Eth100/1/18    Up       Po100       Po100
     Eth100/1/19    Up       Po100       Po100
     Eth100/1/20    Up       Po100       Po100
     Eth100/1/21    Up       Po100       Po100
     Eth100/1/22    Up       Po100       Po100
     Eth100/1/23    Up       Po100       Po100
     Eth100/1/24    Up       Po100       Po100
     Eth100/1/25    Up       Po100       Po100
     Eth100/1/26    Up       Po100       Po100
     Eth100/1/27    Up       Po100       Po100
     Eth100/1/28    Up       Po100       Po100
     Eth100/1/29    Up       Po100       Po100
     Eth100/1/30    Up       Po100       Po100
     Eth100/1/31    Up       Po100       Po100
     Eth100/1/32    Up       Po100       Po100
     Eth100/1/33    Up       Po100       Po100
     Eth100/1/34    Up       Po100       Po100
     Eth100/1/35    Up       Po100       Po100
     Eth100/1/36    Up       Po100       Po100
     Eth100/1/37    Up       Po100       Po100
     Eth100/1/38    Up       Po100       Po100
     Eth100/1/39    Up       Po100       Po100
     Eth100/1/40  Down       Po100       Po100
     Eth100/1/41    Up       Po100       Po100
     Eth100/1/42    Up       Po100       Po100
     Eth100/1/43    Up       Po100       Po100
     Eth100/1/44    Up       Po100       Po100
     Eth100/1/45    Up       Po100       Po100
     Eth100/1/46    Up       Po100       Po100
     Eth100/1/47    Up       Po100       Po100
     Eth100/1/48    Up       Po100       Po100
Logs:
02/05/2010 20:12:17.764153: Module register received
02/05/2010 20:12:17.765408: Registration response sent
02/05/2010 20:12:17.845853: Module Online Sequence
02/05/2010 20:12:23.447218: Module Online
```

This example shows how to display the Fabric Extender interfaces pinned to a specific switch interface:

```
switch# show interface port-channel 100 fex-intf
Fabric           FEX
Interface        Interfaces
---------------------------------------------------
 Po100           Eth100/1/48   Eth100/1/47   Eth100/1/46   Eth100/1/45
                 Eth100/1/44   Eth100/1/43   Eth100/1/42   Eth100/1/41
                 Eth100/1/40   Eth100/1/39   Eth100/1/38   Eth100/1/37
                 Eth100/1/36   Eth100/1/35   Eth100/1/34   Eth100/1/33
                 Eth100/1/32   Eth100/1/31   Eth100/1/30   Eth100/1/29
                 Eth100/1/28   Eth100/1/27   Eth100/1/26   Eth100/1/25
                 Eth100/1/24   Eth100/1/22   Eth100/1/20   Eth100/1/19
```

```
Eth100/1/18   Eth100/1/17   Eth100/1/16   Eth100/1/15
Eth100/1/14   Eth100/1/13   Eth100/1/12   Eth100/1/11
Eth100/1/10   Eth100/1/9    Eth100/1/8    Eth100/1/7
Eth100/1/6    Eth100/1/5    Eth100/1/4    Eth100/1/3
Eth100/1/2    Eth100/1/1
```

This example shows how to display the switch interfaces that are connected to a Fabric Extender uplink:

```
switch# show interface fex-fabric
     Fabric      Fabric       Fex                FEX
Fex  Port      Port State    Uplink    Model          Serial
--------------------------------------------------------------
100   Eth1/29        Active     3    N2K-C2248TP-1GE  JAF1339BDSK
100   Eth1/30        Active     4    N2K-C2248TP-1GE  JAF1339BDSK
102   Eth1/33        Active     1    N2K-C2232P-10GE  JAS12334ABC
102   Eth1/34        Active     2    N2K-C2232P-10GE  JAS12334ABC
102   Eth1/35        Active     3    N2K-C2232P-10GE  JAS12334ABC
102   Eth1/36        Active     4    N2K-C2232P-10GE  JAS12334ABC
101   Eth1/37        Active     5    N2K-C2232P-10GE  JAF1333ADDD
101   Eth1/38        Active     6    N2K-C2232P-10GE  JAF1333ADDD
101   Eth1/39        Active     7    N2K-C2232P-10GE  JAF1333ADDD
101   Eth1/40        Active     8    N2K-C2232P-10GE  JAF1333ADDD
```

This example shows how to display the SFP+ transceiver and diagnostic optical monitoring (DOM) information for Fabric Extender uplinks for an SFP+ transceiver that is plugged into the parent switch interface:

```
switch# show interface ethernet 1/40 transceiver
Ethernet1/40
    sfp is present
    name is CISCO-MOLEX INC
    part number is 74752-9026
    revision is A0
    serial number is MOC13321057
    nominal bitrate is 12000 MBits/sec
    Link length supported for copper is 3 m(s)
    cisco id is --
    cisco extended id number is 4
```

This example shows how to display the SFP+ transceiver and DOM information for Fabric Extender uplinks for an SFP+ transceiver that is plugged into the uplink port on the Fabric Extender:

```
switch# show interface ethernet 1/40 transceiver fex-fabric
Ethernet1/40
    sfp is present
    name is CISCO-MOLEX INC
    part number is 74752-9026
    revision is A0
    serial number is MOC13321057
    nominal bitrate is 12000 MBits/sec
    Link length supported for 50/125mm fiber is 0 m(s)
    Link length supported for 62.5/125mm fiber is 0 m(s)
    cisco id is --
    cisco extended id number is 4
```

Verifying the Chassis Management Information

Use the following to display configuration information used on the switch supervisor to manage the Fabric Extender.

Command or Action	Purpose
show diagnostic result fex *FEX-number*	Displays results from the diagnostic test for a Fabric Extender.
show environment fex {**all** | *FEX-number*} [**temperature** | **power** | **fan**]	Displays the environmental sensor status.
show inventory fex *FEX-number*	Displays inventory information for a Fabric Extender.
show module fex [*FEX-number*]	Displays module information about a Fabric Extender.
show sprom fex *FEX-number* {**all** | **backplane** | **powersupply** *ps-num*} | **all**	Displays the contents of the serial PROM (SPROM) on the Fabric Extender. The unit of the power for the **show sprom** command is displayed in centi-amperes.

Configuration Examples for Chassis Management

This example shows how to display the module information about all connected Fabric Extender units:

```
switch# show module fex
FEX Mod Ports Card Type                          Model               Status.
--- --- ----- ---------------------------------- ------------------ -----------
100 1   48    Fabric Extender 48x1GE + 4x10G Mod N2K-C2248TP-1GE    present
101 1   32    Fabric Extender 32x10GE + 8x10G Mo N2K-C2232P-10GE    present
102 1   32    Fabric Extender 32x10GE + 8x10G Mo N2K-C2232P-10GE    present

FEX Mod Sw              Hw      World-Wide-Name(s) (WWN)
--- --- --------------  ------  -----------------------------------------------
100 1   4.2(1)N1(1)     0.103   --
101 1   4.2(1)N1(1)     1.0     --
102 1   4.2(1)N1(1)     1.0     --

FEX Mod  MAC-Address(es)                         Serial-Num
--- ---  --------------------------------------  ----------
100 1    000d.ece3.2800 to 000d.ece3.282f        JAF1339BDSK
101 1    000d.ecca.73c0 to 000d.ecca.73df        JAF1333ADDD
102 1    000d.ecd6.bec0 to 000d.ecd6.bedf        JAS12334ABC
```

This example shows how to display the module information about a specific Fabric Extender:

```
switch# show module fex 100
FEX Mod Ports Card Type                          Model               Status.
--- --- ----- ---------------------------------- ------------------ -----------
100 1   48    Fabric Extender 48x1GE + 4x10G Mod N2K-C2248TP-1GE    present

FEX Mod Sw              Hw      World-Wide-Name(s) (WWN)
--- --- --------------  ------  -----------------------------------------------
100 1   4.2(1)N1(1)     0.103   --

FEX Mod  MAC-Address(es)                         Serial-Num
```

```
--- ---  ------------------------------------  ----------
100 1    000d.ece3.2800 to 000d.ece3.282f        JAF1339BDSK
```

This example shows how to display the inventory information about a specific Fabric Extender:

```
switch# show inventory fex 101
NAME: "FEX 101 CHASSIS", DESCR: "N2K-C2248TP-1GE  CHASSIS"
PID: N2K-C2248TP-1GE   , VID: V00 , SN: SSI13380FSM

NAME: "FEX 101 Module 1", DESCR: "Fabric Extender Module: 48x1GE, 4x10GE Supervisor"
PID: N2K-C2248TP-1GE   , VID: V00 , SN: JAF1339BDSK

NAME: "FEX 101 Fan 1", DESCR: "Fabric Extender Fan module"
PID: N2K-C2248-FAN     , VID: N/A , SN: N/A

NAME: "FEX 101 Power Supply 2", DESCR: "Fabric Extender AC power supply"
PID: NXK-PAC-400W      , VID:  000, SN:      LIT13370QD6
```

This example shows how to display diagnostic test results for a specific Fabric Extender:

```
switch# show diagnostic result fex 101
FEX-101: 48x1GE/Supervisor  SerialNo   : JAF1339BDSK
Overall Diagnostic Result for FEX-101  : OK

Test results: (. = Pass, F = Fail, U = Untested)
TestPlatform:
0)              SPROM: ----------------> .
1)   Inband interface: ----------------> .
2)                Fan: ----------------> .
3)       Power Supply: ----------------> .
4) Temperature Sensor: ----------------> .

TestForwardingPorts:
Eth    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Port ------------------------------------------------------------------------
       .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

Eth   25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
Port ------------------------------------------------------------------------
       .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

TestFabricPorts:
Fabric 1  2  3  4
Port ------------
       .  .  .  .
```

This example shows how to display the environment status for a specific Fabric Extender:

```
switch# show environment fex 101


Temperature Fex 101:
-----------------------------------------------------------------------
Module   Sensor     MajorThresh   MinorThres   CurTemp     Status
                    (Celsius)     (Celsius)    (Celsius)
-----------------------------------------------------------------------
1        Outlet-1   60            50           33          ok
1        Outlet-2   60            50           38          ok
1        Inlet-1    50            40           35          ok
1        Die-1      100           90           44          ok
```

```
Fan Fex: 101:
--------------------------------------------------
Fan             Model                Hw         Status
--------------------------------------------------
Chassis         N2K-C2148-FAN        --         ok
PS-1            --                   --         absent
PS-2            NXK-PAC-400W         --         ok


Power Supply Fex 101:
------------------------------------------------------------------------------
Voltage: 12 Volts
------------------------------------------------
PS  Model                Power       Power     Status
                         (Watts)     (Amp)
------------------------------------------------
1   --                       --        --     --
2   NXK-PAC-400W            4.32      0.36     ok


Mod Model                Power     Power       Power     Power       Status
                         Requested Requested   Allocated Allocated
                         (Watts)   (Amp)       (Watts)   (Amp)
--- -------------------  -------   ----------  --------- ----------  ----------
1    N2K-C2248TP-1GE     0.00      0.00        0.00      0.00        powered-up


Power Usage Summary:
--------------------
Power Supply redundancy mode:                 redundant

Total Power Capacity                               4.32 W

Power reserved for Supervisor(s)                   0.00 W
Power currently used by Modules                    0.00 W

                                                  -------------
Total Power Available                              4.32 W
                                                  -------------
```

This example shows how to display the SPROM for a specific Fabric Extender:

```
switch# show sprom fex 101 all
DISPLAY FEX 101 SUP sprom contents
Common block:
 Block Signature : 0xabab
 Block Version   : 3
 Block Length    : 160
 Block Checksum  : 0x1a1e
 EEPROM Size     : 65535
 Block Count     : 3
 FRU Major Type  : 0x6002
 FRU Minor Type  : 0x0
 OEM String      : Cisco Systems, Inc.
 Product Number  : N2K-C2248TP-1GE
 Serial Number   : JAF1339BDSK
 Part Number     : 73-12748-01
 Part Revision   : 11
 Mfg Deviation   : 0
 H/W Version     : 0.103
 Mfg Bits        : 0
 Engineer Use    : 0
 snmpOID         : 9.12.3.1.9.78.3.0
 Power Consump   : 1666
```

```
 RMA Code        : 0-0-0-0
 CLEI Code       : XXXXXXXXXTBDV00
 VID             : V00
Supervisor Module specific block:
 Block Signature : 0x6002
 Block Version   : 2
 Block Length    : 103
 Block Checksum  : 0x2686
 Feature Bits    : 0x0
 HW Changes Bits : 0x0
 Card Index      : 11016
 MAC Addresses   : 00-00-00-00-00-00
 Number of MACs  : 0
 Number of EPLD  : 0
 Port Type-Num   : 1-48;2-4
 Sensor #1       : 60,50
 Sensor #2       : 60,50
 Sensor #3       : -128,-128
 Sensor #4       : -128,-128
 Sensor #5       : 50,40
 Sensor #6       : -128,-128
 Sensor #7       : -128,-128
 Sensor #8       : -128,-128
 Max Connector Power: 4000
 Cooling Requirement: 65
 Ambient Temperature: 40

DISPLAY FEX 101 backplane sprom contents:
Common block:
 Block Signature : 0xabab
 Block Version   : 3
 Block Length    : 160
 Block Checksum  : 0x1947
 EEPROM Size     : 65535
 Block Count     : 5
 FRU Major Type  : 0x6001
 FRU Minor Type  : 0x0
 OEM String      : Cisco Systems, Inc.
 Product Number  : N2K-C2248TP-1GE
 Serial Number   : SSI13380FSM
 Part Number     : 68-3601-01
 Part Revision   : 03
 Mfg Deviation   : 0
 H/W Version     : 1.0
 Mfg Bits        : 0
 Engineer Use    : 0
 snmpOID         : 9.12.3.1.3.914.0.0
 Power Consump   : 0
 RMA Code        : 0-0-0-0
 CLEI Code       : XXXXXXXXXTDBV00
 VID             : V00
Chassis specific block:
 Block Signature : 0x6001
 Block Version   : 3
 Block Length    : 39
 Block Checksum  : 0x2cf
 Feature Bits    : 0x0
 HW Changes Bits : 0x0
 Stackmib OID    : 0
 MAC Addresses   : 00-0d-ec-e3-28-00
 Number of MACs  : 64
 OEM Enterprise  : 0
 OEM MIB Offset  : 0
 MAX Connector Power: 0
```

