Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, Release 6.x

First Published: 2013-12-23

Last Modified: 2022-02-08

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

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

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 you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments that are 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.

Convention	Description
`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 includes 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 that 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.

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

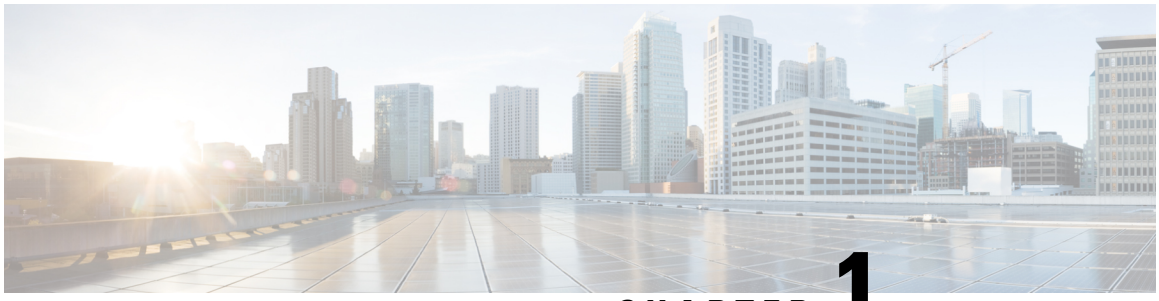
Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Preface



CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.

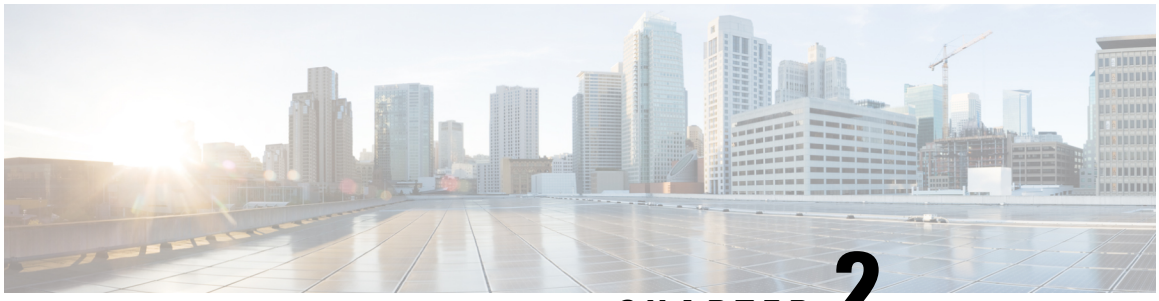
New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide* and where they are documented.

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
vPC forklift upgrade support	Support for upgrading from a pair of Nexus 9000 switches in a vPC topology to a different pair of Nexus 9000 series switches.	6.1(2)I3(4)	vPC Forklift Upgrade Scenario
IP-in-IP tunnel support	Enables encapsulation and decapsulation of packets to create a tunnel.	6.1(2)I3(4)	Point-to-Point IP-in-IP Tunnel Encapsulation and Decapsulation
Subinterface support on port-channel interfaces	Support for one or more subinterfaces on a port-channel interface.	6.1(2)I3(3)	Configuring Layer 3 Interfaces, on page 71
GRE support for IP tunnels	Support for the GRE carrier protocol to enable IP tunnels to enable IPV4 transport between two devices.	6.1(2)I3(2)	Configuring IP Tunnels, on page 229

Feature	Description	Changed in Release	Where Documented
Cisco Nexus 9300 subinterface support	Added support of Cisco Nexus 9300 subinterface support. (Removed restriction of no subinterface support for Cisco Nexus 9300 platforms.)	6.1(2)I3(1)	Configuring Layer 3 Interfaces, on page 71
FEX support	Added Cisco Nexus 2000 Fabric Extender(FEX) support.	6.1(2)I2(3)	Configuring vPCs, on page 161
Cisco QSFP+ to SFP+ Adapter (QSA) module	Added the Cisco QSFP+ to SFP+ Adapter (QSA) module feature to provide 40G to 10G conversion support.	6.1(2)I2(2)	Configuring Basic Interface Parameters, on page 9



CHAPTER 2

Overview

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide*.

About Interfaces

Cisco NX-OS supports multiple configuration parameters for each of the interface types supported. Most of these parameters are covered in this guide but some are described in other documents.

The following table shows where to get further information on the parameters you can configure for an interface.

Table 2: Interface Parameters

Feature	Parameters	Further Information
Basic parameters	Description, duplex, error disable, flow control, MTU, beacon	"Configuring Basic Interface Parameters"
Layer 3	Medium, IPv4 and IPv6 addresses	"Configuring Layer 3 Interfaces"
Layer 3	Bandwidth, delay, IP routing, VRFs	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* *Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide*
Port Channels	Channel group, LACP	"Configuring Port Channels"

Feature	Parameters	Further Information
Security	EOU	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*

Ethernet Interfaces

- Ethernet interfaces include routed ports.

Access Ports

An access port carries traffic for one VLAN. This type of port is a Layer 2 interface only.

For more information on access ports, see the “Information About Access and Trunk Interfaces” section.

Routed Ports

A routed port is a physical port that can route IP traffic to another device. A routed port is a Layer 3 interface only.

For more information on routed ports, see the “Routed Interfaces” section.

Management Interface

You can use the management Ethernet interface to connect the device to a network for remote management using a Telnet client, the Simple Network Management Protocol (SNMP), or other management agents. The management port (mgmt0) is autosensing and operates in full-duplex mode at a speed of 10/100/1000 Mb/s.

For more information on the management interface, see the Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide.

Port-Channel Interfaces

A port channel is a logical interface that is an aggregation of multiple physical interfaces. You can bundle up to 32 individual links (physical ports) into a port channel to improve bandwidth and redundancy. For more information about port-channel interfaces, see the “Configuring Port Channels” section.

Subinterfaces

You can create virtual subinterfaces using a parent interface configured as a Layer 3 interface. A parent interface can be either a physical port or a port-channel. A parent interface can be a physical port. Subinterfaces divide the parent interface into two or more virtual interfaces on which you can assign unique Layer 3 parameters such as IP addresses and dynamic routing protocols.

Loopback Interfaces

A loopback interface is a virtual interface with a single endpoint that is always up. Any packet that is transmitted over a virtual loopback interface is immediately received by that interface. Loopback interfaces emulate a physical interface. For more information about subinterfaces, see the “Loopback Interfaces” section.

Breakout Interfaces

Cisco NX-OS supports the breakout of a high bandwidth interface into one or more low bandwidth interfaces at the module level or at the per-port level.

Module Level Breakout

For module level breakout, the **interface breakout** command splits the high bandwidth 40G interface of a module into four 10G interfaces. The module is reloaded and the configuration for the interface is removed when the command is executed.

The following is an example of the command:

```
switch# configure terminal
switch(config)# interface breakout module 1
Module will be reloaded. Are you sure you want to continue(yes/no)? yes
```

The **no interface breakout module** *module_number* command undoes the breakout configuration. It puts all interfaces of the module in 40G mode and deletes the configuration for the previous 10G interfaces.

About the Lane Selector

The lane selector is a push button switch and 4 LEDs located on the Cisco Nexus switch (left side of front panel, labeled "LS"). The push button switch and LEDs are used to indicate the status of the ports. The lane selector is supported on Cisco Nexus Series 9000 series switches and the Cisco Nexus 3164 and 3232 switches.

By default, the LEDs indicate the link/activity status of a 1 x 40G configuration. When the ports are configured as 4 x 10G, you can access the link status of each individual 10G port with the lane selector.

By pressing the lane selector push button, the port LED shows the selected lane's link/activity status. The 1st time the push button is pressed, the first LED displays the status of the first port. Pressing the push button a 2nd time displays the status of the second port, and so on. You can display the status of each of the four ports by pressing the push button in this manner.

For example, if port 60 is configured as 4 x 10G, pressing the lane selector push button once displays the link status of 60/1/1. Pressing the push button a second time displays the link status of 60/1/2.

When you press the push button after displaying the status of the last port, all four of the LEDs should extinguish to indicate that the lane selector has returned to display the status for the default 1 x 40G configuration.

Note A 10G breakout port's LED blinks when the beacon feature has been configured for it.

Note When a port is configured to be in 10G breakout mode and no lane is selected, the 40G port's LED illuminates as green even though only one of the 10G breakout ports is up.

Notes About Breakout Interfaces

Cisco Nexus 9516 switch does not support breakout on Modules 8 to 16.

The following table provides detailed information of the supported or not supported breakout modes. For more information, see Cisco Nexus Data Sheets:

Caveats

- As of Cisco NX-OS Release 7.0(3)I7(2), manual breakout of QSA ports is not supported.

  Manual breakout is supported on the following platforms because auto-breakout does not happen successfully on them—N9K-C93128TX, N9K-9332, N9K-C9396PX, N9K-C9396TX, N9K-C9372PX, N9K-C9372TX, N9K-C9332PQ, N9K-C93120TX, N9K-9432PQ, N9K-9536PQ, N9K-9636PQ, N9K-X9632PC-QSFP100, N9K-X9432C-S, N3K-C3132Q-V, N3K-C3164Q, N3K-C3132C, N3K-C3232C, N3K-C3264Q, N3K-C3264C, N3K-3064Q, N3K-3016, N3K-3172.

  You need to perform manual breakout using "interface breakout module <*module number*> port <*port range*> map <*breakout mapping*>" command.

- When a break-out port is configured as a part of a port-channel, you need to apply the configuration twice (after write-erase/reload), to ensure the effectiveness of the port-channel.

- When you upgrade a Cisco Nexus 9000 device to Cisco NX-OS Release 7.0(3)I7(2) or later releases, if a QSFP port is configured with the manual breakout command and is using a QSA, the configuration of the interface Ethernet 1/50/1 is no longer supported and will need to be removed. To restore the configuration, you must manually configure the Ethernet 1/50 on the device.

  This behaviour is not applicable to the following platforms—N9K-C93128TX, N9K-9332, N9K-C9396PX, N9K-C9396TX, N9K-C9372PX, N9K-C9372TX, N9K-C9332PQ, N9K-C93120TX, N9K-9432PQ, N9K-9536PQ, N9K-9636PQ, N9K-X9632PC-QSFP100, N9K-X9432C-S, N3K-C3132Q-V, N3K-C3164Q, N3K-C3132C, N3K-C3232C, N3K-C3264Q, N3K-C3264C, N3K-3064Q, N3K-3016, N3K-3172—because manual breakout is supported on these platforms.

- Beginning with Cisco NX-OS Release 7.0(3)I7(3) you see two additional options to configure FEC such as **rs-cons16** and **rs-ieee** as per IEEE standards.

Note Auto-FEC is not supported in Cisco NX-OS Release 7.0(3)I7(x)

High Bandwidth Interfaces

The breakout of high bandwidth interfaces are supported only on:

- The X9636PQ, X9432PQ, X9536PQ, and X9732C-EX line cards on a Cisco Nexus 9500 Series switch.
- The Cisco Nexus 9332PQ switch.
- The Cisco Nexus 3164Q switch.

Cisco Nexus 9000 C93180LC-EX Switch

For 7.0(3)I7(1) and later, Cisco Nexus 9000 C93180LC-EX switch provides three different modes of operation:

- Mode 1: 28 x 40G + 4 x 40G/100G (Default configuration)
  - Hardware profile portmode 4x100g + 28x40g.
  - 10x4 breakout is supported on the top ports from 1 to 27 (ports 1,3,5, 7...27). If any of the top port is broken out, the corresponding bottom port becomes non-operational. For example, if port 1 is broken out port 2 becomes non-operational.

    - 1 Gigabit and 10 Gigabit QSA is supported on ports 29, 30, 31, and 32. However, QSAs on the top and bottom front panel ports must be of same speed.
    - Ports 29, 30, 31, and 32 support 10x4, 25x4, and 50x2 breakout.

- Mode 2: 24 x 40G + 6 x 40G/100G
    - Hardware profile portmode 6x100g + 24x40g.
    - 10x4 breakout is supported on the top ports from 1 to 23 (ports 1,3,5, 7...23). If any of the top port is broken out the corresponding bottom port becomes non-operational.
    - Ports 25, 27, 29, 30, 31, and 32 support 10x4, 25x4, and 50x2 breakout.
    - 1 Gigabit and 10 Gigabit QSA is supported on ports 29, 30, 31, and 32. However, QSAs on the top and bottom front panel ports must be of same speed.

- Mode 3: 18 x 40G/100G
    - Hardware profile portmode 18x100g.
    - 10x4, 25x4, and 50x2 breakout is supported on top ports from 1 to 27 (ports 1,3,5, 7...27) and on ports 29,30,31,32.
    - 1 Gigabit and 10 Gigabit QSA is supported on all the 18 ports.

Changing Mode 3 to any other mode or vice versa requires **copy running-config startup-config** command followed by **reload** command to take effect. However, moving between Modes 1 and 2 is dynamic and requires only **copy running-config startup-config** command.

Use the **show running-config | grep portmode** command to display the current operation mode.

Example:

```
switch(config-if-range)# show running-config | grep portmode

hardware profile portmode 4x100G+28x40G
```

With the Cisco Nexus C93180LC-EX switch, there are three breakout modes:

- 40G to 4x10G breakout ports
    - Enables the breakout of 40G ports into 4 X 10G ports.
    - Use the **interface breakout module 1 port** *x* **map 10g-4x** command.

- 100G to 4x25G breakout ports
    - Enables the breakout of 100G ports into 4 X 25G ports.
    - Use the **interface breakout module 1 port** *x* **map 25g-4x** command.

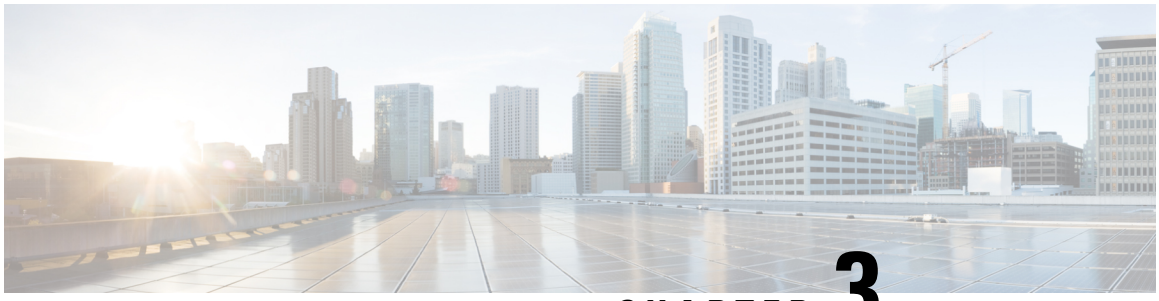
- 100G to 2x50G breakout ports
    - Enables the breakout of 100G ports into 2 X 50G ports.
    - Use the **interface breakout module 1 port** *x* **map 50g-2x** command.

Virtual Device Contexts

Cisco NX-OS can segment operating system and hardware resources into virtual device contexts (VDCs) that emulate virtual devices. The Cisco Nexus 9000 Series switch does not support multiple VDCs. All switch resources are managed in the default VDC.

High Availability for Interfaces

Interfaces support stateful and stateless restarts. A stateful restart occurs on a supervisor switchover. After the switchover, Cisco NX-OS applies the runtime configuration.



CHAPTER 3

Configuring Basic Interface Parameters

This chapter describes how to configure the basic interface parameters on Cisco NX-OS devices.

About the Basic Interface Parameters

Description

For the Ethernet and management interfaces, you can configure the description parameter to provide a recognizable name for the interface. Using a unique name for each interface allows you to quickly identify the interface when you are looking at a listing of multiple interfaces.