```
WWN software-module specific block:
 Block Signature : 0x6005
 Block Version   : 1
 Block Length    : 0
 Block Checksum  : 0x66
wwn usage bits:
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00
 00 00
License software-module specific block:
 Block Signature : 0x6006
 Block Version   : 1
 Block Length    : 16
 Block Checksum  : 0x86f
lic usage bits:
 ff ff ff ff ff ff ff ff

DISPLAY FEX 101 power-supply 2 sprom contents:
Common block:
 Block Signature : 0xabab
 Block Version   : 3
 Block Length    : 160
 Block Checksum  : 0x1673
 EEPROM Size     : 65535
 Block Count     : 2
 FRU Major Type  : 0xab01
 FRU Minor Type  : 0x0
 OEM String      : Cisco Systems Inc   NXK-PAC-400W
 Product Number  : NXK-PAC-400W
 Serial Number   :      LIT13370QD6
 Part Number     :         341
 Part Revision   : -037
 CLEI Code       : 5-01     01  000
 VID             :  000
 snmpOID         : 12336.12336.12336.12336.12336.12336.12374.12336
```

```
 H/W Version     : 43777.2
 Current         : 36
 RMA Code        : 200-32-32-32
Power supply specific block:
 Block Signature : 0x0
 Block Version   : 0
 Block Length    : 0
 Block Checksum  : 0x0
 Feature Bits    : 0x0
 Current 110v    : 36
 Current 220v    : 36
 Stackmib OID    : 0
```

Configuring the Cisco Nexus N2248TP-E Fabric Extender

The Cisco Nexus 2248TP-E Fabric Extender supports all of the CLI commands of the Cisco Nexus 2248TP Fabric Extender with additional commands to configure the following:

- Shared buffer (FEX global level)
- Queue limit in ingress direction (FEX global level and interface level)
- Queue limit in egress direction (FEX global level and interface level)
- No drop class over a distance of 3000 meters between the FEX and switch (FEX global level)

Configuring the Shared Buffer

The following are guidelines for the configuration of the shared buffer:

- Configuring the shared buffer is done at the FEX global level.
- The total available buffer is 32 MB which is shared in both the ingress and egress directions.
- The default size of the shared buffer is 25392KB.

  However, when configuring an Ethernet-based pause no-drop class, the shared buffer size changes to 10800 KB. This change is required to increase the dedicated buffer that supports the pause no-drop class. The pause no-drop class does not use buffer space from the shared-pool.

Note Performing these commands might result in traffic disruption on all ports.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**fex** *chassis_id*
	Example:
	`switch(config)# fex 100`
	`switch(config-fex)#`
		Enters configuration mode for the specified FEX.
		The range of the *chassis_id* value is 100 to 199.

Step 3	**hardware N2248TP-E shared-buffer-size** *buffer-size* **Example:** `switch(config-fex)# hardware N2248TP-E shared-buffer-size 25000`	Specifies the shared buffer size (KB). The range of the *buffer-size* value is 10800 KB to 25392 KB. **Note** The **hardware N2248TP-E shared-buffer-size** command specifies the default shared buffer size of 25392 KB.

Example

This example shows how to configure the shared buffer.

```
switch# configure terminal
switch(config)# fex 100
switch(config-fex)# hardware N2248TP-E shared-buffer-size 25000
switch(config-fex)#
```

Configuring the Queue Limit at the Global Level

The following are guidelines for the configuration of the queue limit:

- The tx queue limit specifies the buffer size used for each queue in the egress (n2h) direction.
- The rx queue limit specifies the buffer size used for each port in the ingress (h2n) direction.
- You can adjust the ingress queue limit when the FEX uplink experiences temporary congestion.
- You can adjust the egress queue limit for improved burst absorption or in a situation where there is a many to one traffic pattern.
- When you disable the tx queue limit, any output port is able to use the entire shared buffer.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**fex** *chassis_id* **Example:**	Enters configuration mode for the specified FEX.

	Command or Action	Purpose
	`switch(config)# fex 100` `switch(config)#`	The range of the *chassis_id* value is 100 to 199.
Step 3	**hardware N2248TP-E queue-limit** *queue-limit* **tx|rx** **Example:** `switch(config-fex)# hardware N2248TP-E queue-limit 83000 tx`	Controls the egress (tx) or ingress (rx) queue tail drop threshold level on a FEX. • The default queue limit for tx (egress) is 4 MB. **Note** The **hardware N2248TP-E queue-limit** command specifies the default tx queue limit. • The default queue-limit for rx (ingress) is 1 MB. **Note** The **hardware N2248TP-E queue-limit rx** command specifies the default rx queue limit.

Example

This example shows how to configure the queue limit.

```
switch# configure terminal
switch(config)# fex 100
switch(config-fex)# hardware N2248TP-E queue-limit 83000 tx
switch(config-fex)#
```

Configuring the Queue Limit at the Port Level

You can overwrite the global level configuration by configuring the queue limit at the port level.

You can also disable the queue limit at the port level.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *chassis_id / slot/port* **Example:** `switch(config)# interface ethernet 100/1/1`	Enters interface configuration mode.

	Command or Action	Purpose
Step 3	**hardware N2248TP-E queue-limit** *queue-limit* **tx|rx** **Example:** `switch(config-if)# hardware N2248TP-E queue-limit 83000 tx`	Controls the egress (tx) or ingress (rx) queue tail drop threshold level on a FEX. • The default queue limit for tx (egress) is 4 MB. • The default queue limit for rx (ingress) is 1 MB.

Example

This example shows how to configure the queue limit.

```
switch# configure terminal
switch(config)# interface ethernet 100/1/1
switch(config-if)# hardware N2248TP-E queue-limit 83000 tx
switch(config-if)#
```

Configuring the Uplink Distance

The Cisco Nexus N2248TP-E FEX supports a pause no-drop class up to a distance of 3000 meters between the FEX and the switch.

The default cable length between the FEX and the switch is 300 meters.

Note When the pause no-drop class is not configured, the uplink distance configuration has no effect.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**fex** *chassis_id*
	Example:
	`switch(config)# fex 100`
	`switch(config-fex)#`
		Enters configuration mode for the specified FEX.
		The range of the *chassis_id* value is 100 to 199.

Step 3	**hardware N2248TP-E uplink-pause-no-drop distance** *distance-value* **Example:** `switch(config-fex)# hardware N2248TP-E uplink-pause-no-drop distance 3000`	Specifies the no-drop distance between the FEX and the switch. The maximum distance is 3000 meters.

	Command or Action	Purpose
		Note The **hardware N2248TP-E uplink-pause-no-drop distance** command specifies the default 300 meter cable length.

Example

This example shows how to configure the uplink distance.

```
switch# configure terminal
switch(config)# fex 100
switch(config-fex)# hardware N2248TP-E uplink-pause-no-drop distance 3000
switch(config-fex)#
```

Configuring the Cisco Nexus N2248PQ Fabric Extender

The Cisco Nexus 2248PQ Fabric Extender supports all of the CLI commands of the Cisco Nexus 2248TP Fabric Extender with additional commands to configure the following:

- Shared buffer (FEX global level)
- Load-balancing queues (FEX global level)
- No drop class over a distance of 3000 meters between the FEX and switch (FEX global level)

Configuring the Shared Buffer

The following are guidelines for the configuration of the shared buffer:

- Configuring the shared buffer is done at the FEX global level.
- The total available buffer is 16 MB which is shared in both the ingress and egress directions.
- The default size of the shared buffer is 10240KB.

Note Performing these commands might result in traffic disruption on all ports.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**fex** *chassis_id*
	Example:
	`switch(config)# fex 100`
	`switch(config-fex)#`
		Enters configuration mode for the specified FEX.
		The range of the *chassis_id* value is 100 to 199.

Step 3	**hardware N2248PQ shared-buffer-size** *buffer-size* **Example:** `switch(config-fex)# hardware N2248PQ shared-buffer-size 8096`	Specifies the shared buffer size (KB). The range of the *buffer-size* value is 3072 KB to 10240 KB. **Note** The **hardware N2248PQ shared-buffer-size** command specifies the default shared buffer size of 10240 KB.

Example

This example shows how to configure the shared buffer.

```
switch# configure terminal
switch(config)# fex 100
switch(config-fex)# hardware N2248PQ shared-buffer-size 8096
switch(config-fex)#
```

Configuring the Uplink Distance

The Cisco Nexus N2248PQ FEX supports a pause no-drop class up to a distance of 3000 meters between the FEX and the switch.

The default cable length between the FEX and the switch is 300 meters.

Note When the pause no-drop class is not configured, the uplink distance configuration has no effect.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**fex** *chassis_id*
	Example:
	`switch(config)# fex 100`
	`switch(config-fex)#`
		Enters configuration mode for the specified FEX.
		The range of the *chassis_id* value is 100 to 199.

	Command or Action	Purpose
Step 3	**hardware N2248PQ uplink-pause-no-drop distance** *distance-value* **Example:** `switch(config-fex)# hardware N2248PQ uplink-pause-no-drop distance 3000`	Specifies the no-drop distance between the FEX and the switch. The maximum distance is 3000 meters. **Note** The **hardware N2248PQ uplink-pause-no-drop distance** command specifies the default 300 meter cable length.

Example

This example shows how to configure the uplink distance.

```
switch# configure terminal
switch(config)# fex 100
switch(config-fex)# hardware N2248PQ uplink-pause-no-drop distance 3000
switch(config-fex)#
```

Configuring Slow Drain

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **fex** *chassis_id*	Enters configuration mode for the specified FEX. The range of the *chassis_id* value is 100 to 199.
Step 3	switch(config-fex)# **hardware** *fex* **slow-port-error-disable-time** *val*	Specifies the FEX and the time threshold. The value of *fex* is the PID of the configured FEX. The range of *val* is from 200ms to 1000ms. The default value is 1000 ms.

Example

This example shows how to configure the slow drain feature on the N2232P FEX:

```
switch# configure terminal
switch(config)# fex N2232P
switch(config-fex)# hardware N2232P slow-port-error-disable-time 500
```

Load-balancing queues at the FEX global level

The Cisco Nexus 2248PQ provides 8 load balancing queues. These load balancing queues are designed to resolve port congestion.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

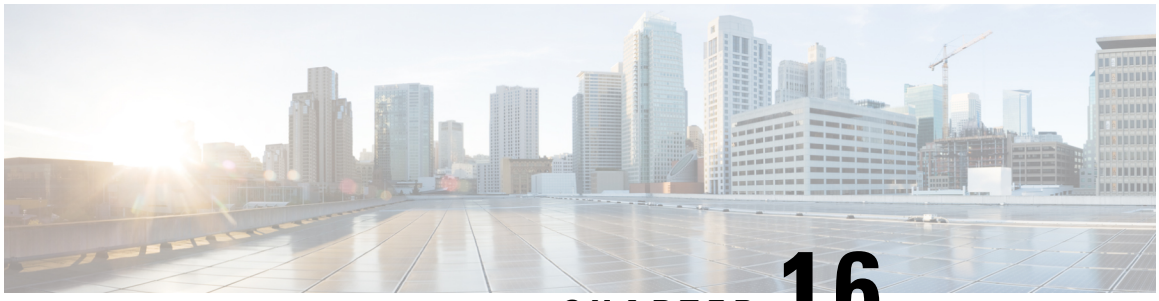
Step 2	**fex** *chassis_id*
	Example:
	`switch(config)# fex 100`
	`switch(config)#`
		Enters configuration mode for the specified FEX.
		The range of the *chassis_id* value is 100 to 199.

Step 3	**hardware N2248PQ uplink-load-balance-mode** **Example:** `switch(config-fex)# hardware N2248PQ uplink-load-balance-mode`	Enables and disables load balancing queues at the FEX global level.

Example

This example shows how to configure the load balance queues.