For information about setting the description parameter for port-channel interfaces, see the “Configuring a Port-Channel Description” section. For information about configuring this parameter for other interfaces, see the “Configuring the Description” section.

Beacon

The beacon mode allows you to identify a physical port by flashing its link state LED with a green light. By default, this mode is disabled. To identify the physical port for an interface, you can activate the beacon parameter for the interface.

For information about configuring the beacon parameter, see the “Configuring the Beacon Mode” section.

Error Disabled

A port is in the error-disabled (err-disabled) state when the port is enabled administratively (using the **no shutdown** command) but disabled at runtime by any process. For example, if UDLD detects a unidirectional link, the port is shut down at runtime. However, because the port is administratively enabled, the port status

displays as err-disable. Once a port goes into the err-disable state, you must manually reenable it or you can configure a timeout value that provides an automatic recovery. By default, the automatic recovery is not configured, and by default, the err-disable detection is enabled for all causes.

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 error-disabled recovery timeout for a particular error-disabled cause and configure the recovery period.

The **errdisable recovery cause** command provides an automatic recovery after 300 seconds.

You can use the **errdisable recovery interval** command to change the recovery period within a range of 30 to 65535 seconds. You can also configure the recovery timeout for a particular err-disable cause.

If you do not enable the error-disabled recovery for the cause, the interface stays in the error-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 error-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.

Interface Status Error Policy

Cisco NX-OS policy servers such as Access Control List (ACL) Manager and Quality of Service (QoS) Manager, maintain a policy database. A policy is defined through the command-line interface.

Policies are pushed when you configure a policy on an interface to ensure that policies that are pushed are consistent with the hardware policies. To clear the errors and to allow the policy programming to proceed with the running configuration, enter the **no shutdown** command. If the policy programming succeeds, the port is allowed to come up. If the policy programming fails, the configuration is inconsistent with the hardware policies and the port is placed in an error-disabled policy state. The error-disabled policy state remains and the information is stored to prevent the same port from being brought up in the future. This process helps to avoid unnecessary disruption to the system.

Modifying Interface MTU Size

The maximum transmission unit (MTU) size specifies the maximum frame size that an Ethernet port can process. For transmissions to occur between two ports, you must configure the same MTU size for both ports. A port drops any frames that exceed its MTU size.

By default, the Cloud-Scale ASIC NX-OS system always allows an extra 166B in the MTU on top of the configured value in order to fully support/accept different types of encapsulations in the hardware.

Cisco NX-OS allows you to configure MTU on an interface, with options to configure it on different level in the protocol stack. By default, each interface has an MTU of 1500 bytes, which is the IEEE 802.3 standard for Ethernet frames. Larger MTU sizes are possible for more efficient processing of data to allow different application requirements. The larger frames, are also called jumbo frames, can be up to 9216 bytes in size.

MTU is configured per interface, where an interface can be a Layer 2 or a Layer 3 interface. For a Layer 2 interface, you can configure the MTU size with one of two values, the value system default MTU value or the system jumbo MTU value. The system default MTU value is 1500 bytes. Every Layer 2 interface is configured with this value by default. You can configure an interface with the default system jumbo MTU value, that is 9216 bytes. To allow an MTU value from 1500 through 9216, you must adjust the system jumbo MTU to an appropriate value where interface can be configured with the same value.

Note You can change the system jumbo MTU size. When the value is changed, the Layer 2 interfaces that use the system jumbo MTU value, will automatically changes to the new system jumbo MTU value.

A Layer 3 interface, can be Layer 3 physical interface (configure with no switchport), switch virtual interface (SVI), and sub-interface, you can configure an MTU size between 576 and 9216 bytes.

For the Cisco Nexus 9372 switch, the following applies:

- The 10-G interfaces are mapped to specific hardware ports where the default MTU is 1500.
- The 40-G interfaces are mapped as a HiGiG port where the default MTU is 3FFF and the MTU limit check is disabled.
- In the case of 40-G interfaces, since the MTU limit check is disabled, it ignores the packet size and traffic flows irrespective of its MTU.
- When the configured MTU of all interfaces on the switch do not match, the switch's behavior may vary depending on the specific port that is mismatched as well as the traffic flow. The following are examples of the switch's behavior in various scenarios:
  - When a Layer 3 port receives a frame whose length exceeds the port's MTU size, the port will drop the frame.
  - When a Layer 3 port receives a frame whose length is less than the ingress port's MTU size, but greater than the egress Layer 3 port's MTU size, then the frame is punted to the supervisor of the switch.
    1. If the frame is an IP packet that has the Don't Fragment (DF) bit set, then the frame will be dropped in software. Otherwise, the frame will be fragmented in software.
    2. Otherwise, the frame will be fragmented in software.
    3. This can cause performance issues (such as increased latency or packet loss for affected traffic flows) due to Control Plane Policing (CoPP) enabled by default on Cisco Nexus switches. For more information about Control Plane Policing, refer to the **Configuring Control Plane Policing** chapter of the **Cisco Nexus 9000 Series NX-OS Security Configuration Guide**.
  - When a Layer 2 port receives a frame whose length exceeds the port's MTU size, the port will drop the frame.
  - When a Layer 2 port receives a frame whose length is less than the ingress port's MTU size, but greater than the egress Layer 2 port's MTU size, and the frame is routed between VLANs by the switch, then the frame is punted to the supervisor of the switch.
    1. If the frame is an IP packet that has the Don't Fragment (DF) bit set, then the frame will be dropped in software. Otherwise, the frame will be fragmented in software.
    2. Otherwise, the frame will be fragmented in software.
    3. This can cause performance issues (such as increased latency or packet loss for affected traffic flows) due to Control Plane Policing (CoPP) enabled by default on Cisco Nexus switches. For more information about Control Plane Policing, refer to the **Configuring Control Plane Policing** chapter of the **Cisco Nexus 9000 Series NX-OS Security Configuration Guide**.

- When a Layer 2 port receives a frame whose length is less than the ingress port's MTU size, but greater than the egress Layer 2 port's MTU size, and the frame is switched within the same VLAN by the switch, then the switch will drop the frame.

For information about setting the MTU size, see the *Configuring the MTU Size* section.

Note On Cisco Nexus 9300-FX2 and 9300-GX devices, if ingress interface is configured with an MTU less than 9216, FTE does not capture input errors and does not display any events. However, if the ingress interface is configured with an MTU of 9216, FTE displays all the events.

Bandwidth

Ethernet ports have a fixed bandwidth of 1,000,000 Kb at the physical layer. Layer 3 protocols use a bandwidth value that you can set for calculating their internal metrics. The value that you set is used for informational purposes only by the Layer 3 protocols—it does not change the fixed bandwidth at the physical layer. For example, the Enhanced Interior Gateway Routing Protocol (EIGRP) uses the minimum path bandwidth to determine a routing metric, but the bandwidth at the physical layer remains at 1,000,000 Kb.

For information about configuring the bandwidth parameter for port-channel interfaces, see the “Configuring the Bandwidth and Delay for Informational Purposes” section. For information about configuring the bandwidth parameter for other interfaces, see the “Configuring the Bandwidth” section.

Throughput Delay

Specifying a value for the throughput-delay parameter provides a value used by Layer 3 protocols; it does not change the actual throughput delay of an interface. The Layer 3 protocols can use this value to make operating decisions. For example, the Enhanced Interior Gateway Routing Protocol (EIGRP) can use the delay setting to set a preference for one Ethernet link over another, if other parameters such as link speed are equal. The delay value that you set is in the tens of microseconds.

For information about configuring the bandwidth parameter for port-channel interfaces, see the “Configuring the Bandwidth and Delay for Informational Purposes” section. For information about configuring the throughput-delay parameter for other interfaces, see the “Configuring the Throughput Delay” section.

Administrative Status

The administrative-status parameter determines whether an interface is up or down. When an interface is administratively down, it is disabled and unable to transmit data. When an interface is administratively up, it is enabled and able to transmit data.

For information about configuring the administrative status parameter for port-channel interfaces, see the “Shutting Down and Restarting the Port-Channel Interface” section. For information about configuring the administrative-status parameter for other interfaces, see the “Shutting Down and Activating the Interface” section.

Unidirectional Link Detection Parameter

UDLD Overview

The Cisco-proprietary Unidirectional Link Detection (UDLD) protocol allows devices that are connected through fiber-optic 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 a device detects a unidirectional link, UDLD shuts down the affected LAN port and alerts the user. Unidirectional links can cause a variety of problems.

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 detections work to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

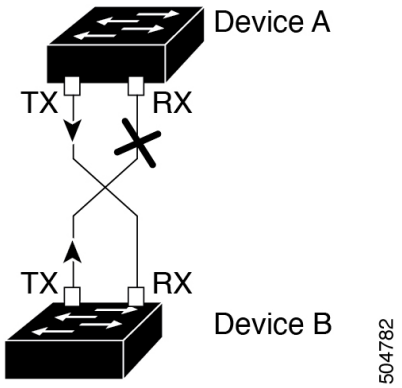
A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does 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, UDLD 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.

The Cisco Nexus 9000 Series 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. You can configure the transmission interval for the UDLD frames, either globally or for the specified interfaces.

Note By default, UDLD is locally disabled on copper LAN ports to avoid sending unnecessary control traffic on this type of media.

The 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 3: UDLD Default Configuration

Feature	Default Value
UDLD global enable state	Globally 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	Disabled on all Ethernet 10/100 and 1000BASE-TX LAN ports
UDLD aggressive mode	Disabled
UDLD message interval	15 seconds

For information about configuring the UDLD for the device and its port, see the "Configuring the UDLD Mode" section.

UDLD Normal and Aggressive Modes

UDLD supports Normal and Aggressive modes of operation. By default, Normal mode is enabled.

In Normal mode, UDLD detects the following link errors by examining the incoming UDLD packets from the peer port:

- Empty echo packet
- Uni-direction
- TX/RX loop
- Neighbor mismatch

By default, UDLD aggressive mode is disabled. 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 frame, UDLD tries to re-establish the connection with the neighbor. After eight failed retries, the port is disabled.

In the following scenarios, enabling the UDLD aggressive mode disables one of the ports to prevent the discarding of traffic.

- 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

Note You enable the UDLD aggressive mode globally to enable that mode on all the fiber ports. You must enable the UDLD aggressive mode on copper ports on specified interfaces.

Tip When a line card upgrade is being performed during an in-service software upgrade (ISSU) and some of the ports on the line card are members of a Layer 2 port channel and are configured with UDLD aggressive mode, if you shut down one of the remote ports, UDLD puts the corresponding port on the local device into an error-disabled state. This behavior is correct.

To restore service after the ISSU has completed, enter the **shutdown** command followed by the **no shutdown** command on the local port.

Port-Channel Parameters

A port channel is an aggregation of physical interfaces that comprise a logical interface. You can bundle up to 32 individual interfaces into a port channel to provide increased bandwidth and redundancy. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational if at least one physical interface within the port channel is operational.

You can create Layer 3 port channels by bundling compatible Layer 3 interfaces.

Any configuration changes that you apply to the port channel are applied to each interface member of that port channel.

For information about port channels and for information about configuring port channels, see Chapter 6, “Configuring Port Channels.”

Cisco QSFP+ to SFP+ Adapter Module Support

The Cisco QSFP+ to SFP+ Adapter (QSA) module provides 10G support for the 40G uplink ports that are a part of the Cisco Nexus M6PQ and Cisco Nexus M12PQ uplink modules of specific Cisco Nexus 9300 devices.

A group of six consecutive ports in the M6PQ or M12PQ uplink module must be operating at the same speed (40G or 10G) to use the QSA/QSFP modules.

- For Cisco Nexus 9396PX devices, 2/1-6 ports form the first port speed group and the remaining 2/7-12 ports form the second port speed group.
- For Cisco Nexus 93128PX/TX devices, 2/1-6 ports form the first port speed group and the remaining 2/7-8 ports form the second port speed group.
- For Cisco Nexus 937xPX/TX devices, 1/49-54 ports form the only port speed group.
- For Cisco Nexus 93120TX devices, 1/97-102 ports form the only port speed group.
- For Cisco Nexus 9332PQ devices, 1/27-32 ports form the only port speed group.

Use the **speed-group 10000** command to configure the first port of a port speed group for the QSA. This command specifies the administrator speed preference for the port group. (The default port speed is 40G.)

- The **speed-group 10000** command specifies a speed of 10G.
- The **no speed-group 10000** command specifies a speed of 40G.

After the speed has been configured, the compatible transceiver modules are enabled. The remaining transceiver modules in the port group (incompatible transceiver modules) become error disabled with a reason of "check speed-group config".

Note The Cisco QSFP+ to SFP+ Adapter (QSA) module does not provide 10G support for the 40G line cards for Cisco Nexus 9500 devices.

25G Autonegotiation Overview

Guidelines and Limitations

Basic interface parameters have the following configuration guidelines and limitations:

- MDIX is enabled by default on copper ports. It is not possible to disable it.
- **show** commands with the **internal** keyword are not supported.
- Fiber-optic Ethernet ports must use Cisco-supported transceivers. To verify that the ports are using Cisco-supported transceivers, use the **show interface transceivers** command. Interfaces with Cisco-supported transceivers are listed as functional interfaces.
- A port can be either a Layer 2 or a Layer 3 interface; it cannot be both simultaneously.

  By default, each port is 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.
- You usually configure Ethernet port speed and duplex mode parameters to auto to allow the system to negotiate the speed and duplex mode between ports. If you decide to configure the port speed and duplex modes manually for these ports, consider the following:
  - Before you configure the speed and duplex mode for an Ethernet or management interface, see the Default Settings section for the combinations of speeds and duplex modes that can be configured at the same time.
  - If you set the Ethernet port speed to auto, the device automatically sets the duplex mode to auto.
  - If you enter the **no speed** command, the device automatically sets both the speed and duplex parameters to auto (the **no speed** command produces the same results as the **speed auto** command).
  - If you configure an Ethernet port speed to a value other than auto (for example, 1G, 10G, or 40G), you must configure the connecting port to match. Do not configure the connecting port to negotiate the speed.
  - To configure speed, duplex, and automatic flow control for an Ethernet interface, you can use the **negotiate auto** command. To disable automatic negotiation, use the **no negotiate auto** command.

Note The device cannot automatically negotiate the Ethernet port speed and duplex mode if the connecting port is configured to a value other than auto.

Caution Changing the Ethernet port speed and duplex mode configuration might shut down and reenable the interface.

- For BASE-T copper ports, auto-negotiation is enabled even when fixed speed is configured.
- The port profile feature is not supported.
- Auto-negotiation is not supported on 25-Gigabit Ethernet Transceiver Modules on Cisco Nexus 9200 and 9300-EX platform switches; and Cisco Nexus 9500 platform switches that uses N9K-X9700-EX line cards.
- When using a QSFP-40G-CR4 on Cisco Nexus 9000 switches, you must configure the default speed as 40G in the auto-negotiation parameters. Otherwise, the interface may not be able to bring the link up.
- The following line cards do not support Link Training:

  Nexus 9300 Modules:

  - N9K-M12PQ (C9396PX, C9396TX, C93128PX, C93128TX)

  Nexus 9500 Modules:

  - X9536PQ
  - X9564PX
  - X9564TX

- If cable length is more than 5 meters, Auto Negotiation is not supported. This cable length limitation is applicable only to copper cables and not applicable to optical cables.

Default Settings

The following lists the default settings for the basic interface parameters.

Parameter	Default
Description	Blank
Beacon	Disabled
Bandwidth	Data rate of interface
Throughput delay	100 microseconds
Administrative status	Shutdown
MTU	1500 bytes
UDLD global	Globally disabled
UDLD per-port enable state for fiber-optic media	Enabled on all Ethernet fiber-optic LAN ports

Parameter	Default
UDLD per-port enable state for copper media	Disabled on all Ethernet 1G, 10G, or 40G LAN ports
UDLD message interval	Disabled
UDLD aggressive mode	Disabled
Error disable	Disabled
Error disable recovery	Disabled
Error disable recovery interval	300 seconds
Buffer-boost	Enabled **Note** Feature available on N9K-X9564TX and N9K-X9564PX line cards and Cisco Nexus 9300 series devices.

Configuring the Basic Interface Parameters

When you configure an interface, you must specify the interface before you can configure its parameters.

Specifying the Interfaces to Configure

Before you begin

Before you can configure the parameters for one or more interfaces of the same type, you must specify the type and the identities of the interfaces.

The following table shows the interface types and identities that you should use for specifying the Ethernet and management interfaces.

Table 4: Information Needed to Identify an Interface for Configurations

Interface Type	Identity
Ethernet	I/O module slot numbers and port numbers on the module
Management	0 (for port 0)

The interface range configuration mode allows you to configure multiple interfaces with the same configuration parameters. After you enter the interface range configuration mode, all command parameters you enter are attributed to all interfaces within that range until you exit out of the interface range configuration mode.

You enter a range of interfaces using dashes (-) and commas (,). Dashes separate contiguous interfaces and commas separate noncontiguous interfaces. When you enter noncontiguous interfaces, you must enter the media type for each interface.

This example shows how to configure a contiguous interface range:

```
switch(config)# interface ethernet 2/29-30
switch(config-if-range)#
```

This example shows how to configure a noncontiguous interface range:

```
switch(config)# interface ethernet 2/29, ethernet 2/33, ethernet 2/35
switch(config-if-range)#
```

You can specify subinterfaces in a range only when the subinterfaces are on the same port, for example, 2/29.1-2. But you cannot specify the subinterfaces in a range of ports, for example, you cannot enter 2/29.2-2/30.2. You can specify two of the subinterfaces discretely, for example, you can enter 2/29.2, 2/30.2.

This example shows how to configure a a breakout cable:

```
switch(config)# interface ethernet 1/2/1
switch(config-if-range)#
```

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface*

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *interface*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
	Example:
	`switch(config)# interface mgmt0`
	`switch(config-if)#`
		Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use **ethernet** *slot/port*. For the management interface, use **mgmt0**.
		Examples:
		• The 1st example shows how to specify the slot 2, port 1 Ethernet interface.
		• The 2nd example shows how to specify the management interface.
		Note You do not need to add a space between the interface type and identity (port or slot/port number) For example, for the Ethernet slot 4, port 5 interface, you can specify either “ethernet 4/5” or “ethernet4/5.” The management interface is either “mgmt0” or “mgmt 0.”
		When you are in the interface configuration mode, the commands that you enter configure the interface that you specified for this mode.

Configuring the Description

You can provide textual interface descriptions for the Ethernet and management interfaces.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface*
3. **description** *text*
4. **show interface** *interface*
5. **exit**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *interface*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
	Example:
	`switch(config)# interface mgmt0`
	`switch(config-if)#`
		Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use **ethernet** *slot/port*. For the management interface, use **mgmt0**.
		Examples:
		• The 1st example shows how to specify the slot 2, port 1 Ethernet interface.
		• The 2nd example shows how to specify the management interface.

Step 3	**description** *text*
	Example:
	`switch(config-if)# description Ethernet port 3 on module 1`
	`switch(config-if)#`
		Specifies the description for the interface.

Step 4	**show interface** *interface* **Example:** `switch(config)# show interface ethernet 2/1`	(Optional) Displays the interface status, which includes the description parameter.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

	Command or Action	Purpose
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the interface description to Ethernet port 24 on module 3:

```
switch# configure terminal
switch(config)# interface ethernet 3/24
switch(config-if)# description server1
switch(config-if)#
```

The output of the **show interface eth** command is enhanced as shown in the following example:

```
Switch# show version
Software
BIOS: version 06.26
NXOS: version 6.1(2)I2(1) [build 6.1(2)I2.1]
BIOS compile time: 01/15/2014
NXOS image file is: bootflash:///n9000-dk9.6.1.2.I2.1.bin
NXOS compile time: 2/25/2014 2:00:00 [02/25/2014 10:39:03]

switch# show interface ethernet 6/36
Ethernet6/36 is up
admin state is up, Dedicated Interface
Hardware: 40000 Ethernet, address: 0022.bdf6.bf91 (bia 0022.bdf8.2bf3)
Internet Address is 192.168.100.1/24
MTU 9216 bytes, BW 40000000 Kbit, DLY 10 usec
```

Configuring the Beacon Mode

You can enable the beacon mode for an Ethernet port to flash its LED to confirm its physical location.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. [**no**] **beacon**
4. **show interface ethernet** *slot/port*
5. **exit**
6. **copy running-config startup-config**

DETAILED STEPS

	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 3/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	[**no**] **beacon**
	Example:
	`switch(config)# beacon`
	`switch(config-if)#`
		Enables the beacon mode or disables the beacon mode. The default mode is disabled.

Step 4	**show interface ethernet** *slot/port*
	Example:
	`switch(config)# show interface ethernet 2/1`
	`switch(config-if)#`
		(Optional) Displays the interface status, which includes the beacon mode state.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to enable the beacon mode for the Ethernet port 3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# beacon
switch(config-if)#
```

This example shows how to disable the beacon mode for the Ethernet port 3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# no beacon
switch(config-if)#
```

This example shows how to configure the dedicated mode for Ethernet port 4/17 in the group that includes ports 4/17, 4/19, 4/21, and 4/23:

```
switch# configure terminal
switch(config)# interface ethernet 4/17, ethernet 4/19, ethernet 4/21, ethernet 4/23
switch(config-if)# shutdown
switch(config-if)# interface ethernet 4/17
switch(config-if)# no shutdown
switch(config-if)#
```

Configuring the Error-Disabled State

You can view the reason that an interface moves to the error-disabled state and configure automatic recovery.

Enabling the Error-Disable Detection

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

SUMMARY STEPS

1. **configure terminal**
2. **errdisable detect cause** {**acl-exception** | **all** | **link-flap** | **loopback**}
3. **shutdown**
4. **no shutdown**
5. **show interface status err-disabled**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**errdisable detect cause** {**acl-exception** | **all** | **link-flap** | **loopback**}
	Example:
	`switch(config)# errdisable detect cause all`
	`switch(config-if)#`
		Specifies a condition under which to place the interface in an error-disabled state. The default is enabled.

Step 3	**shutdown**
	Example:
	`switch(config-if)# shutdown`
	`switch(config)#`
		Brings the interface down administratively. To manually recover the interface from the error-disabled state, enter this command first.

Step 4	**no shutdown**
	Example:
	`switch(config-if)# no shutdown`
	`switch(config)#`
		Brings the interface up administratively and enables the interface to recover manually from the error-disabled state.

	Command or Action	Purpose
Step 5	**show interface status err-disabled** **Example:** `switch(config)# show interface status err-disabled`	(Optional) Displays information about error-disabled interfaces.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

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

```
switch(config)# errdisable detect cause all
switch(config)#
```

Enabling the Error-Disabled Recovery

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

SUMMARY STEPS

1. **configure terminal**
2. **errdisable recovery cause** {**all** | **bpduguard** | **failed-port-state** | **link-flap** | **loopback** | **miscabling** | **psecure-violation** | **security-violation** | **storm-control** | **udld** | **vpc-peerlink**}
3. **show interface status err-disabled**
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**errdisable recovery cause** {**all**	**bpduguard**	**failed-port-state**	**link-flap**	**loopback**	**miscabling**	**psecure-violation**	**security-violation**	**storm-control**	**udld**	**vpc-peerlink**} **Example:** `switch(config)# errdisable recovery cause all` `switch(config-if)#`	Specifies a condition under which the interface automatically recovers from the error-disabled state, and the device retries bringing the interface up. The device waits 300 seconds to retry. The default is disabled.

	Command or Action	Purpose

Step 3	**show interface status err-disabled**
	Example:
	`switch(config)# show interface status err-disabled`
	`switch(config-if)#`
		(Optional) Displays information about error-disabled interfaces.

Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

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

```
switch(config)# errdisable recovery cause all
switch(config)#
```

Configuring the Error-Disabled Recovery Interval

You can configure the error-disabled recovery timer value.

SUMMARY STEPS

1. **configure terminal**
2. **errdisable recovery interval** *interval*
3. **show interface status err-disabled**
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**errdisable recovery interval** *interval*
	Example:
	`switch(config)# errdisable recovery interval 32`
	`switch(config-if)#`
		Specifies the interval for the interface to recover from the error-disabled state. The range is from 30 to 65535 seconds, and the default is 300 seconds.

Step 3	**show interface status err-disabled**
	Example:
	`switch(config)# show interface status err-disabled`
	`switch(config-if)#`
		(Optional) Displays information about error-disabled interfaces.

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

Example

This example shows how to configure the error-disabled recovery timer to set the interval for recovery to 32 seconds:

```
switch(config)# errdisable recovery interval 32
switch(config)#
```

Configuring the MTU Size

MTU is configured per interface, where the interface can be a Layer 2 or a Layer 3 interface. Every interface has default MTU of 1500 bytes. This value is called system default MTU. You can configure a Layer 2 interface, with a a value of 9216 bytes, which is the default value of the system jumbo MTU. To allow an MTU value that is between 1500 and 9216, system jumbo MTU needs to be adjusted to appropriate value where interface can be configured with the same value.

Note You can change the system jumbo MTU size. When the value is changed, the Layer 2 interfaces that use the system jumbo MTU value, will automatically changes to the new system jumbo MTU value.

A Layer 3 interface, can be Layer 3 physical interface switch virtual interface (SVI), and subinterface, you can configure an MTU size between 576–9216 bytes.

Configuring the Interface MTU Size

For Layer 3 interfaces, you can configure an MTU with keyword MTU and value in bytes where value is between 576–9216 bytes.

For Layer 2 interfaces, you can configure an interface using the keyword MTU with value in bytes. The value can be a system default MTU size of 1500 bytes or the system jumbo MTU value that can be adjusted to the default size of 9216 bytes.

If you need to use a different system jumbo MTU size for Layer 2 interfaces, see the *Configuring the System Jumbo MTU Size* section.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*, vlan *vlan-id* mgmt *0*
3. **mtu** *size*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot/port*, vlan *vlan-id* mgmt *0*
	Example:
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)#`
	`switch(config)# interface vlan 100`
	`switch(config-if)#`
	`switch(config)# interface mgmt 0`
	`switch(config-if)#`
		Specifies an Ethernet interface to configure, and enters interface configuration mode.

Step 3	**mtu** *size*
	Example:
	`switch(config-if)# mtu 9216`
	`switch(config-if)#`
		Configure the MTU value on an interface.
		For a Layer 3 interface, a physical Layer 3 interface, an SVI or sub-interface, then the value can be between 576-9216 bytes. If the interface is a physical Layer 2 interface, then the value can be 1500 or system jumbo MTU value.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Example

This example shows how to configure the Layer 2 Ethernet port 3/1 with the default MTU size (1500):

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport
switch(config-if)# mtu 1500
switch(config-if)#
```

This example displays the output of show running-config interface command:

```
switch# show run int mgmt0
!Command: show running-config interface mgmt0
!Running configuration last done at: Fri May 31 11:32:28 2019
!Time: Fri May 31 11:32:33 2019
version 9.3(1) Bios:version 07.65
interface mgmt0
mtu 9216
vrf member management
ip address 168.51.170.73/82
```

Configuring the System Jumbo MTU Size

You can configure the system jumbo MTU size, which can be used to specify the MTU size for Layer 2 interfaces. You can specify an even number between 1500 and 9216. If you do not configure the system jumbo MTU size, it defaults to 9216 bytes.

Note To configure jumbo frames for FEX modules, configure the FEX fabric port-channel interface with the required MTU size for the FEX module.

SUMMARY STEPS

1. **configure terminal**
2. **system jumbomtu** *size*
3. **show running-config all**
4. **interface** *type slot/port*
5. **interface** *type*
6. **mtu** *size*
7. **exit**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**system jumbomtu** *size*
	Example:
	`switch(config)# system jumbomtu 8000`
	`switch(config)#`
		Specifies the system jumbo MTU size. Use an even number between 1500 and 9216.
		Note In general accepted practice, a jumbo frame is considered to have an MTU size greater than 9000 bytes.

Step 3	**show running-config all** **Example:** `switch(config)# show running-config all | include jumbomtu`	(Optional) Displays the current operating configuration, which includes the system jumbo MTU size.

Step 4	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Specifies an interface to configure and enters interface configuration mode.

Step 5	**interface** *type* **Example:**	Specifies the management interface to configure.

	Command or Action	Purpose
	`switch(config if)# interface mgmt0` `switch(config-if)#`	

Step 6	**mtu** *size*
	Example:
	`switch(config-if)# mtu 1500`
	`switch(config-if)#`
		For a Layer 2 interface, specifies either the default MTU size (1500) or the system jumbo MTU size that you specified earlier.
		For a Layer 3 interface, specifies any even size between 576 and 9216.

Step 7	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure the system jumbo MTU as 8000 bytes and how to change the MTU specification for an interface that was configured with the previous jumbo MTU size:

```
switch# configure terminal
switch(config)# system jumbomtu 8000
switch(config)# show running-config
switch(config)# interface ethernet 2/2
switch(config-if)# switchport
switch(config-if)# mtu 1500
switch(config-if)#
```

Configuring the Bandwidth

You can configure the bandwidth for Ethernet interfaces. The physical layer uses an unchangeable bandwidth of 1G, 10G, or 40G, but you can configure a value of 1 to 100,000,000 KB for Level 3 protocols.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **bandwidth** *kbps*
4. **show interface ethernet** *slot/port*
5. **exit**
6. **copy running-config startup-config**

DETAILED STEPS

	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 3/1`
	`switch(config-if)#`
		Specifies an Ethernet interface to configure, and enters interface configuration mode.

Step 3	**bandwidth** *kbps*
	Example:
	`switch(config-if)# bandwidth 1000000`
	`switch(config-if)#`
		Specifies the bandwidth as an informational-only value between 1 and 100,000,000.

Step 4	**show interface ethernet** *slot/port* **Example:** `switch(config)# show interface ethernet 2/1`	(Optional) Displays the interface status, which includes the bandwidth value.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure an informational value of 1,000,000 Kb for the Ethernet slot 3, port 1 interface bandwidth parameter:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# bandwidth 1000000
switch(config-if)#
```

Configuring the Throughput Delay

You can configure the interface throughput delay for Ethernet interfaces. The actual delay time does not change, but you can set an informational value between 1 and 16777215, where the value represents the number of tens of microseconds.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **delay** *value*
4. **show interface ethernet** *slot/port*
5. **exit**
6. **copy running-config startup-config**

DETAILED STEPS

	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 3/1`
	`switch(config-if)#`
		Specifies an Ethernet interface to configure, and enters interface configuration mode.

Step 3	**delay** *value*
	Example:
	`switch(config-if)# delay 10000`
	`switch(config-if)#`
		Specifies the delay time in tens of microseconds. You can set an informational value range between 1 and 16777215 tens of microseconds.