```
switch# configure terminal
switch(config)# fex 100
switch(config-fex)# hardware N2248PQ uplink-load-balance-mode
switch(config-fex)#
```



CHAPTER 16

Configuring VM-FEX

This chapter contains the following sections:

Information About VM-FEX

VM-FEX Overview

Based on the (prestandard) IEEE 802.1Qbh port extender technology, Cisco Virtual Machine Fabric Extender (VM-FEX) extends the fabric from the switch chassis to the Virtual Machine (VM). Each VM is associated with a network adapter vNIC, which is associated with a virtual Ethernet (vEthernet or vEth) port on the parent switch. This dedicated virtual interface can be managed, monitored, and spanned in the same way as a physical interface. Local switching in the hypervisor is eliminated, with all switching being performed by the physical switch.

VM-FEX Components

Server

VM-FEX is supported by Cisco UCS C-Series rack-mount servers with the VMware virtualization environment as the hypervisor.

The configuration of the server is performed using the Cisco Integrated Management Controller (CIMC) interface, which provides both a GUI and a CLI interface. The configuration of the hypervisor and virtualization services is performed using the VMware vSphere client.

For information about CIMC and VM-FEX configuration, see the following documents:

- *Cisco UCS C-Series Servers Integrated Management Controller GUI Configuration Guide*
- *Cisco UCS Manager VM-FEX for VMware GUI Configuration Guide*

Virtual Interface Card Adapter

VM-FEX is supported by the Cisco UCS P81E Virtual Interface Card (VIC), a dual-port 10 Gigabit Ethernet PCIe adapter that supports static or dynamic virtualized interfaces, including up to 128 virtual network interface cards (vNICs).

The configuration of the VIC and its vNICs is performed using the CIMC interface on the Cisco UCS C-Series servers.

FEX

The physical ports of the server can be connected directly to the switch or to a fabric extender (FEX) connected to the switch. VM-FEX is supported by the Cisco Nexus Fabric Extender.

VM-FEX and AFEX require that the FEX is connected with a fabric PO and not individual links.

Switch

VM-FEX is supported by the Cisco Nexus device. Although a single switch chassis can be connected with VM-FEX, a typical application uses a pair of switches deployed as a virtual port channel (vPC) domain.

On the switch, a vEthernet interface represents the vNIC. All operations performed by the network administrator are performed on the vEthernet interface.

VM-FEX Terminology

The following terms are used in describing VM-FEX components and interfaces:

virtual Ethernet interface

A virtual Ethernet interface (vEthernet or vEth) represents the switch port that is connected to the vNIC of a virtual machine. Unlike a traditional switch interface, a vEth interface's name does not indicate the module with which the port is associated. Where a traditional physical switch port is specified as GigX/Y, where X is the module number and Y is the port number on the module, a vEth interface is specified as vEthY. This notation allows the interface to keep the same name when the VM migrates to another physical server.

dynamic interface

A dynamic interface is a vEthernet interface that is configured automatically as a result of adapter and switch communications. The provisioning model of a dynamic interface consists of the configuration on the switch of a vEthernet port profile, which is propagated to the network adapter as a port group, followed by the association of the port group with the vNIC. The port profile is created in the switch by the network administrator, while the association with the vNIC is performed on the adapter by the server administrator.

static interface

A static interface is configured manually on the switch and the adapter. A static virtual adapter can be a vNIC or a virtual host adapter bus (vHBA). A static interface can be a vEthernet or a virtual Fibre Channel (vFC) interface bound to a static vEthernet interface.

In one method of creating a static vEthernet, the network administrator assigns a channel number (equivalent to a VN-Tag or prestandard IEEE 802.1BR tag number) to the vEthernet. The server administrator must be sure to define a vNIC on the adapter with the same channel number.

In another method, the network administrator can create a static floating vEthernet by configuring the vEthernet with a virtual switching instance (VSI) MAC address and DVPort ID.

floating vEthernet interface

In a hypervisor environment, each vNIC on the network adapter is associated with one virtual machine (VM). VMs can migrate from one physical server to another. A virtual interface that migrates with a VM and virtual network link is called a floating vEthernet interface.

fixed vEthernet interface

A fixed vEthernet interface is a virtual interface that does not support migration across physical interfaces. For fixed vEthernet (static or dynamic), an administrator can change configurations at any time. The binding of the vEthernet interface number to a channel number is persistent unless the administrator changes it.

Licensing Requirements for VM-FEX

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	A VM-FEX license is required for each Cisco Nexus device. The license package name is VMFEX_ FEATURE_PKG. A grace period of 120 days starts when you first configure the licensed feature. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Default Settings for VM-FEX

The following table lists the default settings for parameters that are relevant to VM-FEX:

Parameters	Default
Virtualization feature set	Disabled
FEX	Disabled
VM-FEX	Disabled
LLDP	Enabled
vPC	Disabled
svs vethernet auto-setup	Enabled
FCoE	Disabled

Configuring VM-FEX

Overview of the VM-FEX Configuration Steps

The following steps outline the necessary sequence of procedures for configuring VM-FEX between the switch and the server hosting the VMs. Procedures to be performed on the switch are described in this document. For procedures to be performed on the server or the VMware vCenter, refer to the server and vCenter documentation.

Procedure

Step 1 Server: Create vNICs on VIC adapter.
a) Create two static vNICs to be used as uplinks from the host.
b) Create up to 112 VM-FEX interfaces.
c) Reboot the server.

Step 2 Switch: Enable VM-FEX and other required services.

See Enabling Features Required for VM-FEX, on page 235.

Step 3 Switch: Configure two static vEthernet interfaces and bind them to the physical port and channel.

See Configuring the Fixed Static Interfaces, on page 236.

Step 4 Switch: Define port profiles to be associated with the VMs.

See Configuring a Port Profile for the Dynamic Interfaces, on page 238.

Step 5 Switch: Verify that the two static vEthernet interfaces are active and associated with the vEthernet interfaces of the switch.

See Verifying the Status of the Virtual Interfaces, on page 242.

Step 6 Switch and vCenter: Install XML certificate from switch to vCenter.
a) Switch: Enable HTTP using the **feature http** command in global configuration mode.
b) From a web browser, access the IP address of the switch and download the displayed XML certificate.
c) Switch: Disable HTTP using the **no feature http** command in global configuration mode.
d) vCenter: Install the XML certificate plugin.

Step 7 Switch: Enable vPC and register the vPC system to the vCenter as a distributed virtual switch (DVS).

See Configuring an SVS Connection to the vCenter Server, on page 239.

Step 8 vCenter: Create a datacenter on the vCenter.

Step 9 Switch: Activate and verify the SVS connection to the vCenter.

See Activating an SVS Connection to the vCenter Server, on page 241 and Verifying the Connection to the vCenter Server, on page 244.

Step 10 vCenter: Verify that the port profiles (port groups) are propagated to the vCenter.

Step 11 Server: Add resources to the DVS.

a) Add the ESX host to the DVS.
b) Add the static vNICs as uplinks to the DVS.
c) Associate VMs to the port groups defined by the switch.
d) Activate the VMs.

Step 12 Switch: Verify that the dynamic vNICs are active, assigned to VMs, and connected to the vEthernet interfaces of the switch.

See Verifying the Status of the Virtual Interfaces, on page 242.

Step 13 Server: Verify that the interfaces are active and assigned to the VMs.

Step 14 vCenter: Verify that the dynamic vNICs are active.

Enabling Features Required for VM-FEX

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**install feature-set virtualization**	Installs the virtualization feature set on the switch.
Step 3	**feature-set virtualization**	Enables the virtualization feature set on the switch. This feature set enables the use of static vEthernet interfaces.
Step 4	**feature fex**	Enables FEX features on the switch.
Step 5	**feature vmfex**	Enables VM-FEX features on the switch. This feature set enables the use of dynamic vEthernet interfaces.
Step 6	**feature vpc**	Enables a virtual port channel (vPC) on the switch.
Step 7	(Optional) **vethernet auto-create**	Globally enables the automatic creation of virtual Ethernet interfaces. This feature is not required if the fixed vEthernet interfaces are statically configured.
Step 8	(Optional) **feature fcoe**	Enables Fibre Channel over Ethernet (FCoE) on the switch.
Step 9	(Optional) **end**	Return to privileged EXEC mode.
Step 10	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.
Step 11	(Optional) **reload**	Reloads the switch.

Example

This example shows how to enable the features required for VM-FEX:

```
switch# configure terminal
switch(config)# install feature-set virtualization
switch(config)# feature-set virtualization
switch(config)# feature fex
switch(config)# feature vmfex
switch(config)# feature vpc
switch(config)# vethernet auto-create
switch(config)# feature fcoe
switch(config)# end
switch# copy running-config startup-config
switch# reload
```

Configuring the Fixed Static Interfaces

You can configure two physical interfaces and binds two virtual interfaces to each physical interface, creating fixed static vEthernet interfaces. For more information on configuring fixed static interfaces, see the Adapter-FEX Configuration Guide for your device.

With redundant switches, you can perform the following procedure with identical settings on both the primary and secondary switches.

Before you begin

- VM-FEX and other required services must be enabled on the switches.
- Two static vNICs must be configured on the VIC adapter installed in the host server.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**interface ethernet** *slot/port*	Enters interface configuration mode for the first Ethernet port.
Step 3	**shutdown**	Disables local traffic on the interface. **Note** Shutting down the interface before enabling VN-Tag mode prevents the dynamic creation of a fixed vEthernet interface.
Step 4	**switchport mode vntag**	Enables port extender support on the interface.
Step 5	**interface ethernet** *slot/port*	Enters interface configuration mode for the second Ethernet port.
Step 6	**shutdown**	Disables local traffic on the interface.

	Command or Action	Purpose
Step 7	**switchport mode vntag**	Enables port extender support on the interface.
Step 8	**interface vethernet** *interface-number*	Enters configuration mode for the first virtual interface for the first Ethernet port.
Step 9	**bind interface ethernet** *slot/port* **channel** *channel-number*	Binds the virtual interface to the physical interface and the specified port channel. **Note** The port channel numbers of the virtual interfaces must match those configured on the vNICs.
Step 10	**no shutdown**	Enables local traffic on the interface.
Step 11	**interface vethernet** *interface-number*	Enters configuration mode for the second virtual interface for the first Ethernet port.
Step 12	**bind interface ethernet** *slot/port* **channel** *channel-number*	Binds the virtual interface to the physical interface and the specified port channel.
Step 13	**no shutdown**	Enables local traffic on the interface.
Step 14	**interface vethernet** *interface-number*	Enters configuration mode for the first virtual interface for the second Ethernet port.
Step 15	**bind interface ethernet** *slot/port* **channel** *channel-number*	Binds the virtual interface to the physical interface and the specified port channel.
Step 16	**no shutdown**	Enables local traffic on the interface.
Step 17	**interface vethernet** *interface-number*	Enters configuration mode for the second virtual interface for the second Ethernet port.
Step 18	**bind interface ethernet** *slot/port* **channel** *channel-number*	Binds the virtual interface to the physical interface and the specified port channel.
Step 19	**no shutdown**	Enables local traffic on the interface.
Step 20	**interface ethernet** *slot/port*	Enters configuration mode for the first Ethernet port.
Step 21	**no shutdown**	Enables local traffic on the interface.
Step 22	**interface ethernet** *slot/port*	Enters configuration mode for the second Ethernet port.
Step 23	**no shutdown**	Enables local traffic on the interface.
Step 24	With redundant switches, repeat this procedure with identical settings on the secondary switch.	

Example

This example shows how to configure two physical interfaces, binds two virtual interfaces to each physical interface, and enables the interfaces:

```
switch-1# configure terminal
switch-1(config)# interface ethernet 1/17
switch-1(config-if)# shutdown
switch-1(config-if)# switchport mode vntag
switch-1(config-if)# interface ethernet 1/18
switch-1(config-if)# shutdown
switch-1(config-if)# switchport mode vntag

switch-1(config-if)# interface vethernet 1
switch-1(config-if)# bind interface ethernet 1/17 channel 10
switch-1(config-if)# no shutdown
switch-1(config-if)# interface vethernet 3
switch-1(config-if)# bind interface ethernet 1/17 channel 11
switch-1(config-if)# no shutdown

switch-1(config-if)# interface vethernet 2
switch-1(config-if)# bind interface ethernet 1/18 channel 10
switch-1(config-if)# no shutdown
switch-1(config-if)# interface vethernet 4
switch-1(config-if)# bind interface ethernet 1/18 channel 11
switch-1(config-if)# no shutdown

switch-1(config-if)# interface ethernet 1/17
switch-1(config-if)# no shutdown
switch-1(config-if)# interface ethernet 1/18
switch-1(config-if)# no shutdown

switch-1(config-if)#
```

What to do next

Verify the status of the connection between the static interfaces and the static vNICs on the host server.

Configuring a Port Profile for the Dynamic Interfaces

You can configure a port profile for dynamic virtual interfaces. This port profile is exported to the VMware vCenter distributed virtual switch (DVS) as a port-group.

With redundant switches, you can perform the following procedure with identical settings on both the primary and secondary switches.

Before you begin

- Dynamic vNICs must be configured on the VIC adapter installed in the host server.
- The VLAN specified in the port profile must be created.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**port-profile type vethernet** *profilename*	Enters configuration mode for the specified port profile, creating it if necessary.
Step 3	(Optional) **switchport mode access**	Configures the interface to be in access mode.
Step 4	(Optional) **switchport access vlan** *vlan-id*	Specifies the VLAN when the interface is in access mode.
Step 5	**dvs-name** {**all** | *name*}	Specifies the vCenter DVS to which the port profile is exported as a port-group. With the keyword **all**, the port profile is exported to all DVSs in the vCenter.
Step 6	(Optional) **port-binding dynamic**	Specifies dynamic port binding. The port is connected when the VM is powered on and disconnected when the VM is powered off. Max-port limits are enforced. The default is static port binding.
Step 7	**state enabled**	Enables the port profile.

Example

This example configures a port profile for dynamic virtual interfaces:

```
switch-1# configure terminal
switch-1(config)# port-profile type vethernet vm-fex-vlan-60
switch-1(config-port-prof)# switchport mode access
switch-1(config-port-prof)# switchport access vlan 60
switch-1(config-port-prof)# dvs-name all
switch-1(config-port-prof)# port-binding dynamic
switch-1(config-port-prof)# state enabled
switch-1(config-port-prof)#
```

Configuring an SVS Connection to the vCenter Server

You can configure a secure connection from the switch to the vCenter Server.

With redundant switches, perform this procedure on both the primary and the secondary switches. In normal operation, only the primary switch connects to the vCenter, with the secondary switch connecting only upon a failure of the primary.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**svs connection** *svs-name*	Enables and enters configuration mode for an SVS connection from the switch to the vCenter Server.
Step 3	**protocol vmware-vim**	Enables the VMware Infrastructure Software Development Kit (VI SDK), which allows clients to communicate with the vCenter.
Step 4	**vmware dvs datacenter-name** *dc-name*	Creates a VMware distributed virtual switch (DVS) in the specified datacenter.
Step 5	**dvs-name** *dvs-name*	Configures a name for the DVS in the vCenter Server.
Step 6	Choose one: • **remote ip address** *ipv4-addr* [**port** *port-num*] [**vrf** {*vrf-name* | **default** | **management**}] • **remote hostname** *host-name* [**port** *port-num*] [**vrf** {*vrf-name* | **default** | **management**}]	Specifies the hostname or IP address for the vCenter Server. Optionally, specifies the port number and VRF.
Step 7	**install certificate** {**bootflash:**[*//server/*] | **default**}	Installs a certificate that is used to connect to the vCenter Server. The *server* argument specifies the boot flash memory location to install the certificate. The argument value can be **module-1**, **sup-1**, **sup-active**, or **sup-local**.
Step 8	**extension-key:** *extn-ID*	Configures the extension key to be used to connect to the vCenter Server. **Note** With redundant switches, perform this step only on the primary switch. The key is automatically synchronized with the secondary switch.

Example

This example shows how to configure the SVS connection on the primary switch and the secondary switch:

```
switch-1# configure terminal
switch-1(config)# svs connection 2VC
switch-1(config-svs-conn)# protocol vmware-vim
switch-1(config-svs-conn)# vmware dvs datacenter-name DC1
switch-1(config-svs-conn)# dvs-name Pod1
switch-1(config-svs-conn)# remote ip address 192.0.20.125 port 80 vrf management
switch-1(config-svs-conn)# install certificate default
```

```
switch-1(config-svs-conn)# extension-key: Cisco_Nexus_6004_1543569268
switch-1(config-svs-conn)#

switch-2# configure terminal
switch-2(config)# svs connection 2VC
switch-2(config-svs-conn)# protocol vmware-vim
switch-2(config-svs-conn)# vmware dvs datacenter-name DC1
switch-2(config-svs-conn)# dvs-name Pod1
switch-2(config-svs-conn)# remote ip address 192.0.20.125 port 80 vrf management
switch-2(config-svs-conn)# install certificate default
switch-2(config-svs-conn)#
```

What to do next

Activate the connection on the primary switch only.

Activating an SVS Connection to the vCenter Server

You can activate a connection from the switch to the vCenter Server.

Before you begin

- The vCenter Server must be running and reachable.
- You must have already registered an extension with the vCenter Server.
- The SVS connection must be configured on the switch.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**svs connection** *svs-name*	Enables and enters configuration mode for an SVS connection from the switch to the vCenter Server.
Step 3	[**no**] **connect**	Initiates a connection with the vCenter Server. **Note** With redundant switches, perform this step on both the primary and secondary switches. Only the primary will connect. The switch connects to the vCenter and becomes a DVS.

Example

This example shows how to connect to a vCenter Server:

```
switch-1# configure terminal
switch-1(config)# svs connection 2VC
switch-1(config-svs-conn)# connect
Note: Command execution in progress..please wait
switch-1(config-svs-conn)#
```

Verifying the VM-FEX Configuration

Verifying the Status of the Virtual Interfaces

Use the following commands to display status information for virtual interfaces.

Command	Purpose
show interface vethernet *interface-number* [**detail**]	Displays the status of the virtual interface. Perform this procedure on each static virtual interface to verify that the interface is active and bound to the physical interface.
show interface virtual status vm-fex	Displays information about all floating virtual interfaces.
show interface virtual summary vm-fex	Displays summary information about virtual Ethernet interfaces.
show interface virtual status bound interface ethernet *port/slot*	Displays information about virtual interfaces on a bound Ethernet interface.
show interface virtual summary bound interface ethernet *port/slot*	Displays summary information about virtual interfaces on a bound Ethernet interface.

This example shows how to display status and configuration information about a static interface:

```
switch-1# show interface vethernet 1

Vethernet1 is up
Bound Interface is Ethernet1/17
Hardware is Virtual, address is 0005.73fc.24a0
Port mode is access
Speed is auto-speed
Duplex mode is auto
300 seconds input rate 0 bits/sec, 0 packets/sec
300 seconds output rate 0 bits/sec, 0 packets/sec
Rx
0 unicast packets  0 multicast packets  0 broadcast packets
0 input packets  0 bytes
0 input packet drops
Tx
0 unicast packets  0 multicast packets  0 broadcast packets
0 output packets  0 bytes
0 flood packets
0 output packet drops

switch-1# show interface vethernet 1 detail
```

```
vif_index: 20
--------------------------
  veth is bound to interface Ethernet1/17 (0x1a010000)
  priority: 0
  vntag: 16
  status: active
  channel id: 10
  registered mac info:
      vlan 0 - mac 00:00:00:00:00:00
      vlan 0 - mac 58:8d:09:0f:0b:3c
      vlan 0 - mac ff:ff:ff:ff:ff:ff

switch-1#
```

This example shows how to display status and summary information about all virtual interfaces:

```
switch-1# show interface virtual status vm-fex

Interface VIF-index   Bound If      Chan  Vlan  Status     Mode     Vntag
-------------------------------------------------------------------------
Veth32769 VIF-37      Eth1/20       ----   101  Up         Active       7
Veth32770 VIF-39      Eth1/20       ----     1  Up         Active       8
Veth32771 VIF-41      Eth1/20       ----     1  Up         Standby      9
Veth32772 VIF-43      Eth1/20       ----     1  Up         Active      10
Veth32773 VIF-47      Eth1/20       ----     1  Up         Active      12
Veth32774 VIF-48      Eth1/20       ----     1  Up         Standby     13
Veth32775 VIF-49      Eth1/20       ----     1  Up         Active      14


switch-1# show interface virtual summary vm-fex

Veth       Bound      Channel/  Port                     Mac               VM
Interface  Interface  DV-Port   Profile                  Address           Name
-------------------------------------------------------------------------------------
Veth32769  Eth1/20    7415    Unused_Or_Quarantine_Veth  00:50:56:9b:33:a7  ESX145_1_RH55.
Veth32770  Eth1/20    7575    Unused_Or_Quarantine_Veth  00:50:56:9b:33:a8  ESX145_1_RH55.
Veth32771  Eth1/20    7576    Unused_Or_Quarantine_Veth  00:50:56:9b:33:a9  ESX145_1_RH55.
Veth32772  Eth1/20    7577    Unused_Or_Quarantine_Veth  00:50:56:9b:33:aa  ESX145_1_RH55.
Veth32773  Eth1/20    7578    Unused_Or_Quarantine_Veth  00:50:56:9b:33:ac  ESX145_1_RH55.
Veth32774  Eth1/20    7579    Unused_Or_Quarantine_Veth  00:50:56:9b:33:ad  ESX145_1_RH55.
Veth32775  Eth1/20    7580    Unused_Or_Quarantine_Veth  00:50:56:9b:33:ae  ESX145_1_RH55.
Veth32776  Eth1/20    7607    Unused_Or_Quarantine_Veth  00:50:56:9b:33:ab  ESX145_1_RH55.

switch-1#
```

This example shows how to display status and summary information about fixed vEthernet interfaces:

```
switch-1# show interface virtual status bound interface ethernet 1/20

Interface VIF-index   Bound If      Chan  Vlan  Status     Mode     Vntag
-------------------------------------------------------------------------
Veth32769 VIF-16      Eth1/20          1     1  Up         Active       2
Veth32770 VIF-17      Eth1/20          5     1  Up         Active      46
Veth32771 VIF-18      Eth1/20          8     1  Up         Active      49
Veth32772 VIF-19      Eth1/20          9     1  Up         Active      50
Veth32773 VIF-20      Eth1/20         11     1  Up         Active      52
Veth32774 VIF-21      Eth1/20         12     1  Up         Active      53
Veth32775 VIF-22      Eth1/20         13     1  Up         Active      54
Veth32776 VIF-23      Eth1/20         14     1  Up         Active      55
Veth32777 VIF-24      Eth1/20         15     1  Up         Active      56
Total 9 Veth interfaces
```

```
switch-1# show interface virtual summary bound interface ethernet 1/20

Veth       Bound        Channel/  Port        Mac               VM
Interface  Interface    DV-Port   Profile     Address           Name
------------------------------------------------------------------------------
Veth32769  Eth1/20         1    sample
Veth32770  Eth1/20         5    sample
Veth32771  Eth1/20         8    sample
Veth32772  Eth1/20         9    sample
Veth32773  Eth1/20        11    sample
Veth32774  Eth1/20        12    sample
Veth32775  Eth1/20        13    sample
Veth32776  Eth1/20        14    sample
Veth32777  Eth1/20        15    sample
Total 9 Veth interfaces

switch-1#
```

Verifying the Connection to the vCenter Server

Procedure

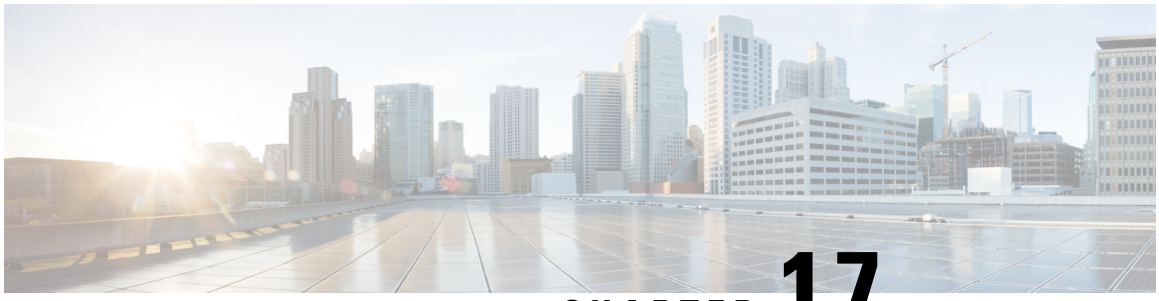
	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**show svs connections** [*svs-name*]	Displays the current SVS connections.

Example

This example shows how to display the details of the SVS connection:

```
switch-1# configure terminal
switch-1(config)# show svs connections

Local Info:
-----------
connection 2VC:
    ip address: 192.0.20.125
    remote port: 80
    vrf: management
    protocol: vmware-vim https
    certificate: default
    datacenter name: DC1
    extension key: Cisco_Nexus_6004_1945593678
    dvs name: Pod1
    DVS uuid: cd 05 25 50 6d a9 a5 c4-eb 9c 8f 6b fa 51 b1 aa
    config status: Enabled
    operational status: Connected
    sync status: in progress
    version: VMware vCenter Server 6.0.2 build-388657


Peer Info:
----------
    hostname: -
    ip address: -
    vrf:
```

```
    protocol: -
    extension key: Cisco_Nexus_6004_1945593678
    certificate: default
      certificate match: TRUE
    datacenter name: DC1
    dvs name: Pod1
    DVS uuid: cd 05 25 50 6d a9 a5 c4-eb 9c 8f 6b fa 51 b1 aa
    config status: Disabled
    operational status: Connected

switch-1(config)#
```



CHAPTER 17

Configuring MAC/ARP Hardware Resource Carving Template

This chapter contains the following sections:

Information About MAC/ARP Hardware Resource Carving Template

On the Cisco Nexus device, the IPv4/IPv6 and unicast/multicast entries share the same tables. In addition, the same tables are shared by Station Table Management (STM) and the Host Route Table (HRT). STM is the part of the host table that holds the MAC entries. HRT is the part of the host table that holds ARP, IPv6 ND, and /32 host routes. The STM/HRT template profile feature is specific to the Cisco Nexus device. This feature provides you with a flexibility to carve STM & HRT table sizes per their requirements. The total table size is 256k. You can apply any of the following four pre-defined templates:

Template Profiles	Specifications
hrt-128-stm-128	HRT size: 128k, STM size: 128k (default size)
hrt-96-stm-160	HRT size: 96k, STM size: 160k
hrt-64-stm-192	HRT size: 64k, STM size: 192k
hrt-32-stm-224	HRT size: 32k, STM size: 224k

Note The hrt-96-stm-160 and hrt-32-stm-224 template profiles are not recommended in the presence of IPv6 entries. This is because these two profiles result in an odd number of SRAMs available for the HRT table. Insertion of IPv6 entries need free spaces in 2 consecutive SRAMs.

The recommended maximum ARP percentage of the configured value is 50%. The recommended maximum MAC percentage of the configured value is 90%. For example, if the profile is set to hrt-96-stm-160, 50% of 96k (48k) is the recommended maximum ARP entries that a switch can have.

When applying or unapplying a template profile, you need to enter the **copy running-config startup-config** command and reload the switch in order to activate the newly applied/default template. These commands are per-switch based, therefore they need to be configured explicitly on a vPC peer switch.

Configuring the MAC/ARP Hardware Resource Template

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware profile route resource service-template** *template-name*	Commits a specified pre-defined template. Four pre-defined stm/hrt templates exist: • hrt-128-stm-128 Default value • hrt-96-stm-160 • hrt-64-stm-192 • hrt-32-stm-224 When entering this command, a message is displayed telling you the applied stm/hrt template will be activated upon switch reload. Upon rebooting, this pre-defined template is applied. If this command is issued multiple times, the latest stm/hrt template is applied.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the hrt-96-stm-160 template:

```
switch# configure terminal
switch(config)# hardware profile route resource service-template hrt-96-stm-160
switch(config)# copy running-config startup-config
```

What to do next

Reload the switch.

Applying the Default Template

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no hardware profile route resource service-template**	Applies the default template.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to apply the default template.