Step 4	**show interface ethernet** *slot/port*
	Example:
	`switch(config)# show interface ethernet 3/1`
	`switch(config-if)#`
		(Optional) Displays the interface status, which includes the throughput-delay time.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure the throughput-delay time so that one interface is preferred over another. A lower delay value is preferred over a higher value. In this example, Ethernet 7/48 is preferred over 7/47. The default delay for 7/48 is less than the configured value on 7/47, which is set for the highest value (16777215):

```
switch# configure terminal
switch(config)# interface ethernet 7/47
switch(config-if)# delay 16777215
switch(config-if)# ip address 192.168.10.1/24
switch(config-if)# ip router eigrp 10
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface ethernet 7/48
switch(config-if)# ip address 192.168.11.1/24
switch(config-if)# ip router eigrp 10
switch(config-if)# no shutdown
switch(config-if)#
```

Note You must first ensure the EIGRP feature is enabled by running the **feature eigrp** command.

Shutting Down and Activating the Interface

You can shut down and restart Ethernet or management interfaces. When you shut down interfaces, they become disabled and all monitoring displays show them as being down. This information is communicated to other network servers through all dynamic routing protocols. When the interfaces are shut down, the interface is not included in any routing updates. To activate the interface, you must restart the device.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface*
3. **shutdown**
4. **show interface** *interface*
5. **no shutdown**
6. **show interface** *interface*
7. **exit**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *interface*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
	
	`switch(config)# interface mgmt0`
	`switch(config-if)#`
		Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use *ethernet slot/port*. For the management interface, use *mgmt0*.
		Examples:

	Command or Action	Purpose
		• The 1st example shows how to specify the slot 2, port 1 Ethernet interface. • The 2nd example shows how to specify the management interface.

Step 3	**shutdown**
	Example:
	`switch(config-if)# shutdown`
	`switch(config-if)#`
		Disables the interface.

Step 4	**show interface** *interface*
	Example:
	`switch(config-if)# show interface ethernet 2/1`
	`switch(config-if)#`
		(Optional) Displays the interface status, which includes the administrative status.

Step 5	**no shutdown**
	Example:
	`switch(config-if)# no shutdown`
	`switch(config-if)#`
		Reenables the interface.

Step 6	**show interface** *interface*
	Example:
	`switch(config-if)# show interface ethernet 2/1`
	`switch(config-if)#`
		(Optional) Displays the interface status, which includes the administrative status.

Step 7	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to change the administrative status for Ethernet port 3/1 from disabled to enabled:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# shutdown
switch(config-if)# no shutdown
switch(config-if)#
```

Configuring the UDLD Mode

You can configure normal unidirectional link detection (UDLD) modes for Ethernet interfaces on devices configured to run UDLD.

Before you can enable the aggressive UDLD mode for an interface, you must make sure that UDLD is already enabled globally on the device and on the specified interfaces.

Note If the interface is a copper port, you must use the command enable UDLD to enable the UDLD. If the interface is a fiber port you need not explicitly enable UDLD on the interface. However if you attempt to enable UDLD on a fiber port using the enable UDLD command, you may get an error message indicating that is not a valid command.

The following table lists CLI details to enable and disable UDLD on different interfaces

Table 5: CLI Details to Enable or Disable UDLD on Different Interfaces

Description	Fiber port	Copper or Nonfiber port
Default setting	Enabled	Disabled
Enable UDLD command	**no udld disable**	**udld enable**
Disable UDLD command	**udld disable**	**no udld enable**

Before you begin

You must enable UDLD for the other linked port and its device.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature udld**
3. **udld message-time** *seconds*
4. **udld aggressive**
5. **interface ethernet** *slot/port*
6. **udld** [**enable** | **disable**]
7. **show udld** [**ethernet** *slot/port* | **global** | **neighbors**]
8. **exit**
9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	[**no**] **feature udld**
	Example:
	`switch(config)# feature udld`
	`switch(config)#`
	`switch(config)# no feature udld`
	`switch(config)#`
		Enables/Disables UDLD for the device.

Step 3	**udld message-time** *seconds*
	Example:
	`switch(config)# udld message-time 30`
	`switch(config)#`
		(Optional) Specifies the interval between sending UDLD messages. The range is from 7 to 90 seconds, and the default is 15 seconds.

Step 4	**udld aggressive**
	Example:
	`switch(config)# udld aggressive`
	`switch(config)#`
		Enables UDLD in aggressive mode by default on all fiber interfaces. Use the **no** form to disable aggressive mode UDLD on all fibers ports by default.
		Note Use the udld aggressive command to configure the ports to use a UDLD mode:
		• To enable fiber interfaces for the aggressive mode, enter the **udld aggressive** command in the global command mode and all the fiber interfaces will be in aggressive UDLD mode.
		• To enable the copper interfaces for the aggressive mode, you must enter the **udld aggressive** command in the interface mode, specifying each interface you want in aggressive UDLD mode.
		To use the aggressive UDLD mode, you must configure the interfaces on both ends of the link for the aggressive UDLD mode.

Step 5	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)#`
		(Optional) Specifies an interface to configure, and enters interface configuration mode.

Step 6	**udld** [**enable** | **disable**]
	Example:
	`switch(config-if)# udld enable`
	`switch(config-if)#`
		Enables UDLD in normal mode by default on all fiber interfaces. Use the **no** form to disable normal mode UDLD on all fibers ports by default.

Step 7	**show udld** [**ethernet** *slot/port* | **global** | **neighbors**] **Example:**	(Optional) Displays the UDLD status.

	Command or Action	Purpose
	`switch(config)# show udld` `switch(config)#`	

Step 8	**exit**
	Example:
	`switch(config-if-range)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 9	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to enable the UDLD for the device:

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

This example shows how to set the UDLD message interval to 30 seconds:

```
switch# configure terminal
switch(config)# feature udld
switch(config)# udld message-time 30
switch(config)#
```

This example shows how to disable UDLD for Ethernet port 3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if-range)# no udld enable
switch(config-if-range)# exit
```

This example shows how to disable UDLD for the device:

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

This example shows how to enable fiber interfaces for the aggressive UDLD mode:

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

This example shows how to enable the aggressive UDLD mode for the copper Ethernet interface3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3
switch(config-if)# udld aggressive
```

This example shows how to check if aggressive mode is enabled.

```
switch# sh udld global

UDLD global configuration mode: enabled-aggressive
UDLD global message interval: 15
switch#
```

This example shows how to check if udld aggressive mode is operational for a given interface.

```
switch# sh udld ethernet 8/2

Interface Ethernet8/2
------------------------------
Port enable administrative configuration setting: device-default
Port enable operational state: enabled-aggressive
Current bidirectional state: bidirectional
Current operational state:  advertisement - Single neighbor detected
Message interval: 15
Timeout interval: 5
<>
```

Configuring Debounce Timers

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.

Note The link state of 10G and 100G ports may change repeatedly when connected to service provider network. As a part of *link reset* or *break-link* functionality, it is expected that the Tx power light on the SFP to change to N/A state, at an event of link state change.

However, to prevent this behavior during the link state change, you may increase the link debounce timer to start from 500ms and increase it in 500ms intervals until the link stabilizes. On the DWDM, UVN, and WAN network, it is recommended to disable automatic link suspension (ALS) whenever possible. ALS suspends the link on the WAN when the Nexus turn off the link.

Note The **link debounce time** and **link debounce link-up time** commands can only be applied to a physical Ethernet interface.

Use the **show interface debounce** command to display the debounce times for all Ethernet ports.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **link debounce time** *time*

DETAILED STEPS

	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*
	Example:
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)#`
		Specifies an Ethernet interface to configure, and enters interface configuration mode.

Step 3	**link debounce time** *time*
	Example:
	`switch(config-if)# link debounce time 1000`
	`switch(config-if)#`
		Enables the debounce timer for the specified time (1 to 5000 milliseconds).
		If you specify 0 milliseconds, the debounce timer is disabled.

Example

- The following example enables the debounce timer and sets the debounce time to 1000 milliseconds for an Ethernet interface:

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

- The following example disables the debounce timer for an Ethernet interface:

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

Configuring link mac-up timer

This procedure describes how to configure mac up timers on DWDM/Dark fiber circuits.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **link mac-up timer** *seconds*

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port* **Example:**	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
	`switch(config)# interface ethernet1/2` `switch(config-if)#`	
Step 3	**link mac-up timer** *seconds* **Example:** `switch(config-if)# link mac-up timer 10`	Enables modification of the link mac-up timer. The link mac-up timer range is 0-120. **Note** This should only be done on DWDM links.

Verifying the Basic Interface Parameters

You can verify the basic interface parameters by displaying their values. You can also clear the counters listed when you display the parameter values.

To display basic interface configuration information, perform one of the following tasks:

Command	Purpose
show cdp all	Displays the CDP status.
show interface *interface*	Displays the configured states of one or all interfaces.
show interface brief	Displays a table of interface states.
show interface status err-disabled	Displays information about error-disabled interfaces.
show udld *interface*	Displays the UDLD status for the current interface or all interfaces.
show udld global	Displays the UDLD status for the current device.

Monitoring the Interface Counters

You can display and clear interface counters using Cisco NX-OS.

Displaying Interface Statistics

You can set up to three sampling intervals for statistics collections on interfaces.

SUMMARY STEPS

1. **configure terminal**
2. **interface ether** *slot/port*
3. **load-interval counters** [**1** | **2** | **3**] *seconds*
4. **show interface** *interface*
5. **exit**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface ether** *slot/port*
	Example:
	`switch(config)# interface ether 4/1`
	`switch(config)#`
		Specifies interface.

Step 3	**load-interval counters** [**1** | **2** | **3**] *seconds*
	Example:
	`switch(config)# load-interval counters 1 100`
	`switch(config)#`
		Sets up to three sampling intervals to collect bit-rate and packet-rate statistics. The default values for each counter is as follows:
		1—30 seconds (60 seconds for VLAN)
		2—300 seconds
		3—not configured

Step 4	**show interface** *interface*
	Example:
	`switch(config)# show interface ethernet 2/2`
	`switch#`
		(Optional) Displays the interface status, which includes the counters.

Step 5	**exit**
	Example:
	`switch(config-if-range)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the three sample intervals for the Ethernet port 3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# load-interval counter 1 60
switch(config-if)# load-interval counter 2 135
switch(config-if)# load-interval counter 3 225
switch(config-if)#
```

Clearing Interface Counters

You can clear the Ethernet and management interface counters by using the **clear counters interface** command. You can perform this task from the configuration mode or interface configuration mode.

SUMMARY STEPS

1. **clear counters interface** [**all** | **ethernet** *slot/port* | **loopback** *number* | **mgmt** *number* | **port channel** *channel-number*]
2. **show interface** *interface*
3. **show interface** [**ethernet** *slot/port* | **port channel** *channel-number*] **counters**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**clear counters interface** [**all** | **ethernet** *slot/port* | **loopback** *number* | **mgmt** *number* | **port channel** *channel-number*]
	Example:
	`switch# clear counters ethernet 2/1`
	`switch#`
		Clears the interface counters.

Step 2	**show interface** *interface*
	Example:
	`switch# show interface ethernet 2/1`
	`switch#`
		(Optional) Displays the interface status.

Step 3	**show interface** [**ethernet** *slot/port* | **port channel** *channel-number*] **counters**
	Example:
	`switch# show interface ethernet 2/1 counters`
	`switch#`
		(Optional) Displays the interface counters.

Example

This example shows how to clear the counters on Ethernet port 5/5:

```
switch# clear counters interface ethernet 5/5
switch#
```

Configuration Example for QSA

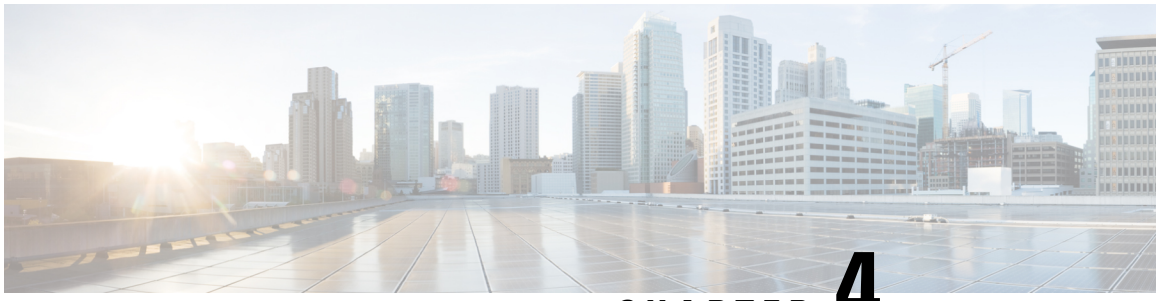
For a Cisco Nexus 9396PX:

- Using the default configuration on port 2/1, all the QSFPs in port group 2/1-6 are brought up with a speed of 40G. If there are any QSA modules in port group 2/1-6, they are error disabled.

- Using the **speed-group** [**10000** | **40000**] command to configure port 2/7, all the QSAs in port group 2/7-12 are brought up with a speed of 10G or 40G. If there are any QSFP modules in port group 2/7-12, they are error disabled.

This example shows how to configure QSA for the first port in the speed group for a Cisco Nexus 9396PX:

```
switch# conf t
switch(config)# interface ethernet 2/7
switch(config-if)# speed-group 10000
```



CHAPTER 4

Configuring Layer 2 Interfaces

This chapter describes how to configure Layer 2 switching ports as access or trunk ports on Cisco NX-OS devices.

Note A Layer 2 port can function as either one of the following:

- A trunk port
- An access port

Note See the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide for information about configuring a SPAN destination interface.

You can configure Layer 2 switching ports as access or trunk ports. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network. All Layer 2 switching ports maintain media access control (MAC) address tables.

Note See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for information about VLANs, MAC address tables, private VLANs, and the Spanning Tree Protocol.

Information About Access and Trunk Interfaces

Note See the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide for complete information on high-availability features.

Note The device supports only IEEE 802.1Q-type VLAN trunk encapsulation.

About Access and Trunk Interfaces

A Layer 2 port can be configured as an access or a trunk port as follows:

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

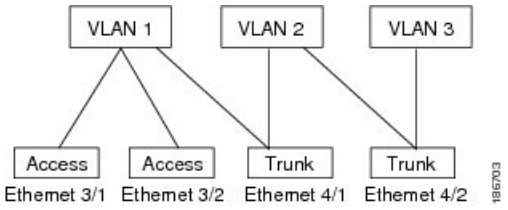
By default, all the ports on Cisco Nexus 9300-EX switches are Layer 3 ports and all the ports on Cisco Nexus 9300 switches are Layer 2 ports.

You can make all ports Layer 2 ports using the setup script or by entering the **system default switchport** command. See the Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide for information about using the setup script. To configure the port as a Layer 2 port using the CLI, use the **switchport** command.

All ports in the same trunk must be in the same VDC, and trunk ports cannot carry VLANs from different VDCs.

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: Trunk and Access Ports and VLAN Traffic



Note See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for information about VLANs.

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 (see the “IEEE 802.1Q Encapsulation” section for more information).

Note See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for information about subinterfaces on Layer 3 interfaces.

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 that it takes the designated port to begin to forward packets.

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.

A Layer 2 interface can function as either an access port or a trunk port; it cannot function as both port types simultaneously.

When you change a Layer 2 interface back to a Layer 3 interface, that interface loses all the Layer 2 configuration and resumes the default VLAN configurations.

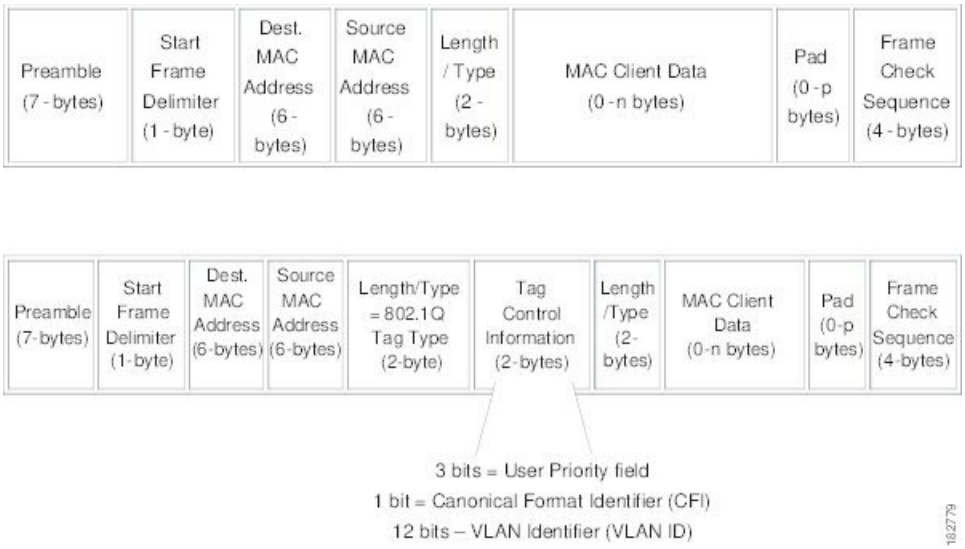
IEEE 802.1Q Encapsulation

Note For information about VLANs, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide.

A trunk is a point-to-point link between the switch 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, or tagging, method that uses a tag that is inserted into the frame header. 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. Also, the encapsulated VLAN tag 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



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 shuts that access port down.

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.

Native VLAN IDs for Trunk Ports

A trunk port can carry nontagged 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. That is, the native VLAN ID is the VLAN that carries untagged traffic on trunk ports.

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

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 You cannot use a Fibre Channel over Ethernet (FCoE) VLAN as a native VLAN for an Ethernet trunk switchport.

Tagging Native VLAN Traffic

The Cisco software supports the IEEE 802.1Q standard on trunk ports. In order to pass untagged traffic through the trunk ports, you must create a VLAN that does not tag any packets (or you can use the default VLAN). Untagged packets can pass through trunk ports and access ports.

However, all packets that enter the device with an 802.1Q tag that matches the value of the native VLAN on the trunk are stripped of any tagging and egress the trunk port as untagged packets. This situation can cause problems because you may want to retain the tagging on packets on the native VLAN for the trunk port.

You can configure the device to drop all untagged packets on the trunk ports and to retain the tagging of packets entering the device with 802.1Q values that are equal to that of the native VLAN ID. All control traffic still passes on the native VLAN. This configuration is global; trunk ports on the device either do or do not retain the tagging for the native VLAN.

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. Later, you can add any specific VLANs 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.

Note See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for more information about STP.

Note You can change the block of VLANs reserved for internal use. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for more information about changing the reserved VLANs.

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, tunnel, and the port-channel interface.

Note A maximum of eight ports can be selected for the default interface. The default interfaces feature is not supported for management interfaces because the device could go to an unreachable state.

Switch Virtual Interface and Autostate Behavior

In Cisco NX-OS, a switch virtual interface (SVI) represents a logical interface between the bridging function and the routing function of a VLAN in the device.

The operational state of this interface is governed by the state of the various ports in its corresponding VLAN. An SVI interface on a VLAN comes up when at least one port in that VLAN is in the Spanning Tree Protocol (STP) forwarding state. Similarly, this interface goes down when the last STP forwarding port goes down or goes to another STP state.

SVI Autostate Enable/Disable

You can also use the SVI for inband management of a device by enabling or disabling the SVI autostate feature. Specifically, you can configure the autostate disable feature to keep an SVI up even if no interface is up in the corresponding VLAN. You can configure this feature for the system (for all SVIs) or for an individual SVI.

High Availability

See the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide for complete information about high availability features.

Virtualization Support

All ports in the same trunk must be in the same VDC, and trunk ports cannot carry VLANs from different VDCs.

Counter Values

See the following information on the configuration, packet size, incremented counter values, and traffic.

Configuration	Packet Size	Incremented Counters	Traffic
L2 port – without any MTU configuration	6400 and 10000	Jumbo, giant, and input error	Dropped
L2 port – with jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Forwarded
L2 port – with jumbo MTU 9216 in network-qos configuration	10000	Jumbo, giant, and input error	Dropped

Configuration	Packet Size	Incremented Counters	Traffic
Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Packets are punted to the CPU (subjected to CoPP configs), get fragmented, and then they are forwarded by the software.
Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Packets are punted to the CPU (subjected to CoPP configs), get fragmented, and then they are forwarded by the software.
Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	10000	Jumbo, giant, and input error	Dropped
Layer 3 port with jumbo Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Forwarded without any fragmentation.
Layer 3 port with jumbo Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	10000	Jumbo, giant, and input error	Dropped
Layer 3 port with jumbo Layer 3 MTU and default L2 MTU configuration	6400 and 10000	Jumbo, giant, and input error	Dropped

Note

- Under 64 bytes packet with good CRC–The short frame counter increments.
- Under 64 bytes packet with bad CRC–The runts counter increments.
- Greater than 64 bytes packet with bad CRC–The CRC counter increments.

Prerequisites for Layer 2 Interfaces

Layer 2 interfaces have the following prerequisites:

- By default, Cisco NX-OS configures Layer 3 parameters. If you want to configure Layer 2 parameters, you need to switch the port mode to Layer 2. You can change the port mode by using the **switchport** command.

- You must configure the port as a Layer 2 port before you can use the **switchport mode** command. By default, all ports on the device are Layer 3 ports. By default, all ports on the Cisco Nexus 9504 and Cisco Nexus 9508 devices are Layer 2 ports.

Guidelines and Limitations for Layer 2 Interfaces

VLAN trunking has the following configuration guidelines and limitations:

- Cisco Nexus 9000 Series switches have the **vlan dot1q tag native** command that can be configured globally. This tags the native VLAN on the configured trunk ports. However, connected switches such as Catalyst 6500 or third-party switches, probably would not have a similar configuration enabled. This could result in unexpected behaviors. Therefore, it is recommended to have the **vlan dot1q tag native** command disabled in case the connected switch does not have it configured.
- Beginning with Release 7.0(3)F3(2), auto-negotiation is not supported on Cisco Nexus 9508 switches with N9K-X9636C-R, N9K-X9636C-RX, and N9K-X9636Q-R lines cards.
- When a store-and-forwarding device receives a packet larger than the MTU size, the packet is dropped. In a cut-through switching mode, packets larger than MTU are transferred.
- Auto-negotiation is not supported on 25-G Ethernet transceiver modules on Cisco Nexus 9200 and 9300-FX platform switches, and Cisco Nexus 9500 platform switches that use N9K-X9700-EX line cards.
- On the Cisco Nexus 9364C switches, auto-negotiation might not work on ports 49-64 when bringing up 100G links using the QSFP-100G-CR4 cable. The workaround for this issue is that you must hard code the speed on ports 49-64 and disable auto-negotiation.
- Autonegotiation (40 G/100 G) and 1 GB with QSA is not supported on the following ports:
  - Cisco Nexus 9336C-FX2 switch: ports 1-6 and 33-36
  - Cisco Nexus 9364C switch: ports 49-66
  - Cisco Nexus 93240YC-FX2 switch: ports 51-54
  - Cisco Nexus 9788TC line card: ports 49-52

Note Peer speed must be set when using copper cables on these ports.

- **show** commands with the **internal** keyword are not supported.
- Starting with Release 7.0(3)I2(1) or later releases on Cisco Nexus 9300 platform switches, a unicast ARP request to SVI is flooded to the other ports within the VLAN.
- ASE2 and ASE3 based Cisco Nexus 9000 Series switches acting as transit switches do not preserve the inner tag for double-tagged packets.

  The following CLI is mandatory only on LSE based Cisco Nexus 9000 Series switches. For seamless packet forwarding and preservation of all VLAN tags on pure transit boxes in the SP cloud that have no Q-in-Q encapsulation or decapsulation requirement, configure the CLI command, **system dot1q-tunnel transit** . To remove the CLI, use **no system dot1q-tunnel transit** CLI command.

The caveats with the CLI that is executed on the switches are:

- L2 frames that egress out of the trunk ports are tagged even on the native VLAN on the port.
- Any other tunneling mechanism, for example, VXLAN and MPLS does not work with the CLI configured.

- A port can be either a Layer 2 or a Layer 3 interface; it cannot be both simultaneously.
- When you change a Layer 3 port to a Layer 2 port or a Layer 2 port to a Layer 3 port, all layer-dependent configuration is lost. When you change an access or trunk port to a Layer 3 port, all information about the access VLAN, native VLAN, allowed VLANs, and so forth, is lost.
- Do not connect devices with access links because access links may partition a VLAN.
- When connecting Cisco devices through an 802.1Q trunk, make sure that the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning tree loops might result.
- Disabling spanning tree on the native VLAN of an 802.1Q trunk without disabling spanning tree on every VLAN in the network can cause spanning tree loops. You must leave spanning tree enabled on the native VLAN of an 802.1Q trunk. If you cannot leave spanning tree enabled, you must disable spanning tree on every VLAN in the network. Make sure that your network has no physical loops before you disable spanning tree.
- When you connect two Cisco devices through 802.1Q trunks, the devices exchange spanning tree bridge protocol data units (BPDUs) on each VLAN allowed on the trunks. The BPDUs on the native VLAN of the trunk are sent untagged to the reserved IEEE 802.1D spanning tree multicast MAC address (01-80-C2-00-00-00). The BPDUs on all other VLANs on the trunk are sent tagged to the reserved Cisco Shared Spanning Tree (SSTP) multicast MAC address (01-00-0c-cc-cc-cd).
- Non-Cisco 802.1Q devices maintain only a single instance of spanning tree (the Mono Spanning Tree) that defines the spanning tree topology for all VLANs. When you connect a Cisco switch to a non-Cisco switch through an 802.1Q trunk, the Mono Spanning Tree of the non-Cisco switch and the native VLAN spanning tree of the Cisco switch combine to form a single spanning tree topology known as the Common Spanning Tree (CST).
- Because Cisco devices transmit BPDUs to the SSTP multicast MAC address on VLANs other than the native VLAN of the trunk, non-Cisco devices do not recognize these frames as BPDUs and flood them on all ports in the corresponding VLAN. Other Cisco devices connected to the non-Cisco 802.1Q cloud receive these flooded BPDUs. This BPDU reception allows Cisco switches to maintain a per-VLAN spanning tree topology across a cloud of non-Cisco 802.1Q devices. The non-Cisco 802.1Q cloud that separates the Cisco devices is treated as a single broadcast segment between all devices connected to the non-Cisco 802.1Q cloud through 802.1Q trunks.
- Make certain that the native VLAN is the same on all of the 802.1Q trunks that connect the Cisco devices to the non-Cisco 802.1Q cloud.
- If you are connecting multiple Cisco devices to a non-Cisco 802.1Q cloud, all of the connections must be through 802.1Q trunks. You cannot connect Cisco devices to a non-Cisco 802.1Q cloud through access ports because doing so places the access port on the Cisco device into the spanning tree “port inconsistent” state and no traffic will pass through the port.
- You can group trunk ports into port-channel groups, but all trunks in the group must have the same configuration. When a group is first created, all ports follow the parameters set for the first port to be added to the group. If you change the configuration of one of these parameters, the device propagates

that setting to all ports in the group, such as the allowed VLANs and the trunk status. For example, if one port in a port group ceases to be a trunk, all ports cease to be trunks.

- If you try to enable 802.1X on a trunk port, an error message appears, and 802.1X is not enabled. If you try to change the mode of an 802.1X-enabled port to trunk, the port mode is not changed.
- Only ingress unicast packet counters are supported for SVI counters.
- When MAC addresses are cleared on a VLAN with the clear mac address-table dynamic command, the dynamic ARP (Address Resolution Protocol) entries on that VLAN are refreshed.
- If a static ARP entry exists on the VLAN and no MAC address to port mapping is present, the supervisor may generate an ARP request to learn the MAC address. Upon learning the MAC address, the adjacency entry points to the correct physical port.
- Cisco NX-OS does not support transparent bridging between two VLANs when one of the SVIs is on the Cisco Nexus 9000 using the BIA MAC (burned-in MAC address). This occurs when the BIA MAC is shared between SVIs/VLANs. A MAC, different from the BIA MAC, can be configured under the SVI for transparent bridging to work properly.

Note This behavior is applicable to Cisco Nexus 9300 Switches (Network Forwarding Engine) and Cisco Nexus 9500 Switches with 95xx,96xx,94xx line cards. This behavior is not applicable to Cisco Nexus 9200 Switches, Cisco Nexus 9300-EX and Cisco Nexus 9500 Switches with 9700-EX line cards.

- Port-local VLANs do not support Fabric Extenders (FEX).
- On Cisco Nexus 9364C switches, auto-negotiation may not work on ports 49-64 when bringing up 100G links using QSFP-100G-CR4 cable. To workaround this issue, you must hard-code the speed on ports 49-64 and disable auto-negotiation.
- Cisco Nexus 9000 switch may fail to connect to a checkpoint with 1G SFP where an autonegotiation enabled Cisco Nexus 93180YC-EX switch is connected. The workaround is to disable autonegotiation and to manually enter the speed and duplex values.

Default Settings for Layer 2 Interfaces

The following table lists the default settings for device access and trunk port mode parameters.

Table 6: Default Access and Trunk Port Mode Parameters

Parameters	Default
Switchport mode	Access
Allowed VLANs	1 to 3967, 4048 to 4094
Access VLAN ID	VLAN1
Native VLAN ID	VLAN1

Parameters	Default
Native VLAN ID tagging	Disabled
Administrative state	Shut

Configuring Access and Trunk Interfaces

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Guidelines for Configuring Access and Trunk Interfaces

All VLANs on a trunk must be in the same VDC.

Configuring a VLAN Interface as a Layer 2 Access Port

You can configure a Layer 2 port as an access port. An access port transmits packets on only one, untagged VLAN. You specify which VLAN traffic that the interface carries, which becomes the access VLAN. If you do not specify a VLAN for an access port, that 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.

Before you begin

Ensure that you are configuring a Layer 2 interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** {{*type slot/port*} | {**port-channel** *number*}}
3. **switchport mode** [**access** | **trunk**]
4. **switchport access vlan** *vlan-id*
5. **exit**
6. **show interface**
7. **no shutdown**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface ethernet** {{*type slot/port*} | {**port-channel** *number*}}
	Example:
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**switchport mode** [**access** | **trunk**] **Example:** `switch(config-if)# switchport mode access`	Sets the interface as a nontrunking nontagged, single-VLAN Layer 2 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	**switchport access vlan** *vlan-id* **Example:** `switch(config-if)# switchport access vlan 5`	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.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface configuration mode.

Step 6	**show interface** **Example:** `switch# show interface`	(Optional) Displays the interface status and information.

Step 7	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set Ethernet 3/1 as a Layer 2 access port that carries traffic for VLAN 5 only:

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

Configuring Access Host Ports

Note You should apply the switchport host command only to interfaces that are connected to an end station.

You can optimize the performance of access ports that are connected to end stations by simultaneously setting that port as an access port. An access host port handles the STP like an edge port and immediately moves to the forwarding state without passing through the blocking and learning states. Configuring an interface as an access host port also disables port channeling on that interface.