```
switch# configure terminal
switch(config)# no hardware profile route resource service-template
switch(config)# copy running-config startup-config
```

What to do next

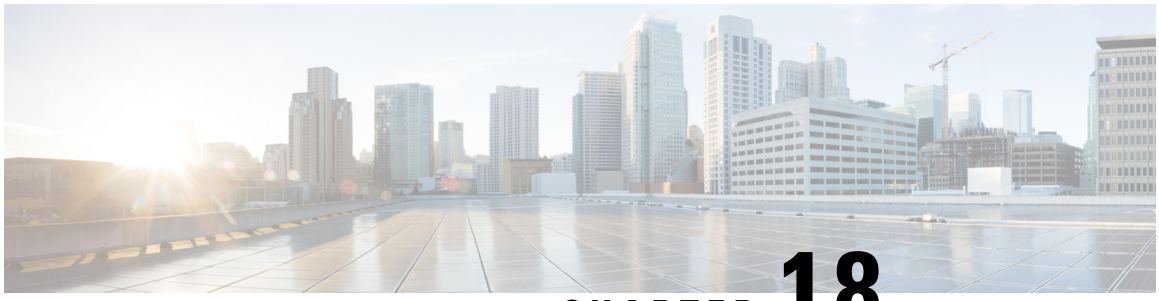
Rebooting the switch applies the default template (hrt-128-stm-128).

Verifying the MAC/ARP Hardware Resource Carving Template Configuration

To display MAC/ARP Hardware Resource Carving Template configuration information, enter one of the following commands:

Command	Purpose
show hardware profile route resource template	Displays all existing templates including the default.
show hardware profile route resource template *template-name*	Displays the details of a specific pre-defined template.

Command	Purpose
show hardware profile route resource template default	Displays the details of the default template.
show running-config hardware profile route resource template	Displays the running configuration information related to the template manager. Displays the currently applied non-default stm/hrt template. If the default template is applied, nothing is displayed here.
show startup-config hardware profile route resource template	Displays the startup configuration information related to the template manager. When entering the **copy running-config startup-config** command, the currently applied non-default stm/hrt template is displayed. If the default template is applied, nothing is displayed.



CHAPTER 18

Configuring VN-Segment

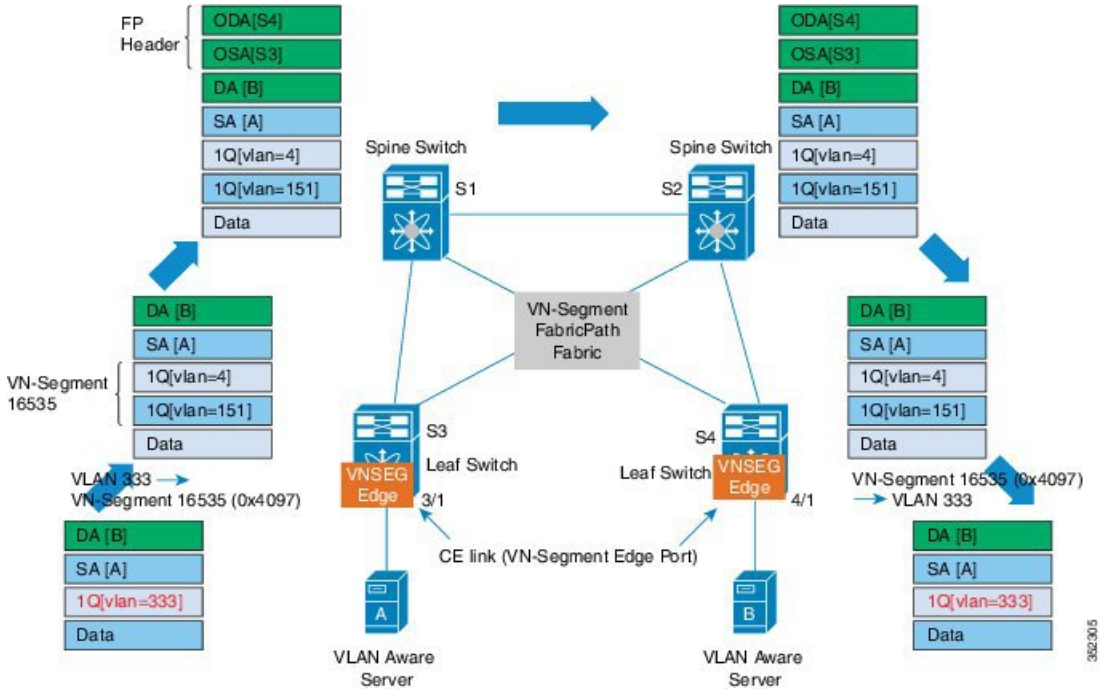
This chapter contains the following sections:

Information About VN-Segment

The VN-Segment feature defines a new way to "tag" packets on the wire replacing the traditional 802.1Q VLAN tag. This feature uses a 24-bit tag also referred to as a Virtual Network Identifier (VNI). CE links (access and trunk) carry traditional VLAN tagged/untagged frames. These are the VN-Segment Edge ports.

FabricPath links (**switchport mode fabricpath**) carry VN-Segment tagged frames for VLANs that have VNIs defined. These are the VN-Segment core ports.

Figure 28: VN-Segment and FabricPath



The previous figure shows a typical Cisco FabricPath network. Switches S1 and S2 are the spine switches. Switches S3 and S4 are the leaf switches and are connected to the spines over FabricPath interfaces. The VN-Segment feature is enabled on all leaf switches.

Server A is connected to leaf switch S3 and server B is connected to leaf switch S4 through normal Layer 2 trunk/access ports. These interfaces are also referred to as the "VNSeg Edge" ports. The servers send and receive traditional .1Q tagged or untagged frames. No new configurations are needed on the servers. The spines forward the VN-Segment tagged frames to the intended leafs.

Assume that servers A and B need to be in the same Layer 2 flood domain.

On the leaf switches, VLAN 333 is mapped to an available VN-Segment ID 16535. This VN-Segment ID identifies the VLAN 333 on the FabricPath network.

Here is a typical packet flow:

1. A data packet from server A to server B tagged with VLAN 333 is received on the VNSEG port of S3.
2. S3 does the packet lookup and sends the packet on the FabricPath port towards the spine. The switch S3 uses the VN-Segment ID corresponding to the VLAN.
3. S1 and S2 performs FabricPath forwarding towards the intended leafs.
4. S4 receives the VN-Segment ID tagged packet and performs packet lookups. Once the packet destination port is identified as a VNSEG edge port, S4 uses the VLAN ID corresponding to the VN-Segment ID in the packet and sends the packet.

Note If the VN-Segment ID to VLAN mapping does not exist, the packet is dropped.

5. Server B receives the .1Q data packet from Server A.

The same process is followed in the data packets from server B to server A.

Guidelines and Limitations for VN-Segment

VN-Segment has the following guidelines and limitations:

- The VN-Segment tag is added to traffic egressing FabricPath (FP) links only.
- Data forwarding semantics is the same as that of the VLANs.
- The devices must be VN-Segment aware with appropriate hardware support.
- Leaf switches must be configured for VN-Segment.
- The Virtual Network Identifier (VNI) is the network global ID, not the VLAN ID.
- Up to 4K VN-Segments and global VLANs are supported per leaf switch. There are only 4K VLANs.
- Different leafs can have different mapping to up to support 50K tenants on the fabric, depending on hardware and software limitations.
- If compatibility checks fail for the image, ISSD might be rejected .
- The VLAN-to-VN-Segment mapping must be consistent on the vPC+ peer switches for correct traffic flow. vPC type 1 consistency checks suspend VLANs on vPC peer swtiches with inconsistent mappings.

Enabling VN-Segment

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **install feature-set fabricpath**	Installs the FabricPath feature set on the switch.
Step 3	switch(config)# **feature-set fabricpath**	Enables the FabricPath feature set on the switch.
Step 4	switch(config)# **feature vn-segment-vlan-based**	Enables the VN-Segment feature on the switch.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable VN-Segment:

```
switch# configure terminal
switch(config)# install feature-set fabricpath
switch(config)# feature-set fabricpath
switch(config)# feature vn-segment-vlan-based

switch(config)# copy running-config startup-config
```

Configuring VN-Segment for a VLAN

Before you begin

The VN-Segment feature must be enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** *vlan-id*	Creates a VLAN.
Step 3	switch(config-vlan)# **mode fabricpath**	Configures the VLAN as a FabricPath VLAN. VN-Segments for a VLAN must be configured in FabricPath mode on the Leaf.
Step 4	switch(config-vlan)# **vn-segment** *segmentation-id*	Defines the network global ID. The *segmentation-id* range is from 4096 to 16,773,119.
Step 5	(Optional) switch(config-vlan)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure VN-Segment for VLAN:

```
switch# configure terminal
switch(config)# vlan 100
switch(config-vlan)# mode fabricpath
switch(config-vlan)# vn-segment 4096
```

Configuring VN-Segment for VLAN in Configure Sync

VN-Segments can be configured with the **configure sync** command for VPCs.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure sync**	Enter configuration sync mode.
Step 2	switch(config-sync)# **switch-profile test**	Creates a switch profile that contains a predetermined configuration.
Step 3	switch(config-sync-sp)# **vlan** *vlan-id*	Creates a VLAN.
Step 4	switch(config-sync-sp-vlan)# **vn-segment** *segmentation-id*	Defines the network global ID. The *segmentation-id* range is from 4096 to 16,773,119.
Step 5	(Optional) switch(config-sync-sp-vlan)# **commit**	Synchronizes the configuration with the peer switch and applies the configuration locally.
Step 6	switch(config-sync-sp-vlan)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure VN-Segment for a VLAN in configure sync mode:

```
switch# configure sync
switch(config-sync)# switch-profile test
Switch-Profile started, Profile ID is 1
switch(config-sync-sp)# vlan 3500
switch(config-sync-sp-vlan)# vn-segment 40001
switch(config-sync-sp-vlan)#
```

Configuring VN-Segment in Transit Mode

Before you begin

The FabricPath feature set must be enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **fabricpath mode transit**	Enables transit mode. You need to save the configuration and reload the spine. **Note** This command is disallowed if vn-segment-vlan-based is configured because they are mutually exclusive.

Example

The example shows how to configure VN-Segment in transit mode:

```
switch# configure terminal
switch(config)# fabricpath mode transit
Enabling transit mode. Please save configuration and reload.
```

What to do next

Enter the **show fabricpath mode** command to show the status of the mode.

Configuring VN-Segment in Non-Transit Mode

You need to enter the **feature vn-segment-vlan-based** command on the spine to enable the spine in non-transit mode.

Before you begin

The FabricPath feature set must be enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature vn-segment-vlan-based**	Enables a VLAN-based VN-Segment.
Step 3	switch(config)# **vni** *vni-id*	The range of *vni-id* is 4096 to 16,773,119.

Example

This example shows how to configure VN-Segment in non-transit mode:

```
switch# configure terminal
switch(config)# feature vn-segment-vlan-based
switch(config)# vni 16896
```

Disabling VN-Segment

Before you begin

VN-Segment configurations must be removed manually prior to disabling the feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature vn-segment-vlan-based**	Disables VN-Segment.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

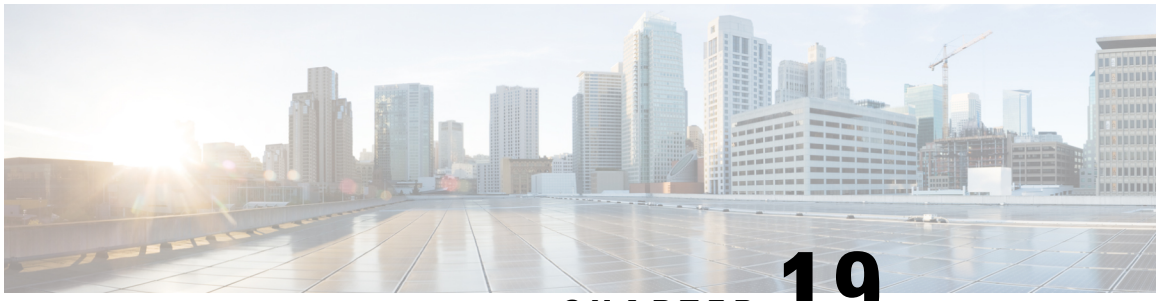
This example shows how to disable VN-Segment:

```
switch# configure terminal
switch(config)# no feature vn-segment-vlan-based
```

Verifying VN-Segment Configuration

Use the following commands to display VN- Segment configuration information:

Command	Purpose
show vlan id *vland-id-list* **vn-segment**	Displays the configured VLAN-to-VN-Segment mappings for the specified list of VLANs.
show vpc consistency-parameters global	Displays information on the number of VLANs and VN-Segment mappings on each VPC switch to help determine any mismatches.
show vpc consistency-parameters vlans	Displays information to identify the VLAN and VN-Segment configuration mismatches.



CHAPTER 19

Configuring VXLANs

This chapter contains the following sections:

Information About VXLAN

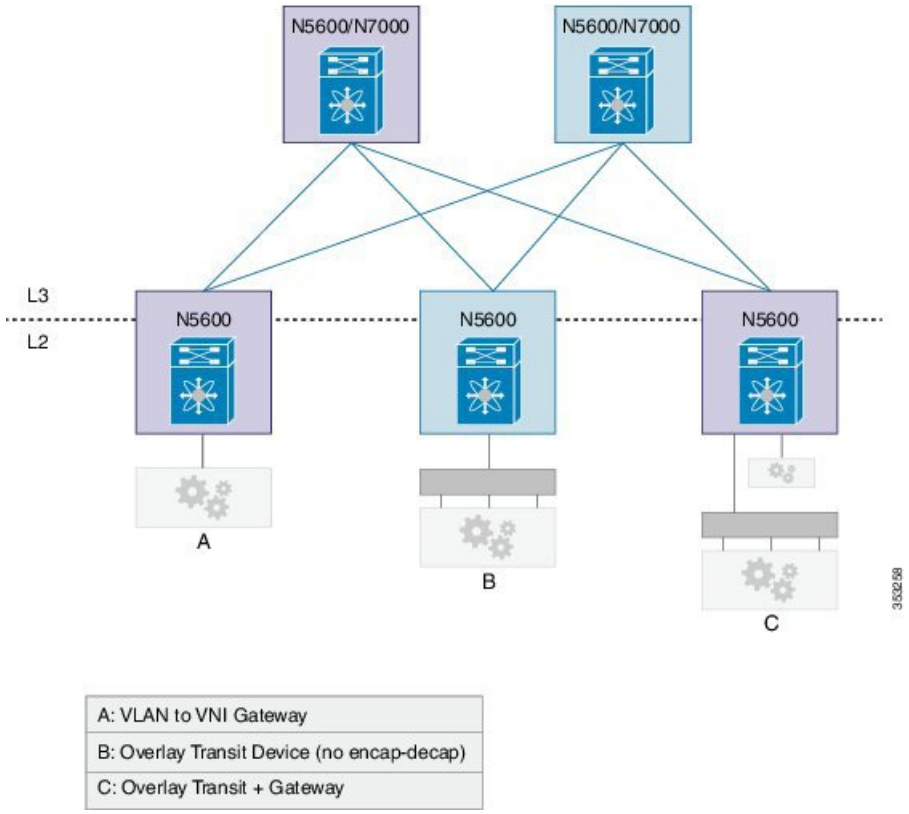
You can use Virtual Extensible Local Area Networks (VXLANs) to extend reachability of a VLAN within a data center over Layer 3. When you use VXLANs, you are no longer restricted to using only 4096 VLANs in a data center.

A Layer 2 VLAN is mapped into a larger (24-bit) ID VXLAN Network Identifier (VNI). All frames on that VLAN are encapsulated in an IP/UDP frame for transport. An additional VXLAN header is added to carry the VNI information. The VNI identifies the Layer 2 segment that the frame belongs to and is used to define a much larger Layer 2 broadcast domain for that frame. Typically, a Layer 2 domain (VLAN) confines the VM's mobility. With a VXLAN, the Layer 2 domain is extended throughout the data center, increasing the VM's mobility by extending the Layer 2 broadcast domain across Layer 3. The 24-bit VNI provides for about 16 million different Layer 2 segments that support a large number of tenants, and their VLANs, in a multitenant data center.

VXLAN Layer 2 Gateway

A VXLAN gateway is a device that encapsulates a classical Ethernet (CE) frame into a VXLAN frame and decapsulates a VXLAN frame into a CE frame. A gateway device transparently provides VXLAN benefits to the physical hosts and virtual machines. The physical hosts or VMs are completely unaware of VXLAN encapsulation. The gateway function can be implemented in a physical network device such as the Cisco Nexus 5600 Series Switch or a vSwitch such as the Cisco Nexus 1000V.

Figure 29: VXLAN Gateway Use Cases



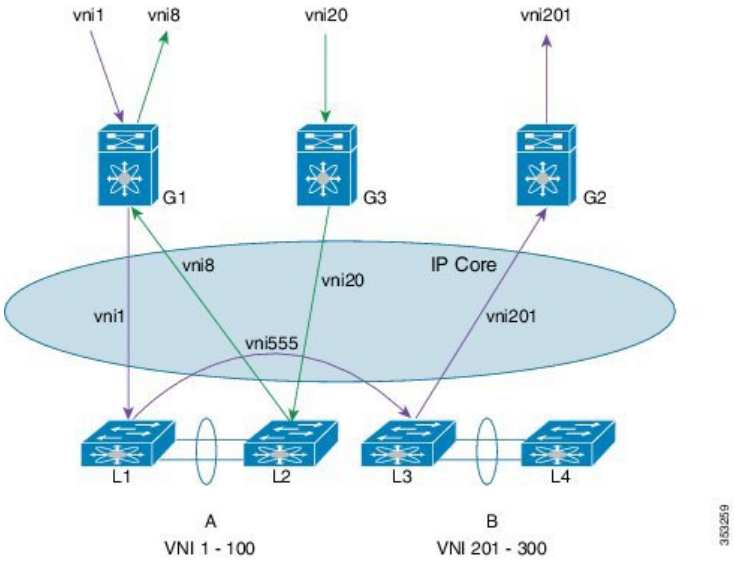
VXLAN Router

Similar to traditional routing between different VLANs, a VXLAN router is required for communication between devices that are in different VXLAN segments. The VXLAN router translates frames from one VNI to another. Depending on the source and destination, this process might require decapsulation and reencapsulation of a frame. The Cisco Nexus device supports all combinations of decapsulation, route, and encapsulation. The routing can also be done across native Layer 3 interfaces and VXLAN segments.

You can enable VXLAN routing at the aggregation layer or on Cisco Nexus device aggregation nodes. The spine only forwards based IP and ignores the encapsulated packets. To help scaling, a few leaf nodes (a pair of border leaves) perform routing between VNIs. A set of VNIs can be grouped into a virtual routing and forwarding (VRF) instance (tenant VRF) to enable routing among those VNIs. If routing must be enabled among a large number of VNIs, you might need to split the VNIs between several VXLAN routers. Each router is responsible for a set of VNIs and a respective subnet. Redundancy is achieved with FHRP.

The following figure shows a configuration example with two Cisco Nexus leaf nodes (each node is a virtual port channel [vPC] pair) that acts as VXLAN routers. Node A routes VNIs 1 to 100 while node B routes VNIs 201 to 300. You must configure a separate VNI (555) per tenant VRF to carry traffic between VXLAN routers and for routing protocols to exchange routing information between the VXLAN routers.

Figure 30: VXLAN Router Configuration



The figure shows two flows. vni-1 to vni-201 and vni-20 to vni-8.

1. vni-1 to vni-201 : The packet in vNI1 at G1 is sent to the default router for vni-1 (L1 and L2). The router finds that the destination address is in vni-201 which is reachable over interface vni-555. The packet is encapsulated with vni-555 and sent to the L3 and L4 pair. The router pair (L3 and L4) routes the packet from vni-555 to vni-201 where the final destination is reachable. The packet is then sent to G2, which uses vni-201 to be delivered to the final destination. This packet takes two router hops.
2. vni-20 to > vni-8: The packet at G3 in vni-20 is sent to the default router (L1 and L2). The final destination is reachable on vni-8. Router (L1 and L2) reencapsulates the packet with vni-8 and sends it to G1 where the final destination resides.

Any packet that originates in vni 1 to 100, but is destined to go outside of its VNI, must come to node A to get routed. Similarly, any packet delivered to vni 201 to 300 whose source is different from the destination VNI is routed into its destination VNI on node B. Packets from vni-1 to vni-201 take two hops (the first hop on node A and the second on node B).

The traffic that is routed between a VNI and outside (nonvirtualized) world might have to go through an external router that is connected to the VXLAN router. This router might need to provide Network Address Translation (NAT) and firewall services as well.

The VXLAN routers can use any routing protocol, for example Open Shortest Path First (OSPF), for routing within the tenant VRF. The routers must form neighbor adjacencies over the transit-VNI, because the tenant VRFs are not visible in the core. The core routers only know about the underlay VRF that is used for routing the packets between VXLAN Tunnel Endpoints (VTEPs) that are based on the outer header.

VXLAN Overlay Network for Broadcast/Unknown-Unicast/Multicast Overlay Traffic

All broadcast/unknown-unicast/multicast overlay traffic must be sent to multiple VTEPs. To identify all the VTEPs that are interested in traffic for a specific VNI, VTEPs build a multicast tree which is identified as the VXLAN Overlay Network for each VNI. This is achieved by mapping the VNI to a multicast group on all the VTEPs that are interested in the VNI. A multicast tree is built using the PIM protocol and all non-unicast traffic is distributed to all the interested VTEPs that join the multicast tree. This is achieved by mapping any given VNI to a multicast group address, which is also called the Delivery Group (DG) for that VNI. When

VTEP sends a non-unicast packet on a VNI over the overlay network, the packet is encapsulated in a VXLAN header and is sent to the DG address instead of sending it to single destination VTEP IP address as in the case of unicast traffic. The VXLAN encapsulated packets destined to the DG get routed in the overlay network by using the PIM tree built for the DG . All the VTEPs that join the PIM tree built for that DG receive the traffic.

Cisco Nexus devices use PIM BIDIR only to build this VXLAN Overlay Network. PIM ASM/SSM is not supported currently, so any multicast group defined as DG to carry VXLAN overlay traffic for a VNI must always be defined as a BIDIR group. The rendezvous point (RP) for this BIDIR group can be anywhere in the Layer 3 overlay network. Multiple VNIs can map to the same DG, and so the overlay traffic for these VNIs is sent across the Overlay Network using the same PIM BIDIR tree. Cisco Nexus devices can support a maximum of 200 DGs on a given VTEP.

VXLAN Multicast Routing

You can configure the VXLAN router as a multicast router for inner (user) multicast groups. Multicast routing must be configured within a tenant VRF. The multicast routing protocol for the inner groups does not have to be PIM BIDIR even though PIM BIDIR is used for the outer multicast. The inner multicast group can use PIM-Any Source Multicast (ASM), ASM, or BIDIR as supported by the platform. If VTEP is a part of a vPC pair, the inner group cannot be a BIDIR group. In a vPC setup, BIDIR can be used only as a DG to build the VXLAN overlay network and cannot be used to carry inner multicast traffic. Similar to VXLAN unicast routing, multicast routing is done among the VNI interfaces that are in a tenant VRF. The VXLAN gateway nodes deliver the multicast data and control frames to the VXLAN multicast router using an outer delivery group (DG).

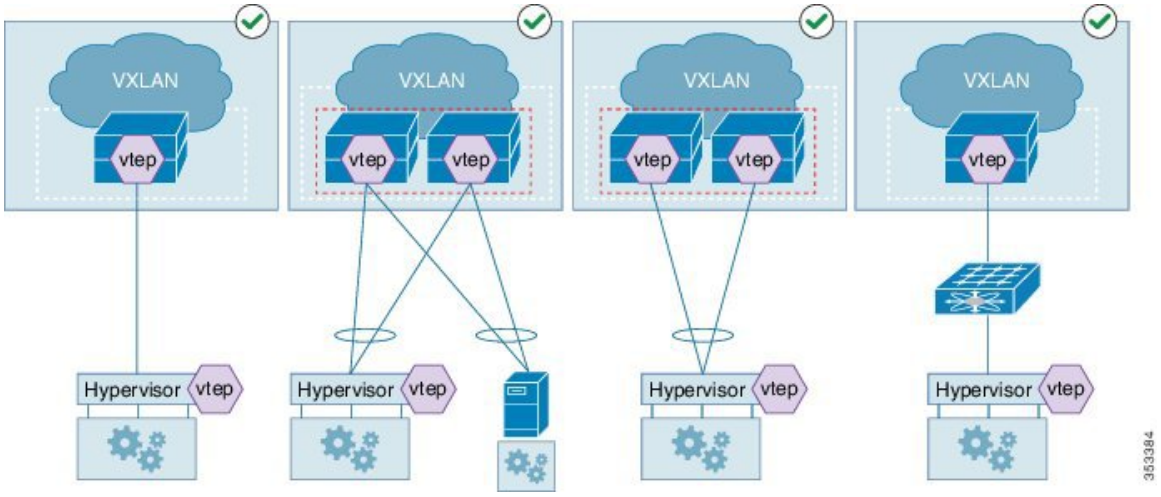
PIM routers for the inner multicast group exchange the PIM messages over a VXLAN network that connects them on all VNIs that are part of the tenant VRF.

Cisco Nexus Device Overlays

The following figure shows a topology with a virtual port channel (vPC), fabric extenders (FEXes), VXLAN hypervisors, and gateway ports that are supported by the Cisco Nexus device. All FEX topologies (AA-FEX, ST-FEX, and 2LvPC) are supported.

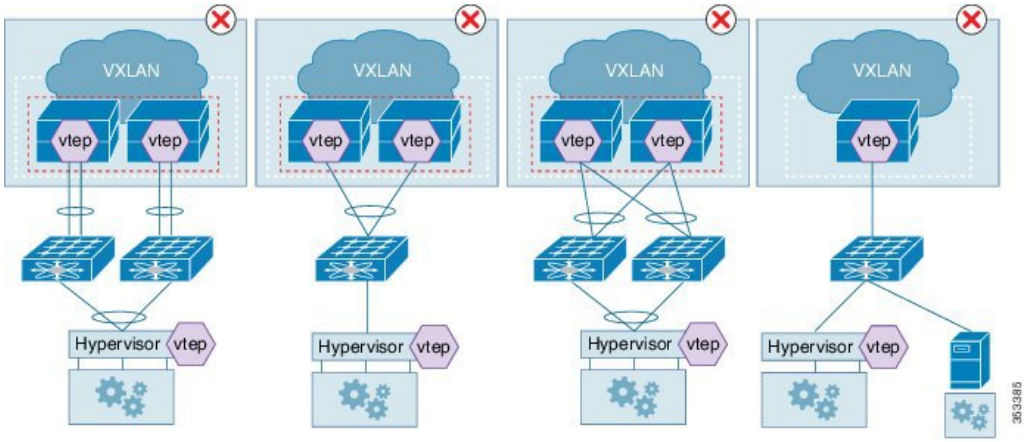
The figure below shows the supported topologies. A VXLAN Tunnel Endpoint (VTEP) hypervisor can be connected through switch vPC as shown in diagrams 2 and 3. Diagram 4 shows that the hypervisor can be connected through Straight-through (ST) FEX (without vPC).

Figure 31: Supported Topologies— Hypervisor directly connected to a VXLAN switch and Hypervisor behind switch vPC



The figure below shows the unsupported topologies. A VXLAN Tunnel Endpoint (VTEP) hypervisor cannot be connected through FEX vPC configurations—ST-FEX vPC, AA-FEX, and 2LVPC, as shown in the first three diagrams. Diagram 4 shows that mixing of overlay and non-overlay devices is not supported on the same fabric extender (FEX).

Figure 32: Unsupported Topologies—(i) Hypervisor behind Straight-through (ST)-FEX vPC (ii) Hypervisor behind Active-Active (AA)-FEX, and (iii) Hypervisor behind two-layer vPC (2LvPC)



VXLAN Tunnel Endpoint

A VXLAN Tunnel Endpoint (VTEP) performs the VXLAN gateway function. A VTEP is represented as an interface in the Cisco NX-OS. All VTEPs are managed by the VXLAN manager. The Cisco Nexus device requires one VTEP for each encapsulation type.

VTEP IP Addresses and VRF Instances

Each VTEP must have at least one IP address. This IP address is used for encapsulation or decapsulation. For vPC configurations, a separate IP address is used for encapsulation or decapsulation of the traffic to and from vPC connected hosts. The emulated IP address must be the same on both switches in a vPC pair. The emulated IP address allows the network to load balance the traffic destined to the vPC-connected devices without using MCT. Similarly, a distinct non-emulated IP address that is used for encapsulation or decapsulation for a singly connected host ensures that traffic to that host arrives on the correct switch in the pair without going through a vPC Peer-Link, also known as Multichassis EtherChannel Trunk (MCT).

The VRF instance specified for the VTEP carries all the encapsulated traffic within the data center.

The Cisco Nexus device supports a single infrastructure (infra)-VRF and multiple tenant VRFs. The infra-VRF carries the VXLAN traffic through the core Layer 3 network. A tenant VRF is not visible to the routing devices in the core. The tenant VRFs are used by VXLAN routers. The Cisco Nexus device supports the default VRF as the infra-VRF.

VTEP IP Multicast Addresses

A VXLAN gateway uses an IP delivery group (DG) to flood multidestination frames within a VNI. Layer 2 broadcast, unknown unicast, and multicast frames are flooded to other VTEPs using the IP multicast DG address. Only one flood-DG address can be used per VNI. To reduce the amount of BUM traffic that reaches all VTEPs, each VNI should be given its own DG address so that the flood domain is contained within the VTEPs that are a gateway for the VNI. The number of VNIs might exceed the distinct DG trees that can be supported by the network. In that case, multiple VNIs must share a DG address for flooding. The user (inner or overlay) multicast frames are also encapsulated using a DG.

VXLAN Tunnel Endpoint Peers

VTEP-Peer Learning

The Cisco Nexus device discovers VXLAN Tunnel Endpoint Peers (VTEPs) using the flood-and-learn technique which is when a VTEP peer is learned when the first VXLAN encapsulated packet is received from the peer.

A gateway device must identify only those VTEP peers that support any of the locally configured VNIs or delivery groups (DG).

The Cisco Nexus device has the capability to snoop unicast, as well as, multicast packets sent by unknown peers. If an unknown VTEP-peer sends packets using any of the multicast DGs configured locally, a notification is received from the hardware, which provides the information about the new peer. In addition to monitoring the multicast DG addresses, the Cisco Nexus device also monitors frames sent to its own VTEP addresses. The multicast and unicast frames snooped by the hardware are not de-capsulated until the sender is a known VTEP-peer.

The VXLAN manager adds the sender VTEP as a new peer. After the VTEP peer is added in the hardware, the hardware would then stop sending the VTEP peer discovery notification for it.

Due to the sharing of DG addresses, the VNI in the packet might not be configured as a gateway VNI. In that case, the VTEP peer avoids further VTEP peer discovery indications.

VTEP-Peer Aging/Removal

A VTEP-peer might shut down, be removed from the network, become unreachable, or just become dormant. In many situations, there is no direct indication to remove the VTEP-peer. Therefore, you must employ an aging mechanism to clean up the VTEP peers that were dynamically learned. The cleanup is essential because the total number of active VTEP peers present at any given time is limited by the hardware. The ageout time is set to 10 minutes.

vPC Considerations

vPC Consistency Checks

Parameter	vPC Check Type	Description
VLAN-VNI mapping	Type-1—nongraceful	Brings down the affected VLANs on vPC ports on both sides.
VTEP-Member-VNI	Type-1—nongraceful	Member VNIs must be the same on both nodes. VNIs that are not common bring down the corresponding VLANs on vPC ports on both sides.
VTEP-emulated IP	Type-1—graceful	If an emulated IP address is not the same on both nodes, all gateway vPC ports on one side (secondary) are brought down. Alternatively, one side of all vPC ports is brought down.
VTEP-node IP address	Type 2	vPC manager issues a warning.

vPC and Multicast

For each outer destination group (DG), you must select one of the vPC peers as a designated Affinity Forwarder (AF). The AF switch forwards the multidestination traffic to the vPC connected devices while a non-AF switch only forwards traffic to singly connected devices. The selection of an AF is done by a multicast group that is based on a vPC permanent role.

QoS/ACL Support

Quality of Service (QoS) and Access Control Lists (ACLs) are applied to the ingress packets for packets from VLAN to VXLAN (encapsulation). During encapsulation, the outer Class of Service (CoS) and differentiated services code point (DSCP) values are derived from the final inner COS and DSCP values. When a packet is decapsulated, the outer CoS is used as the inner CoS, because there is no inner .1Q, or .1P tag carried with the inner frame. The rest of the processing is done on the inner frame.

If traffic is decapsulated and reencapsulated, the inner CoS value is used to derive the outer DSCP value. The CoS is preserved from the ingress frame.

For overlay transit traffic (traffic that is not decapsulated), QoS and ACLs are applied to the outer headers.

TTL Handling

When a native classical Ethernet (CE) packet is encapsulated, the outer Time To Live (TTL) is selected based on a configured value. The default is 32. The outer TTL is decremented based on the outer IP routing and discarded when it goes to zero. The inner TTL is unchanged as the packet traverses the overlay network. After decapsulation, the inner TTL is preserved if the inner packet is Layer 2 switched. The inner TTL is decremented whenever an inner packet is routed.

When a multicast packet is decapsulated and reencapsulated, the outer TTL is decremented by 1 while the inner TTL is preserved. If the inner packet is multicast routed, the inner TTL is decremented whenever an unencapsulated inner packet is delivered to the end station.

Multipathing Support

When a CE packet is encapsulated using VXLAN encapsulation, a 16-bit hash value is created using the Layer 2 and Layer 3 addresses and Layer 4 source and destination ports if available. The hash value is then used as an outer UDP src_port. This hash value represents the inner-packet flow (with some aliasing due to the 16-bit hash result). The outer UDP source port is used by core routers to load balance traffic between two VTEPs based on inner flows.

When the packet is first encapsulated, inner packet headers are used to select one of many available equal cost paths to the destination VTEP.

MTU

The Cisco Nexus device does not support fragmentation or re-assembly of VXLAN traffic. As VXLAN encapsulation adds 50 bytes to the packet, the MTU of the tenant devices must be at least 50 bytes smaller than the MTU of the network devices. The Cisco VXLAN device supports an MTU configuration on a physical interface as well as an SVI interface. Ensure that the MTU on the VNI-mapped SVI is 50 bytes smaller than the physical interfaces's MTU when configuring VXLAN routing. For a VXLAN Layer 2 gateway, the default MTU is 1500. The recommended method is to increase the MTU to 1550.

Guidelines and Limitations for VXLAN

The VXLAN configuration guidelines and limitations are as follows:

- A VXLAN device must be configured in the store-and-forward mode.
- The classical Ethernet (CE) packet on an edge interface is mapped to a Virtual Network Identifier (VNI) based on the VLAN to which it is associated. The VLAN to VNI mapping is created under the VLAN configuration, which limits the number of supported VNIs on a switch to 4000.
- The multicast delivery group used to build the VXLAN overlay network for VNIs must be configured as a Protocol-Independent Multicast (PIM) Bidirectional (BIDIR) group. The VXLAN overlay network cannot be built using PIM SM or PIM SSM.
- PIM-BDIR in a vPC configuration for non-VXLAN traffic is not supported.
- The Cisco Nexus device does not support Layer 3 links on southbound interfaces that are connected to a fabric extender (FEX).

- Only loopback interfaces are supported as the source interface for the NVE interface under an Network Virtualization Edge (NVE) configuration. NVE is equal to VTEP.
- For any protocols that work over inner switched virtual interfaces (SVIs), you should increase the maximum transmission unit (MTU) of that SVI by 50 to allow VXLAN encapsulation. If you use the default MTU, you might get unexpected results.
- A VXLAN Tunnel Endpoint (VTEP) hypervisor cannot be connected through Straight-Through FEX (ST-FEX-VPC), Active-Active FEX (AA-FEX), and 2-Layer vPC.
- The Cisco Nexus device can only support Layer 3 routed port links to carry overlay traffic to the core.
- A Layer 2 trunk cannot be used to carry overlay traffic to the core. Layer 2 trunks with SVIs can be used on southbound interfaces that connect to hypervisors. The overlay traffic that originates to and from hypervisors is carried using an SVI.
- The IP routing protocol must be configured for the underlay network.
- PIM-BIDIR multicast routing must be configured for the underlay network.
- The vn-segment-vlan-based feature must be configured on the VXLAN gateway and router devices.
- IGMP snooping is not supported on VXLAN VLANs.
- Hypervisor VTEPs (such as Cisco Nexus 1000V) cannot be connected using Layer 3 interfaces. They must be connected through Layer 2 interfaces.
- Only one NVE interface is supported on a switch.
- SNMP is not supported on the NVE interface.
- Policy-based routing (PBR) is not supported for tenant traffic.
- Ingress and egress ACLs cannot be applied to the outer header of the VXLAN packet on the VXLAN gateway device.
- A physical port cannot be used as a tenant (gateway) port and overlay port at the same time. Mix of VLANs with and without a VNI is not supported on the same trunk interface.

Note

- Gateway Port—Physical port on which VLAN-VNI mapping is configured.
- Overlay Port—Encapsulated traffic is received and sent on an overlay port. This includes the core (network) facing ports as well as local edge ports where VTEPs (hypervisors) are connected.

- The maximum transmission unit (MTU) must configured throughout the network to accommodate 50 bytes of VXLAN encapsulation.
- Tenant ports and overlay ports that connect to VTEP hypervisors cannot be on the same fabric extender (FEX).
- When you are connecting VTEP hypervisors to FEX ports, all VTEP hypervisors that are connected to a FEX must use the same outer VLAN.

- When a device is running in VXLAN flood and learn mode and packets reach decapsulate VXLAN tunnel endpoint (VTEP) as unicast, after decapsulation, the destination MAC is not known to the decapsualte VTEP and packets are dropped. This helps to avoid flooding back to the core.

  Clear the affected MAC on remote VTEP or stop topology change notifications to avoid traffic loss. This issue is not seen when VXLAN is running in EVPN mode.

- Refer to supported and un-supported topology diagrams when connecting hosts and VTEP hypervisors to a Cisco Nexus device.
- Configured store and forward mode with reload.
- Connecting hypervisors with different overlay encapsulation to the same FEX is not allowed.
- VLAN 1 cannot be used to carry VXLAN traffic.
- There is no support for originating Hot Standby Router Protocol (HSRP) packets with the source MAC as a user-configured HSRP MAC. Support is limited to using standard HSRP MAC addresses (v1 and v2) as the source MAC addresses for HSRP packets.
- The **show interface nve 1 counters** command does not display statistics of VXLAN incoming and outgoing packets.
- DHCP snooping on VXLAN-enabled VLANs is not supported.
- A non-VNI enabled VLAN, with an SVI in the same VRF as the underlay interfaces, is considered as an overlay port. If this VLAN is configured on a FEX HIF, VXLAN encapsulated traffic will egress this port. To avoid this, the non-VNI enabled VLANs should be configured in a VRF that is separate from the VRF that the underlay interfaces belong to.
- When you perform a disruptive upgrade from Cisco NX-OS release 7.0.x to 7.1.x, 7.2.x, or 7.3.x, with the **hardware ethernet store-and-fwd-switching** command configured, there might be some traffic loss. To avoid the above scenario, we recommend that you create a /mnt/pss/qd_sf_sdb file with content as 1 before upgrading. If you have upgraded from Cisco NX-OS release 7.0.x to 7.1.x, 7.2.x, or 7.3.x, with the **hardware ethernet store-and-fwd-switching** command configured, after the upgrade, remove the **hardware ethernet store-and-fwd-switching** command configuration, reconfigure the command again, and reload the switch.

vPC Considerations

- A virtual IP must be configured for the vPC pair
- A virtual IP must be configured for loopback purposes.
- A peer-link switched virtual interfaces (SVI) must be only on a peer-link in external communication. A configuration example:

```
vpc nve peer-link-vlan 99
interface vlan99
no shutdown
no ip redirects
ip address 99.1.1.1/24
ip ospf cost 10
```

```
ip router ospf 1 area 0.0.0.0
ip pim sparse-mode
```

- A special peer-link SVI must be configured on the VPC pair.
- VPC peers must have identical configurations:
  - Consistent VLAN to VN-segment mapping.
  - Consistent NVE1 binding to the same loopback interface.
    - Using the same secondary IP address.
    - Using different primary IP addresses.
  - Consistent VNI to group mapping.
- A VTEP hypervisor cannot be connected to AA-FEX, EVPC, or ST-FEX vPC.
- Supports a line-rate encapsulation or decapsulation of VXLAN switched traffic.

Note VXLAN introduces a 50-byte overhead to the original packet due to VXLAN encapsulation. For example, for a 1000 byte packet, there is a 5% overhead per packet. Overhead varies depending on the packet size and it is expected for VXLAN.

VTP Considerations

On fabric path or EVPN, when feature VTP is enabled and there is a switch reload, the auto-configured VLAN profile information is not saved in the running or start up configuration. We recommend not to enable feature VTP on fabric path or EVPN leaf nodes to avoid getting into VLAN auto-configuration issues.

Enabling VXLAN

Before you begin

You must configure underlay and PIM-bidir multicast.

Configure the switch in the store-and-forward mode. See Configuring a Switch in the Store-and-Forward Mode, on page 271.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature nv overlay**	Enables NV overlay.

	Command or Action	Purpose
Step 3	switch(config)# **feature vn-segment-vlan-based**	Enables the VN-Segment feature on the switch.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable VXLAN:

```
switch# configure terminal
switch(config)# feature nv overlay
switch(config)# feature vn-segment-vlan-based
```

Configuring a VNI

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** *vlan-id*	Creates a VLAN.
Step 3	switch(config)# **vn-segment** *vni-id*	Associates the access VLAN with the VNI. The *vni-id* range is from 4096 to 16773119.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a VNI:

```
switch# configure terminal
switch(config)# vlan 1001
switch(config)# vn-segment 8000
```

Configuring a Network Virtualization Endpoint Interface

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface nve** *nve-id*	Creates the NVE.
Step 3	switch(config-if-nve)# **source interface** *src-if*	Determines the source interface.
Step 4	switch(config-if-nve)# **member vni** *range* **mcast-group**	Assigns a multicast group for BUM traffic.
Step 5	switch(config-if-nve)# **no shutdown**	Returns the interface to its default operational state.
Step 6	switch(config-if-nve)# **copy running-config startup-config**	Saves the changes persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This examples shows how to configure a network virtualization interface:

```
switch# configure terminal
switch(config)# interface nve 1
switch(config-if-nve)# source-interface loopback 0
switch(config-if-nve)# member vni 21000 mcast-group 239.3.5.1
switch(config-if-nve)# no shutdown
switch(config-if-nve)# copy running-config startup-config
```

Configuring a Switch in the Store-and-Forward Mode

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware ethernet store-and-fwd-switching**	Enables store-and-foward switching.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a switch in the store-and-forward mode:

```
switch# configure terminal
switch(config)# hardware ethernet store-and-fwd-switching
switch(config)# copy running-config startup-config
```

What to do next

Switch must now be reloaded.

Disabling VXLAN

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature nv overlay**	Disables NV overlay.
Step 3	switch(config)# **no feature vn-segment-vlan-based**	Disables VLAN based VN segment.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to disable a VXLAN:

```
switch# configure terminal
switch(config)# no feature nv overlay
switch(config)# no feature vn-segment-vlan-based
```

Verifying VXLAN Configuration

Use one of the following commands to verify the configuration:

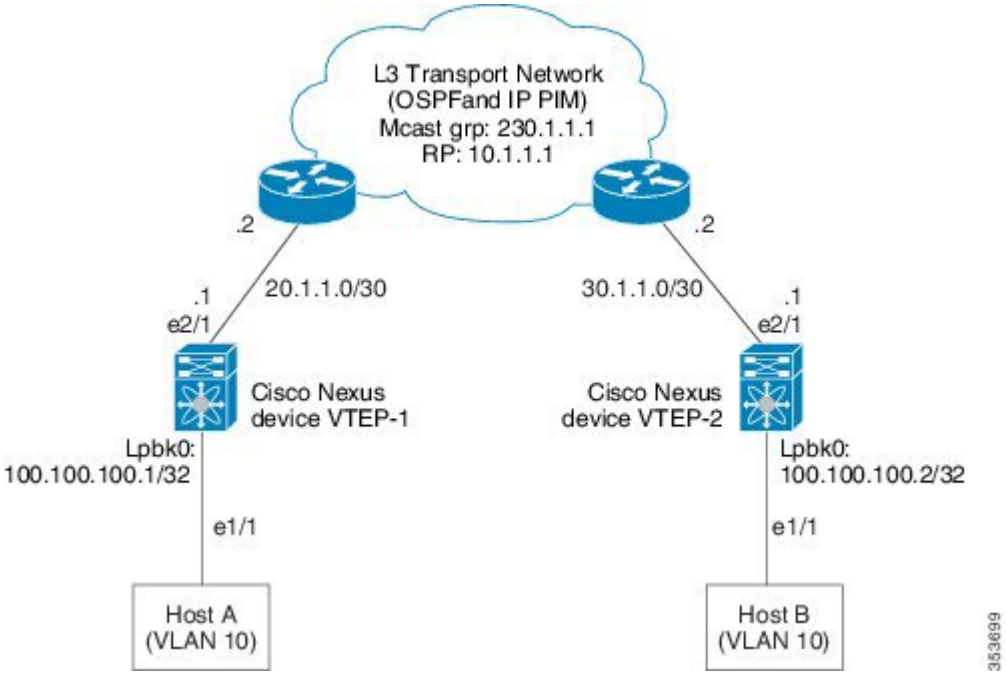
Command	Purpose
switch# **show interface nve** *id*	Displays details of the NVE interface.

Command	Purpose
switch# **show platform fwm info nve peer** [**all**]	Displays a list of NVE peers detected by using their IP address.
switch# **show mac address-table nve** [**count**] [*encap_type*]	Displays MAC addresses behind NVE peers.
switch# **show vlan counters**	Displays packet counters for a VLAN.
switch# **show nve peer**	Displays a list of discovered peers participating in the same VNIs.
switch# **show nve vni**	Displays a list of the configured VNIs.
switch# **show platform fwm info nve vni**	Displays a list of configured VNIs.
switch# **show nve conflict all**	Displays conflicts due to misconfiguration.
switch# **show run | grep "vpc nve"**	
switch# **show platform fwm info global | grep -i "NVE peer"**	

Example of VXLAN Bridging Configuration

An example of loopback interface configuration and routing protocol configuration:

Figure 33:



Cisco Nexus device VTEP-1 configuration:

```
switch-vtep-1(config)# feature ospf
switch-vtep-1(config)# feature pim
switch-vtep-1(config)# router ospf 1
switch-vtep-1(config-router)# router-id 100.100.100.1
switch-vtep-1(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4 bidir
switch-vtep-1(config)# interface loopback0
switch-vtep-1(config-if)# ip address 100.100.100.1/32
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# interface e2/1
switch-vtep-1(config-if)# ip address 20.1.1.1/30
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# feature nv overlay
switch-vtep-1(config)# feature vn-segment-vlan-based
switch-vtep-1(config)# interface e1/1
switch-vtep-1(config-if)# switchport
switch-vtep-1(config-if)# switchport access vlan 10
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config)# interface nve1
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config-if)# source-interface loopback0

switch-vtep-1(config-if)# member vni 10000 mcast-group 230.1.1.1
switch-vtep-1(config)# vlan 10
switch-vtep-1(config-vlan)# vn-segment 10000
switch-vtep-1(config-vlan)# exit
```

Cisco Nexus device VTEP-2 configuration:

```
switch-vtep-2(config)# feature ospf
```

```
switch-vtep-2(config)# feature pim
switch-vtep-2(config)# router ospf 1
switch-vtep-2(config-router)# router-id 100.100.100.2
switch-vtep-2(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4 bidir
switch-vtep-2(config)# interface loopback0
switch-vtep-2(config-if)# ip address 100.100.100.2/32
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# interface e2/1
switch-vtep-2(config-if)# ip address 30.1.1.1/30
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# feature nv overlay
switch-vtep-2(config)# feature vn-segment-vlan-based
switch-vtep-2(config)# interface e1/1
switch-vtep-2(config-if)# switchport
switch-vtep-2(config-if)# switchport access vlan 10
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config)# interface nve1
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# source-interface loopback0

switch-vtep-2(config-if)# member vni 10000 mcast-group 230.1.1.1
switch-vtep-2(config)# vlan 10
switch-vtep-2(config-vlan)# vn-segment 10000
switch-vtep-2(config-vlan)# exit
```

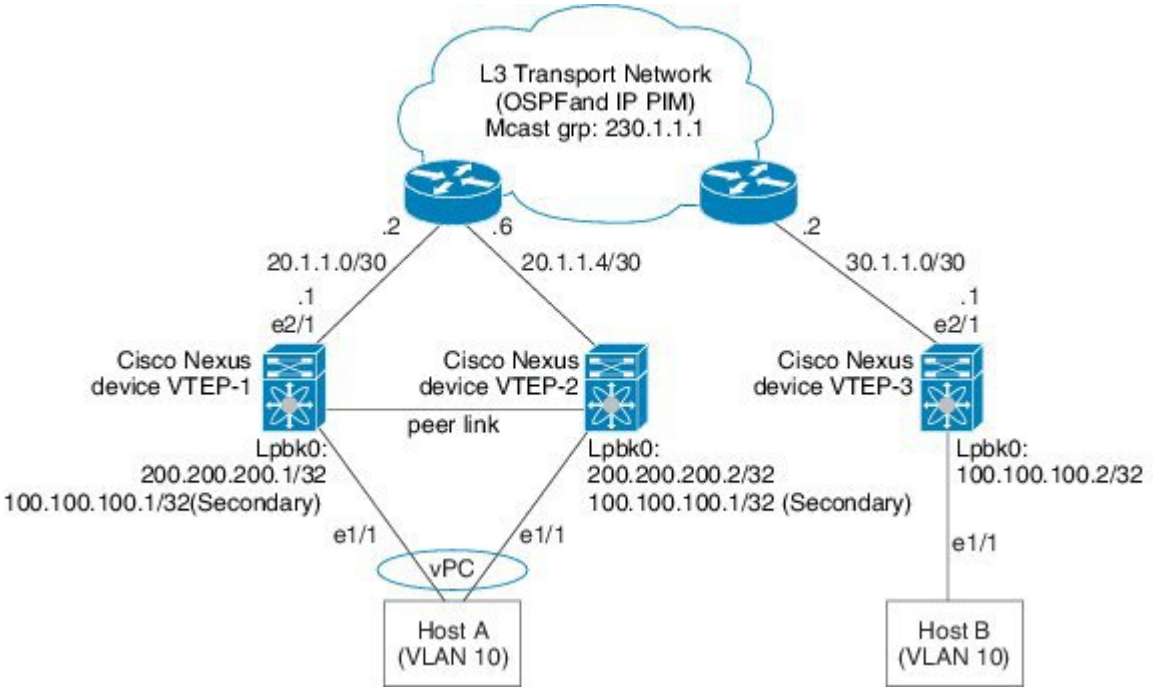
An example of the results of a VXLAN configuration:

```
switch(config)# show nve vni

Interface          VNI        Multicast-group   VNI State
-----------------  --------   ---------------   ---------
nve1               10000      230.1.1.1          up

switch(config)# show nve peers

Interface          Peer-IP           VNI        Up Time
-----------------  ---------------   --------   -----------
nve1               100.100.100.2     10000      06:13:07

switch(config)# show mac address-table

Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*  100     0000.bb01.0001   dynamic  0         F      F    nve1
*  100     0000.bb01.0002   dynamic  0         F      F    nve1
*  100     0000.bb01.0003   dynamic  0         F      F    nve1
*  100     0000.bb01.0004   dynamic  0         F      F    nve1
*  100     0000.bb01.0005   dynamic  0         F      F    nve1
*  100     0000.bb01.0006   dynamic  0         F      F    nve1
```

For a vPC VTEP configuration, the loopback address requires a secondary IP.

An example of a vPC VTEP configuration:

Figure 34:



Cisco Nexus device VTEP-1 configuration:

```
switch-vtep-1(config)# feature nv overlay
switch-vtep-1(config)# feature vn-segment-vlan-based
switch-vtep-1(config)# feature ospf
switch-vtep-1(config)# feature pim
switch-vtep-1(config)# router ospf 1
switch-vtep-1(config-router)# router-id 200.200.200.1
switch-vtep-1(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4 bidir
switch-vtep-1(config)# interface loopback0
switch-vtep-1(config-if)# ip address 200.200.200.1/32
switch-vtep-1(config-if)# ip address 100.100.100.1/32 secondary
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# interface e2/1
switch-vtep-1(config-if)# ip address 20.1.1.1/30
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# interface port-channel 10
switch-vtep-1(config-if)# vpc 10
switch-vtep-1(config-if)# switchport
switch-vtep-1(config-if)# switchport mode access
switch-vtep-1(config-if)# switchport access vlan 10
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config)# interface e1/1
switch-vtep-1(config)# channel-group 10 mode active
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config-if)# interface nve1
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config-if)# source-interface loopback0

switch-vtep-1(config-if)# member vni 10000 mcast-group 230.1.1.1
```

```
switch-vtep-1(config)# vlan 10
switch-vtep-1(config-vlan)# vn-segment 10000
switch-vtep-1(config-vlan)# exit

switch-vtep-1(config)#vpc nve peer-link-vlan 99
interface Vlan99
  no shutdown
  no ip redirects
  ip address 99.1.1.1/24
  ip ospf cost 10
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

Cisco Nexus device VTEP-2 configuration:

```
switch-vtep-2(config)# feature nv overlay
switch-vtep-2(config)# feature vn-segment-vlan-based
switch-vtep-2(config)# feature ospf
switch-vtep-2(config)# feature pim
switch-vtep-2(config)# router ospf 1
switch-vtep-2(config-router)# router-id 200.200.200.2
switch-vtep-2(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4 bidir
switch-vtep-2(config)# interface loopback0
switch-vtep-2(config-if)# ip address 200.200.200.2/32
switch-vtep-2(config-if)# ip address 100.100.100.1/32 secondary
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# interface e2/1
switch-vtep-2(config-if)# ip address 20.1.1.5/30
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# interface port-channel 10
switch-vtep-2(config-if)# vpc 10
switch-vtep-2(config-if)# switchport
switch-vtep-2(config-if)# switchport mode access
switch-vtep-2(config-if)# switchport access vlan 10
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config)# interface e1/1
switch-vtep-2(config)# channel-group 10 mode active
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# interface nve1
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# source-interface loopback0

switch-vtep-2(config-if)# member vni 10000 mcast-group 230.1.1.1
switch-vtep-2(config)# vlan 10
switch-vtep-2(config-vlan)# vn-segment 10000
switch-vtep-2(config-vlan)# exit

switch-vtep-2(config)#vpc nve peer-link-vlan 99
interface Vlan99
  no shutdown
  no ip redirects
  ip address 99.1.1.2/24
  ip ospf cost 10
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

Cisco Nexus device VTEP-3 configuration:

```
switch-vtep-2(config)# feature nv overlay
switch-vtep-2(config)# feature vn-segment-vlan-based
```

```
switch-vtep-2(config)# feature ospf
switch-vtep-2(config)# feature pim
switch-vtep-2(config)# router ospf 1
switch-vtep-2(config-router)# router-id 100.100.100.2
switch-vtep-2(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4 bidir
switch-vtep-2(config)# interface loopback0
switch-vtep-2(config-if)# ip address 100.100.100.2/32
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# interface e2/1
switch-vtep-2(config-if)# ip address 30.1.1.1/30
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# interface e1/1
switch-vtep-2(config-if)# switchport
switch-vtep-2(config-if)# switchport mode access
switch-vtep-2(config-if)# switchport access vlan 10
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config)# interface nve1
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# source-interface loopback0

switch-vtep-2(config-if)# member vni 10000 mcast-group 230.1.1.1
switch-vtep-2(config)# vlan 10
switch-vtep-2(config-vlan)# vn-segment 10000
switch-vtep-2(config-vlan)# exit
```

Note The secondary IP is used by the emulated VTEP for VXLAN.

Note Ensure that all configurations are identical between the VPC primary and VPC secondary.