Note See “Configuring Port Channels” section and the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for information about port-channel interfaces

Before you begin

Ensure that you are configuring the correct interface to an interface that is an end station.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *type slot/port*
3. **switchport host**
4. **exit**
5. **show interface**
6. **no shutdown**
7. **copy running-config startup-config**

DETAILED STEPS

	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** *type slot/port*
	Example:
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**switchport host** **Example:** switch(config-if)# **switchport host**	Sets the interface to be an access host port, which immediately moves to the spanning tree forwarding state and disables port channeling on this interface. **Note** Apply this command only to end stations.

Step 4	**exit**
	Example:
	`switch(config-if-range)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 5	**show interface** **Example:** `switch# show interface`	(Optional) Displays the interface status and information.

Step 6	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set Ethernet 3/1 as a Layer 2 access port with PortFast enabled and port channel disabled:

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

Configuring Trunk Ports

You can configure a Layer 2 port as a trunk port. A trunk port transmits untagged packets for one VLAN plus encapsulated, tagged, packets for multiple VLANs. (See the “IEEE 802.1Q Encapsulation” section for information about encapsulation.)

Note The device supports 802.1Q encapsulation only.

Before you begin

Before you configure a trunk port, ensure that you are configuring a Layer 2 interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** {*type slot/port* | **port-channel** *number*}
3. **switchport mode** [**access** | **trunk**]
4. **exit**
5. **show interface**
6. **no shutdown**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** {*type slot/port*	**port-channel** *number*}
		Example:
		`switch(config)# interface ethernet 3/1`
		`switch(config-if)#`
			Specifies an interface to configure, and enters interface configuration mode.

Step 3	**switchport mode** [**access**	**trunk**] **Example:** `switch(config-if)# switchport mode trunk`	Sets the interface as a Layer 2 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.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 5	**show interface** **Example:** `switch# show interface`	(Optional) Displays the interface status and information.

	Command or Action	Purpose

Step 6	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set Ethernet 3/1 as a Layer 2 trunk port:

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

Configuring the Native VLAN for 802.1Q Trunking Ports

You can configure the native VLAN for 802.1Q trunk ports. If you do not configure this parameter, the trunk port uses the default VLAN as the native VLAN ID.

Note You cannot configure an FCoE VLAN as a native VLAN for an Ethernet interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** {{*type slot/port*} | {**port-channel** *number*}}
3. **switchport trunk native vlan** *vlan-id*
4. **exit**
5. **show vlan**
6. **no shutdown**
7. **copy running-config startup-config**

DETAILED STEPS

	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** {{*type slot/port*} | {**port-channel** *number*}}
	Example:
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**switchport trunk native vlan** *vlan-id* **Example:** `switch(config-if)# switchport trunk native vlan 5`	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.

Step 4	**exit**
	Example:
	`switch(config-if-range)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 5	**show vlan** **Example:** `switch# show vlan`	(Optional) Displays the status and information of VLANs.

Step 6	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the native VLAN for the Ethernet 3/1, Layer 2 trunk port to VLAN 5:

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

Configuring the Allowed VLANs for Trunking Ports

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

Note The **switchport trunk allowed vlan** *vlan-list* command replaces the current VLAN list on the specified port with the new list. You are prompted for confirmation before the new list is applied.

If you are doing a copy and paste of a large configuration, you might see some failures because the CLI is waiting for a confirmation before accepting other commands. To avoid this problem, you can disable prompting by using the **terminal dont-ask** command before you paste the configuration.

Before you begin

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.

Note You can change the block of VLANs reserved for internal use. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for more information about changing the reserved VLANs.

SUMMARY STEPS

1. **configure terminal**
2. **interface** {**ethernet** *slot/port* | **port-channel** *number*}
3. **switchport trunk allowed vlan** {*vlan-list* **add** *vlan-list* | **all** | **except** *vlan-list* | **none** | **remove** *vlan-list*}
4. **exit**
5. **show vlan**
6. **no shutdown**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** {**ethernet** *slot/port*	**port-channel** *number*} **Example:** `switch(config)# interface ethernet 3/1`	Specifies an interface to configure, and enters interface configuration mode.
Step 3	**switchport trunk allowed vlan** {*vlan-list* **add** *vlan-list*	**all**	**except** *vlan-list*	**none**	**remove** *vlan-list*} **Example:** `switch(config-if)# switchport trunk allowed vlan add 15-20`	Sets the 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. By default, all VLANs are allowed on all trunk interfaces. The default reserved VLANs are 3968 to 4094, and you can change the block of reserved VLANs. See the Cisco

	Command or Action	Purpose
		Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for more information. **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.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 5	**show vlan** **Example:** `switch# show vlan`	(Optional) Displays the status and information for VLANs.

Step 6	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to add VLANs 15 to 20 to the list of allowed VLANs on the Ethernet 3/1, Layer 2 trunk port:

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

Configuring a Default Interface

The default interface feature allows you to clear the existing configuration of multiple interfaces such as Ethernet, loopback, VLAN network, port-channel, and tunnel 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 interface feature is not supported for management interfaces because the device could go to an unreachable state.

If the speed group is configured, the **default interface** command displays the following error:

```
Error: default interface is not supported as speed-group is configured
```

SUMMARY STEPS

1. **configure terminal**
2. **default interface** *int-if* [**checkpoint** *name*]
3. **exit**
4. **show interface**
5. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**default interface** *int-if* [**checkpoint** *name*] **Example:** `switch(config)# default interface ethernet 3/1 checkpoint test8`	Deletes the configuration of the interface and restores the default configuration. Use the **?** keyword to display the supported interfaces. Use the **checkpoint** keyword to store a copy of the running configuration of the interface before clearing the configuration.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch(config)#`
		Exits global configuration mode.

Step 4	**show interface** **Example:** `switch# show interface`	(Optional) Displays the interface status and information.

Step 5	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Example

This example shows how to delete the configuration of an Ethernet interface while saving a checkpoint of the running configuration for rollback purposes:

```
switch# configure terminal
switch(config)# default interface ethernet 3/1 checkpoint test8
.......Done
switch(config)#
```

Configuring SVI Autostate Disable for the System

You can manage an SVI with the SVI autostate feature. You can configure the SVI autostate disable feature to keep an SVI up even if no interface is up in the corresponding VLAN. (Similarly, configure the SVI autostate enable feature so an SVI goes down when no interface is up in the corresponding VLAN). Use this procedure to configure this feature for the entire system.

Note The **system default interface-vlan autostate** command enables the SVI autostate feature.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **system default interface-vlan autostate**
3. **no shutdown**
4. **show running-config** [**all**]

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	[**no**] **system default interface-vlan autostate** **Example:** `switch(config)# no system default interface-vlan autostate`	Disables the default autostate behavior for the device. **Note** Use the **system default interface-vlan autostate** command to enable the autostate behavior for the device.

Step 3	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 4	**show running-config** [**all**]	(Optional) Displays the running configuration.

	Command or Action	Purpose
	Example: `switch(config)# show running-config`	To display the default and configured information, use the **all** keyword.

Example

This example shows how to disable the default autostate behavior on the Cisco NX-OS device:

```
switch# configure terminal
switch(config)# no system default interface-vlan autostate
switch(config)# show running-config
```

Configuring SVI Autostate Disable Per SVI

You can configure SVI autostate enable or disable on individual SVIs. The SVI-level setting overrides the system-level SVI autostate configuration for that particular SVI.

SUMMARY STEPS

1. **configure terminal**
2. **feature interface-vlan**
3. **interface vlan** *vlan-id*
4. [**no**] **autostate**
5. **exit**
6. **show running-config interface vlan** *vlan-id*
7. **no shutdown**
8. **show startup-config interface vlan** *vlan-id*

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**feature interface-vlan** **Example:** `switch(config)# feature interface-vlan`	Enables VLAN interface mode.

Step 3	**interface vlan** *vlan-id*
	Example:
	`switch(config-if)# interface vlan10`
	`switch(config)#`
		Creates a VLAN interface and enters interface configuration mode. The range is from 1 and 4094.

	Command or Action	Purpose
Step 4	[**no**] **autostate** **Example:** `switch(config-if)# no autostate`	By default, enables the SVI autostate feature on specified interface. To disable the default settings, use the **no** form of this command.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface configuration mode.

Step 6	**show running-config interface vlan** *vlan-id* **Example:** `switch(config)# show running-config interface vlan10`	(Optional) Displays the running configuration for the specified VLAN interface.

Step 7	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 8	**show startup-config interface vlan** *vlan-id* **Example:** `switch(config)# show startup-config interface vlan10`	(Optional) Displays the VLAN configuration in the startup configuration.

Example

This example shows how to disable the default autostate behavior on an individual SVI:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan10
witch(config-if)# no autostate
```

Configuring the Device to Tag Native VLAN Traffic

When you are working with 802.1Q trunked interfaces, you can maintain the tagging for all packets that enter with a tag that matches the value of the native VLAN ID and drops all untagged traffic (you will still carry control traffic on that interface). This feature applies to the entire device; you cannot apply it to selected VLANs on a device.

The **vlan dot1q tag native global** command changes the behavior of all native VLAN ID interfaces on all trunks on the device.

Note If you enable 802.1Q tagging on one device and disable it on another device, all traffic is dropped on the device and this feature is disabled. You must configure this feature identically on each device.

SUMMARY STEPS

1. **configure terminal**
2. **vlan dot1q tag native**
3. **exit**
4. **show vlan**
5. **no shutdown**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vlan dot1q tag native** **Example:** `switch(config)# vlan dot1q tag native`	Modifies the behavior of a 802.1Q trunked native VLAN ID interface. The interface maintains the taggings for all packets that enter with a tag that matches the value of the native VLAN ID and drops all untagged traffic. The control traffic is still carried on the native VLAN.

Step 3	**exit**
	Example:
	`switch(config-if-range)# exit`
	`switch(config)#`
		Exits the interface configuration mode.

Step 4	**show vlan** **Example:** `switch# show vlan`	(Optional) Displays the status and information for VLANs.

Step 5	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to change the behavior of the native VLAN on an 802.1Q trunked interface to maintain the tagged packets and drop all untagged traffic (except control traffic):

```
switch# configure terminal
switch(config)# vlan dot1q tag native
switch#
```

Changing the System Default Port Mode to Layer 2

You can set the system default port mode to Layer 2 access ports.

SUMMARY STEPS

1. **configure terminal**
2. **system default switchport** [**shutdown**]
3. **exit**
4. **show interface brief**
5. **no shutdown**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**system default switchport** [**shutdown**] **Example:** switch(config-if)# **system default switchport**	Sets the default port mode for all interfaces on the system to Layer 2 access port mode and enters interface configuration mode. By default, all the interfaces are Layer 3. **Note** When the **system default switchport shutdown** command is issued: • Any FEX HIFs that are not configured with **no shutdown** are shutdown. To avoid the shutdown, configure the FEX HIFs with **no shut** • Any Layer 2 port that is not specifically configured with **no shutdown** are shutdown. To avoid the shutdown, configure the Layer 2 port with **no shut**

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface configuration mode.

Step 4	**show interface brief** **Example:** `switch# show interface brief`	(Optional) Displays the status and information for interfaces.

Step 5	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the system ports to be Layer 2 access ports by default:

```
switch# configure terminal
switch(config-if)# system default switchport
switch(config-if)#
```

Verifying the Interface Configuration

To display access and trunk interface configuration information, perform one of the following tasks.

Command	Purpose
show interface ethernet *slot/port* [**brief** | | **counters** | **debounce** | **description** | **flowcontrol** | **mac-address** | **status** | **transceiver**]	Displays the interface configuration.
show interface brief	Displays interface configuration information, including the mode.
show interface switchport	Displays information, including access and trunk interface, information for all Layer 2 interfaces.
show interface trunk [**module** *module-number* | **vlan** *vlan-id*]	Displays trunk configuration information.

Command	Purpose
show interface capabilities	Displays information about the capabilities of the interfaces.
show running-config [**all**]	Displays information about the current configuration. The **all** command displays the default and current configurations.
show running-config interface ethernet *slot/port*	Displays configuration information about the specified interface.
show running-config interface port-channel *slot/port*	Displays configuration information about the specified port-channel interface.
show running-config interface vlan *vlan-id*	Displays configuration information about the specified VLAN interface.

Monitoring the Layer 2 Interfaces

Use the following commands to display Layer 2 interfaces:

Command	Purpose
clear counters interface [**interface**]	Clears the counters.
load- interval {**interval** *seconds* {**1** | **2** | **3**}}	Cisco Nexus 9000 Series devices set three different sampling intervals to bit-rate and packet-rate statistics.
show interface counters [**module** *module*]	Displays input and output octets unicast packets, multicast packets, and broadcast packets.
show interface counters detailed [**all**]	Displays input packets, bytes, and multicast as well as output packets and bytes.
show interface counters errors [**module** *module*]	Displays information on the number of error packets.

Configuration Examples for Access and Trunk Ports

This example shows how to configure a Layer 2 access interface and assign the access VLAN mode for that interface:

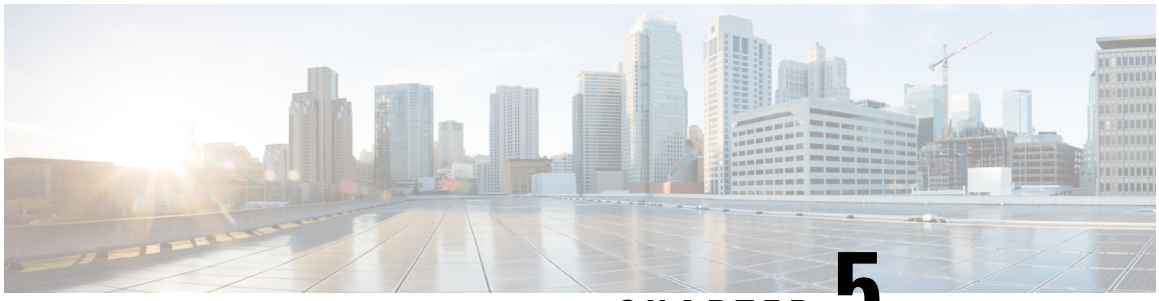
```
switch# configure terminal
switch(config)# interface ethernet 2/30
switch(config-if)# switchport
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
switch(config-if)#
```

This example shows how to configure a Layer 2 trunk interface, assign the native VLAN and the allowed VLANs, and configure the device to tag the native VLAN traffic on the trunk interface:

```
switch# configure terminal
switch(config)# interface ethernet 2/35
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native vlan 10
switch(config-if)# switchport trunk allowed vlan 5, 10
switch(config-if)# exit
switch(config)# vlan dot1q tag native
switch(config)#
```

Related Documents

Related Documents	Document Title
Configuring Layer 3 interfaces	Configuring Layer 2 Interfaces section
Port Channels	Configuring Port Channels section
VLANs, private VLANs, and STP	*Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide*
System management	*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*
High availability	*Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*
Licensing	*Cisco NX-OS Licensing Guide*
Release Notes	*Cisco Nexus 9000 Series NX-OS Release Notes*



CHAPTER 5

Configuring Layer 3 Interfaces

About Layer 3 Interfaces

Layer 3 interfaces forward IPv4 and IPv6 packets to another device using static or dynamic routing protocols. You can use Layer 3 interfaces for IP routing and inter-VLAN routing of Layer 2 traffic.

Routed Interfaces

You can configure a port as a Layer 2 interface or a Layer 3 interface. A routed interface is a physical port that can route IP traffic to another device. A routed interface is a Layer 3 interface only and does not support Layer 2 protocols, such as the Spanning Tree Protocol (STP).

All Ethernet ports are routed interfaces by default. You can change this default behavior with the CLI setup script.

Note The default behavior varies based on the type of switch (Cisco Nexus 9300, Cisco Nexus 9500, or Cisco Nexus 3164).

Note Cisco Nexus 9300 Series switches (except Cisco Nexus 9332 switch) have a Layer 2 default mode.

You can assign an IP address to the port, enable routing, and assign routing protocol characteristics to this routed interface.

You can also create a Layer 3 port channel from routed interfaces. For more information about port channels, see the “Configuring Port Channels” section.

Routed interfaces support exponentially decayed rate counters. Cisco NX-OS tracks the following statistics with these averaging counters:

- Input packets/sec
- Output packets/sec
- Input bytes/sec
- Output bytes/sec

Subinterfaces

You can create virtual subinterfaces on a parent interface configured as a Layer 3 interface. A parent interface can be a physical port.

Subinterfaces divide the parent interface into two or more virtual interfaces on which you can assign unique Layer 3 parameters such as IP addresses and dynamic routing protocols. The IP address for each subinterface should be in a different subnet from any other subinterface on the parent interface.

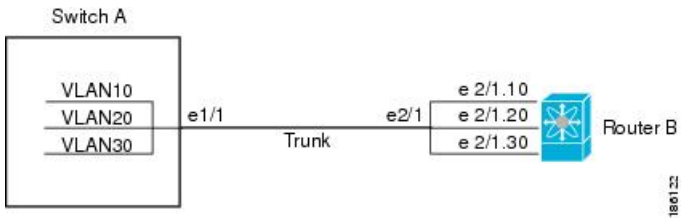
You create a subinterface with a name that consists of the parent interface name (for example, Ethernet 2/1) followed by a period and then by a number that is unique for that subinterface. For example, you could create a subinterface for Ethernet interface 2/1 named Ethernet 2/1.1 where .1 indicates the subinterface.

Cisco NX-OS enables subinterfaces when the parent interface is enabled. You can shut down a subinterface independent of shutting down the parent interface. If you shut down the parent interface, Cisco NX-OS shuts down all associated subinterfaces as well.

One use of subinterfaces is to provide unique Layer 3 interfaces to each virtual local area network (VLAN) supported by the parent interface. In this scenario, the parent interface connects to a Layer 2 trunking port on another device. You configure a subinterface and associate the subinterface to a VLAN ID using 802.1Q trunking.

The following figure shows a trunking port from a switch that connects to router B on interface E 2/1. This interface contains three subinterfaces that are associated with each of the three VLANs carried by the trunking port.

Figure 4: Subinterfaces for VLANs



For more information about VLANs, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide.

VLAN Interfaces

A VLAN interface, or switch virtual interface (SVI), is a virtual routed interface that connects a VLAN on the device to the Layer 3 router engine on the same device. Only one VLAN interface can be associated with a VLAN, but you need to configure a VLAN interface for a VLAN only when you want to route between VLANs or to provide IP host connectivity to the device through a virtual routing and forwarding (VRF) instance that is not the management VRF. When you enable VLAN interface creation, Cisco NX-OS creates a VLAN interface for the default VLAN (VLAN 1) to permit remote switch administration.

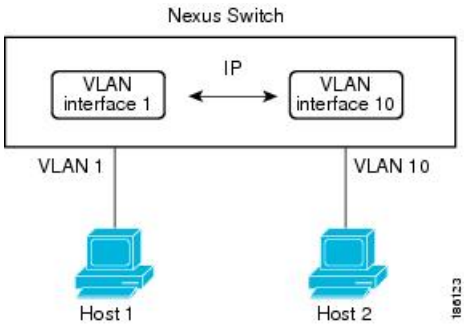
You must enable the VLAN network interface feature before you can see configure it. The system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. See the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide for information on rollbacks and checkpoints.

Note You cannot delete the VLAN interface for VLAN 1.

You can route across VLAN interfaces to provide Layer 3 inter-VLAN routing by configuring a VLAN interface for each VLAN that you want to route traffic to and assigning an IP address on the VLAN interface. For more information about IP addresses and IP routing, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

The following figure shows two hosts connected to two VLANs on a device. You can configure VLAN interfaces for each VLAN that allows Host 1 to communicate with Host 2 using IP routing between the VLANs. VLAN 1 communicates at Layer 3 over VLAN interface 1 and VLAN 10 communicates at Layer 3 over VLAN interface 10.

Figure 5: Connecting Two VLANs with VLAN interfaces



Loopback Interfaces

A loopback interface is a virtual interface with a single endpoint that is always up. Any packet transmitted over a loopback interface is immediately received by this interface. Loopback interfaces emulate a physical interface. You can configure up to 1024 loopback interfaces, numbered 0 to 1023.

You can use loopback interfaces for performance analysis, testing, and local communications. Loopback interfaces can act as a termination address for routing protocol sessions. This loopback configuration allows routing protocol sessions to stay up even if some of the outbound interfaces are down.

High Availability

Layer 3 interfaces support stateful and stateless restarts. After the switchover, Cisco NX-OS applies the runtime configuration after the switchover.

See the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide for complete information about high availability.

Virtualization Support

Layer 3 interfaces support Virtual Routing and Forwarding instances (VRFs). VRFs exist within virtual device contexts (VDCs). By default, Cisco NX-OS places you in the default VDC and default VRF .

Note You must assign an interface to a VRF before you configure the IP address for that interface.

Prerequisites for Layer 3 Interfaces

Layer 3 interfaces have the following prerequisites:

- You are familiar with IP addressing and basic configuration. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about IP addressing.

Guidelines and Limitations

Layer 3 interfaces have the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- If you change a Layer 3 interface to a Layer 2 interface, Cisco NX-OS shuts down the interface, reenables the interface, and removes all configuration specific to Layer 3. (For 6.1(2)I3(4) and earlier)
- If you change a Layer 2 interface to a Layer 3 interface, Cisco NX-OS shuts down the interface, reenables the interface, and deletes all configuration specific to Layer 2. (For 6.1(2)I3(4) and earlier)
- Configuring a subinterface on a port-channel interface is not supported. (For 6.1(2)I3(4) and earlier)
- The Dynamic Host Configuration Protocol (DHCP) option is not supported when configuring a subinterface on a port-channel interface.
- Layer 3 routed traffic for missing Layer 2 adjacency information is not flooded back onto VLAN members of ingress units when the source MAC address of routed traffic is a non-VDC (Virtual Device Context) MAC address. This limitation is for hardware flood traffic and can occur when the SVI (Switched Virtual Interface) has a user-configured MAC address.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings

The following table lists the default settings for Layer 3 interface parameters.

Table 7: Default Layer 3 Interface Parameters

Parameters	Default
Admin state	Shut

Configuring Layer 3 Interfaces

Configuring a Routed Interface

You can configure any Ethernet port as a routed interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **no switchport**
4. [**ip address** *ip-address/length* | **ipv6 address** *ipv6-address/length*]
5. **show interfaces**
6. **no shutdown**
7. **copy running-config startup-config**

DETAILED STEPS

	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 2/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**no switchport**	Configures the interface as a Layer 3 interface.

	Command or Action	Purpose
	Example: `switch(config-if)# no switchport`	

Step 4	[**ip address** *ip-address/length* | **ipv6 address** *ipv6-address/length*]
	Example:
	`switch(config-if)# ip address 192.0.2.1/8`
	Example:
	`switch(config-if)# ipv6 address 2001:0DB8::1/8`
		• Configures an IP address for this interface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about IP addresses.
		• Configures an IPv6 address for this interface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about IPv6 addresses.

Step 5	**show interfaces** **Example:** `switch(config-if)# show interfaces ethernet 2/1`	(Optional) Displays the Layer 3 interface statistics.

Step 6	**no shutdown**
	Example:
	`switch#`
	`switch(config-if)# int e2/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the configuration change.

Example

- Use the **medium** command to set the interface medium to either point to point or broadcast.

Command	Purpose
medium {**broadcast** | **p2p**} Example: `switch(config-if)# medium p2p medium p2p`	Configures the interface medium as either point to point or broadcast.

Note The default setting is **broadcast**, and this setting does not appear in any of the **show** commands. However, if you do change the setting to **p2p**, you will see this setting when you enter the **show running config** command.

- Use the **switchport** command to convert a Layer 3 interface into a Layer 2 interface.

Command	Purpose
switchport Example: `switch(config-if)# switchport`**`switchport`**	Configures the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.

- This example shows how to configure a routed interface:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# no switchport
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

The default setting for interfaces is routed. If you want to configure an interface for Layer 2, enter the **switchport** command. Then, if you change a Layer 2 interface to a routed interface, enter the **no switchport** command.

Configuring a Subinterface on a Routed Interface

You can configure one or more subinterfaces on a routed interface made from routed interfaces.

Before you begin

Configure the parent interface as a routed interface.

See the "Configuring a Routed Interface" section.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port.number*
3. [**ip address** *ip-address/length* | **ipv6 address** *ipv6-address/length*]
4. **encapsulation dot1Q** *vlan-id*
5. **show interfaces**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot/port.number*
	Example:
	`switch(config)# interface ethernet 2/1.1`
	`switch(config-subif)#`
		Creates a subinterface and enters subinterface configuration mode. The number range is from 1 to 4094.

	Command or Action	Purpose

Step 3	[**ip address** *ip-address/length* | **ipv6 address** *ipv6-address/length*]
	Example:
	`switch(config-subif)# ip address 192.0.2.1/8`
	Example:
	`switch(config-subif)# ipv6 address 2001:0DB8::1/8`
		• Configures an IP address for this subinterface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information on IP addresses.
		• Configures an IPv6 address for this subinterface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information on IPv6 addresses.

Step 4	**encapsulation dot1Q** *vlan-id* **Example:** `switch(config-subif)# encapsulation dot1Q 33`	Configures IEEE 802.1Q VLAN encapsulation on the subinterface. The range is from 2 to 4093.
Step 5	**show interfaces** **Example:** `switch(config-subif)# show interfaces ethernet 2/1.1`	(Optional) Displays the Layer 3 interface statistics.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the configuration change.

Example

- This example shows how to create a subinterface:

```
switch# configure terminal
switch(config)# interface ethernet 2/1.1
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# encapsulation dot1Q 33
switch(config-if)# copy running-config startup-config
```

- The output of the **show interface eth** command is enhanced for the subinterfaces as shown in the following :

```
switch# show interface ethernet 1/2.1
Ethernet1/2.1 is down (Parent Interface Admin down)
admin state is down, Dedicated Interface, [parent interface is Ethernet1/2]
Hardware: 40000 Ethernet, address: 0023.ac67.9bc1 (bia 4055.3926.61d4)
Internet Address is 10.10.10.1/24
MTU 1500 bytes, BW 40000000 Kbit, DLY 10 usec
reliability 255/255, txload 1/255, rxload 1/255
Auto-mdix is turned off
EtherType is 0x8100
L3 in Switched:
 ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
L3 out Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
```

Configuring a VLAN Interface

You can create VLAN interfaces to provide inter-VLAN routing.

SUMMARY STEPS

1. **configure terminal**
2. **feature interface-vlan**
3. **interface vlan** *number*
4. [**ip address** *ip-address/length* | **ipv6 address** *ipv6-address/length*]
5. **show interface vlan** *number*
6. **no shutdown**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**feature interface-vlan** **Example:** `switch(config)# feature interface-vlan`	Enables VLAN interface mode.

Step 3	**interface vlan** *number*
	Example:
	`switch(config)# interface vlan 10`
	`switch(config-if)#`
		Creates a VLAN interface. The number range is from 1 to 4094.

Step 4	[**ip address** *ip-address/length* | **ipv6 address** *ipv6-address/length*]
	Example:
	`switch(config-if)# ip address 192.0.2.1/8`
	Example:
	`switch(config-if)# ipv6 address 2001:0DB8::1/8`
		• Configures an IP address for this VLAN interface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information on IP addresses.
		• Configures an IPv6 address for this VLAN interface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information on IPv6 addresses.

Step 5	**show interface vlan** *number* **Example:** `switch(config-if)# show interface vlan 10`	(Optional) Displays the Layer 3 interface statistics.
Step 6	**no shutdown** **Example:**	(Optional) Clears the errors on the interfaces where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up.

	Command or Action	Purpose
	`switch(config)# int e3/1` `switch(config)# no shutdown`	If policies do not correspond, the errors are placed in an error-disabled policy state.
Step 7	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Example

This example shows how to create a VLAN interface:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan 10
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

Configuring a Loopback Interface

You can configure a loopback interface to create a virtual interface that is always up.

Before you begin

Ensure that the IP address of the loopback interface is unique across all routers on the network.

SUMMARY STEPS

1. **configure terminal**
2. **interface loopback** *instance*
3. [**ip address** *ip-address/length* | **ipv6 address** *ipv6-address/length*]
4. **show interface loopback** *instance*
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface loopback** *instance*
	Example:
	`switch(config)# interface loopback 0`
	`switch(config-if)#`
		Creates a loopback interface. The range is from 0 to 1023.

	Command or Action	Purpose

Step 3	[**ip address** *ip-address/length* | **ipv6 address** *ipv6-address/length*]
	Example:
	`switch(config-if)# ip address 192.0.2.1/8`
	Example:
	`switch(config-if)# ipv6 address 2001:0DB8::1/8`
		• Configures an IP address for this interface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about IP addresses.
		• Configures an IPv6 address for this interface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about IPv6 addresses.

Step 4	**show interface loopback** *instance* **Example:** `switch(config-if)# show interface loopback 0`	(Optional) Displays the loopback interface statistics.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Example

This example shows how to create a loopback interface:

```
switch# configure terminal
switch(config)# interface loopback 0
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

Assigning an Interface to a VRF

You can add a Layer 3 interface to a VRF.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type number*
3. **vrf member** *vrf-name*
4. **ip address** *ip-prefix/length*
5. **show vrf** [*vrf-name*] **interface** *interface-type number*
6. **copy running-config startup-config**

DETAILED STEPS

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

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

Step 2	**interface** *interface-type number*
	Example:
	`switch(config)# interface loopback 0`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member RemoteOfficeVRF`	Adds this interface to a VRF.
Step 4	**ip address** *ip-prefix/length* **Example:** `switch(config-if)# ip address 192.0.2.1/16`	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 5	**show vrf** [*vrf-name*] **interface** *interface-type number* **Example:** `switch(config-vrf)# show vrf Enterprise interface loopback 0`	(Optional) Displays VRF information.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Example

This example shows how to add a Layer 3 interface to the VRF:

```
switch# configure terminal
switch(config)# interface loopback 0
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 209.0.2.1/16
switch(config-if)# copy running-config startup-config
```

Configuring a DHCP Client on an Interface

You can configure the DHCP client on an SVI, a management interface, or a physical Ethernet interface for IPv4 or IPv6 address

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *type slot/port* | **mgmt** *mgmt-interface-number* | **vlan** *vlan id*
3. switch(config-if)# [**no**] **ipv6 address use-link-local-only**
4. switch(config-if)# [**no**] [**ip** | **ipv6] address dhcp**

5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *type slot/port* | **mgmt** *mgmt-interface-number* | **vlan** *vlan id*	Creates a physical Ethernet interface, a management interface, or a VLAN interface. The range of *vlan id* is from 1 to 4094.
Step 3	switch(config-if)# [**no**] **ipv6 address use-link-local-only**	Prepares for request to the DHCP server. **Note** This command is only required for an IPv6 address.
Step 4	switch(config-if)# [**no**] [**ip** | **ipv6]** **address dhcp**	Requests the DHCP server for an IPv4 or IPv6 address. The **no** form of this command removes any address that was acquired.
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 configure the IP address of a DHCP client on an SVI:

```
switch# configure terminal
switch(config)# interface vlan 15
switch(config-if)# ip address dhcp
```

This example shows how to configure an IPv6 address of a DHCP client on a management interface:

```
switch# configure terminal
switch(config)# interface mgmt 0
switch(config-if)# ipv6 address use-link-local-only
switch(config-if)# ipv6 address dhcp
```

Verifying the Layer 3 Interfaces Configuration

To display the Layer 3 configuration, perform one of the following tasks:

Command	Purpose
show interface ethernet *slot/port*	Displays the Layer 3 interface configuration, status, and counters (including the 5-minute exponentially decayed moving average of inbound and outbound packet and byte rates).

Command	Purpose
show interface ethernet *slot/port* **brief**	Displays the Layer 3 interface operational status.
show interface ethernet *slot/port* **capabilities**	Displays the Layer 3 interface capabilities, including port type, speed, and duplex.
show interface ethernet *slot/port* **description**	Displays the Layer 3 interface description.
show interface ethernet *slot/port* **status**	Displays the Layer 3 interface administrative status, port mode, speed, and duplex.
show interface ethernet *slot/port.number*	Displays the subinterface configuration, status, and counters (including the f-minute exponentially decayed moving average of inbound and outbound packet and byte rates).
show interface port-channel *channel-id.number*	Displays the port-channel subinterface configuration, status, and counters (including the 5-minute exponentially decayed moving average of inbound and outbound packet and byte rates).
show interface loopback *number*	Displays the loopback interface configuration, status, and counters.
show interface loopback *number* **brief**	Displays the loopback interface operational status.
show interface loopback *number* **description**	Displays the loopback interface description.
show interface loopback *number* **status**	Displays the loopback interface administrative status and protocol status.
show interface vlan *number*	Displays the VLAN interface configuration, status, and counters.
show interface vlan *number* **brief**	Displays the VLAN interface operational status.
show interface vlan *number* **description**	Displays the VLAN interface description.
show interface vlan *number* **status**	Displays the VLAN interface administrative status and protocol status.

Monitoring the Layer 3 Interfaces

Use the following commands to display Layer 3 statistics:

Command	Purpose
load- interval {**interval** *seconds* {**1** | **2** | **3**}}	Cisco Nexus 9000 Series devices set three different sampling intervals to bit-rate and packet-rate statistics. The range for VLAN network interface is 60 to 300 seconds, and the range for Layer interfaces is 30 to 300 seconds.
show interface ethernet *slot/port* **counters**	Displays the Layer 3 interface statistics (unicast, multicast, and broadcast).
show interface ethernet *slot/port* **counters brief**	Displays the Layer 3 interface input and output counters.
show interface ethernet errors *slot/port* **detailed** [**all**]	Displays the Layer 3 interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).
show interface ethernet errors *slot/port* **counters errors**	Displays the Layer 3 interface input and output errors.
show interface ethernet errors *slot/port* **counters snmp**	Displays the Layer 3 interface counters reported by SNMP MIBs.
show interface ethernet *slot/port.number* **counters**	Displays the subinterface statistics (unicast, multicast, and broadcast).
show interface port-channel *channel-id.number* **counters**	Displays the port-channel subinterface statistics (unicast, multicast, and broadcast).
show interface loopback *number* **counters**	Displays the loopback interface input and output counters (unicast, multicast, and broadcast).
show interface loopback *number* **detailed** [**all**]	Displays the loopback interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).
show interface loopback *number* **counters errors**	Displays the loopback interface input and output errors.
show interface vlan *number* **counters**	Displays the VLAN interface input and output counters (unicast, multicast, and broadcast).
show interface vlan *number* **counters detailed** [**all**]	Displays the VLAN interface statistics. You can optionally include all Layer 3 packet and byte counters (unicast and multicast).
show interface vlan *number* **counters snmp**	Displays the VLAN interface counters reported by SNMP MIBs.

Configuration Examples for Layer 3 Interfaces

This example shows how to configure Ethernet subinterfaces:

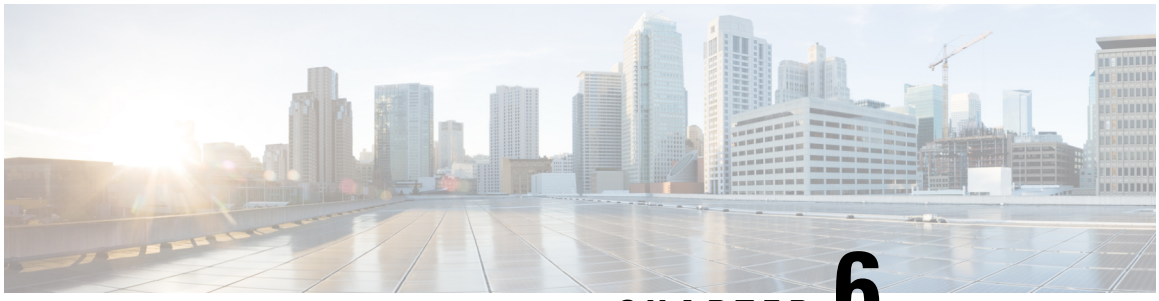
```
interface ethernet 2/1.10
description Layer 3
ip address 192.0.2.1/8
```

This example shows how to configure a loopback interface:

```
interface loopback 3
ip address 192.0.2.2/32
```

Related Documents

Related Documents	Document Title
IP	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*
VLANs	*Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide*



CHAPTER 6

Configuring Bidirectional Forwarding Detection

About BFD

BFD is a detection protocol designed to provide fast forwarding-path failure detection times for media types, encapsulations, topologies, and routing protocols. You can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different protocol hello mechanisms. BFD makes network profiling and planning easier and reconvergence time consistent and predictable.

BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules.

Asynchronous Mode

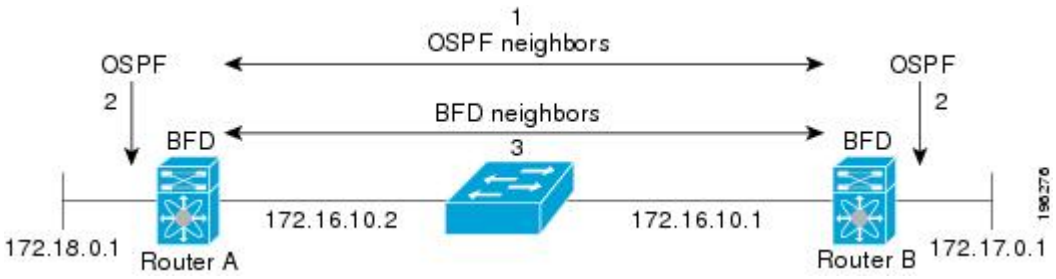
Cisco NX-OS supports the BFD asynchronous mode, which sends BFD control packets between two adjacent devices to activate and maintain BFD neighbor sessions between the devices. You configure BFD on both devices (or BFD neighbors). Once BFD has been enabled on the interfaces and on the appropriate protocols, Cisco NX-OS creates a BFD session, negotiates BFD session parameters, and begins to send BFD control packets to each BFD neighbor at the negotiated interval. The BFD session parameters include the following:

- Desired minimum transmit interval—The interval at which this device wants to send BFD hello messages.
- Required minimum receive interval—The minimum interval at which this device can accept BFD hello messages from another BFD device.

- Detect multiplier—The number of missing BFD hello messages from another BFD device before this local device detects a fault in the forwarding path.

The following figure shows how a BFD session is established. The figure shows a simple network with two routers running Open Shortest Path First (OSPF) and BFD. When OSPF discovers a neighbor (1), it sends a request to the local BFD process to initiate a BFD neighbor session with the OSPF neighbor router (2). The BFD neighbor session with the OSPF neighbor router is now established (3).

Figure 6: Establishing a BFD Neighbor Relationship



BFD Detection of Failures

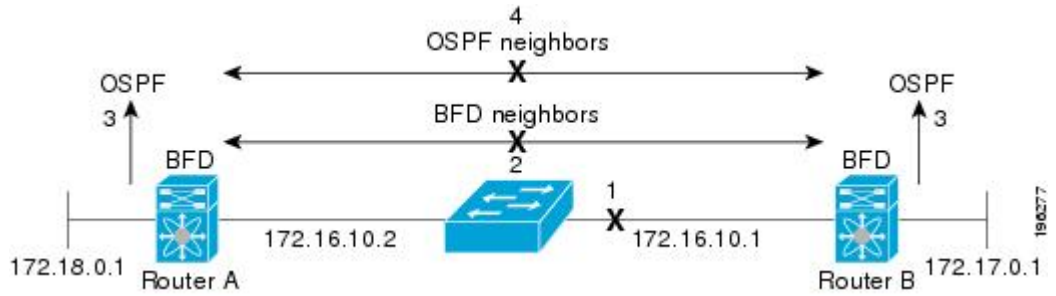
Once a BFD session has been established and timer negotiations are complete, BFD neighbors send BFD control packets that act in the same manner as an IGP hello protocol to detect liveliness, except at a more accelerated rate. BFD detects a failure, but the protocol must take action to bypass a failed peer.

BFD sends a failure detection notice to the BFD-enabled protocols when it detects a failure in the forwarding path. The local device can then initiate the protocol recalculation process and reduce the overall network convergence time.

The following figure shows what happens when a failure occurs in the network (1). The BFD neighbor session with the OSPF neighbor router is torn down (2). BFD notifies the local OSPF process that the BFD neighbor is no longer reachable (3). The local OSPF process tears down the OSPF neighbor relationship (4). If an alternative path is available, the routers immediately start converging on it.

Note Note The BFD failure detection occurs in less than a second, which is much faster than OSPF Hello messages could detect the same failure.

Figure 7: Tearing Down an OSPF Neighbor Relationship



Distributed Operation

Cisco NX-OS can distribute the BFD operation to compatible modules that support BFD. This process offloads the CPU load for BFD packet processing to the individual modules that connect to the BFD neighbors. All BFD session traffic occurs on the module CPU. The module informs the supervisor when a BFD failure is detected.

BFD Echo Function

The BFD echo function sends echo packets from the forwarding engine to the remote BFD neighbor. The BFD neighbor forwards the echo packet back along the same path in order to perform detection; the BFD neighbor does not participate in the actual forwarding of the echo packets. The echo function and the forwarding engine are responsible for the detection process. BFD can use the slow timer to slow down the asynchronous session when the echo function is enabled and reduce the number of BFD control packets that are sent between two BFD neighbors. Also, the forwarding engine tests the forwarding path on the remote (neighbor) system without involving the remote system, so there is less interpacket delay variability and faster failure detection times.

The echo function is without asymmetry when both BFD neighbors are running echo function.

Security

Cisco NX-OS uses the packet Time to Live (TTL) value to verify that the BFD packets came from an adjacent BFD peer. For all asynchronous and echo request packets, the BFD neighbor sets the TTL value to 255 and the local BFD process verifies the TTL value as 255 before processing the incoming packet. For the echo response packet, BFD sets the TTL value to 254.

You can configure SHA-1 authentication of BFD packets.

High Availability

BFD supports stateless restarts. After a reboot or supervisor switchover, Cisco NX-OS applies the running configuration and BFD immediately sends control packets to the BFD peers.

Virtualization Support

BFD supports virtual routing and forwarding instances (VRFs). VRFs exist within virtual device contexts (VDCs). By default, Cisco NX-OS places you in the default VDC and default VRF.

Prerequisites for BFD

BFD has the following prerequisites:

- You must enable the BFD feature.
- Disable Internet Control Message Protocol (ICMP) redirect messages on BFD-enabled interfaces.
- Disable the IP packet verification check for identical IP source and destination addresses.
- See other detailed prerequisites that are listed with the configuration tasks.

Guidelines and Limitations

BFD has the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- Starting with Release 7.0(3)I5(1), BFD per-member link support is added on Cisco Nexus 9000 Series switches.
- BFD supports BFD version 1.
- BFD supports IPv4 and IPv6.
- BFD supports OSPFv3.
- BFD supports IS-ISv6.
- BFD supports BGPv6.
- BFD supports EIGRPv6.
- BFD supports only one session per address family, per Layer 3 interface.
- BFD supports only sessions which have unique (src_ip, dst_ip, interface/vrf) combination.
- BFD supports single-hop BFD.
- BFD for BGP supports single-hop EBGP and iBGP peers.
- BFD supports keyed SHA-1 authentication.
- BFD supports the following Layer 3 interfaces—physical interfaces, port channels, subinterfaces, and VLAN interfaces.
- BFD depends on a Layer 3 adjacency information to discover topology changes, including Layer 2 topology changes. A BFD session on a VLAN interface (SVI) may not be up after the convergence of the Layer 2 topology if there is no Layer 3 adjacency information available.
- For BFD on a static route between two devices, both devices must support BFD. If one or both of the devices do not support BFD, the static routes are not programmed in the Routing Information Base (RIB).
- Port channel configuration limitations:
  - For Layer 3 port channels used by BFD, you must enable LACP on the port channel.
  - For Layer 2 port channels used by SVI sessions, you must enable LACP on the port channel.
- SVI limitations:
  - An ASIC reset causes traffic disruption for other ports and it can cause the SVI sessions on the other ports to flap. For example, if the carrier interface is a virtual port channel (vPC), BFD is not supported over the SVI interface and it could cause a trigger for an ASIC reset. When a BFD session is over SVI using virtual port channel (vPC) Peer-Link, the BFD echo function is not supported. You must disable the BFD echo function for all sessions over SVI between vPC peer nodes.

    An SVI on the Cisco Nexus series switches should not be configured to establish a BFD neighbor adjacency with a device connected to it via a vPC. This is because the BFD keepalives from the

neighbour, if sent over the vPC member link connected to the vPC peer-switch, do not reach this SVI causing the BFD adjacency to fail.

- When you change the topology (for example, add or delete a link into a VLAN, delete a member from a Layer 2 port channel, and so on), the SVI session could be affected. It may go down first and then come up after the topology discovery is finished.
- BFD over FEX HIF interfaces is not supported.
- When a BFD session is over SVI using virtual port-channel (vPC) Peer-Link (either BCM or GEM based ports), the BFD echo function is not supported. You must disable the BFD echo function for all sessions over SVI between vPC peer nodes using the **no bfd echo** command at the SVI configuration level.

Tip If you do not want the SVI sessions to flap and you need to change the topology, you can disable the BFD feature before making the changes and reenable BFD after the changes have been made. You can also configure the BFD timer to be a large value (for example, 5 seconds), and change it back to a fast timer after the above events complete.

- When you configure the BFD Echo function on the distributed Layer 3 port channels, reloading a member module flaps the BFD session hosted on that module, which results in a packet loss.

  If you connect the BFD peers directly without a Layer 2 switch in between, you can use the BFD per-link mode as an alternative solution.

Note Using BFD per-link mode and subinterface optimization simultaneously on a Layer 3 port channel is not supported.

- When you specify a BFD neighbor prefix in the **clear** {**ip** | **ipv6**} **route** *prefix* command, the BFD echo session will flap.
- The **clear** {**ip** | **ipv6**} **route *** command causes BFD echo sessions to flap.
- HSRP for IPv4 is supported with BFD.
- BFD packets generated by the Cisco NX-OS device linecards are sent with COS 6/DSCP CS6. The DSCP/COS values for BFD packets are not user configurable.
- When configuring BFDv6 in no-bfd-echo mode, it is recommended to run with timers of 150 ms with a multiplier of 3.
- BFDv6 is not supported for VRRPv3 and HSRP for v6.
- IPv6 **eigrp bfd** cannot be disabled on an interface.
- Port channel configuration notes:
  - When the BFD per-link mode is configured, the BFD echo function is not supported. You must disable the BFD echo function using the **no bfd echo** command before configuring the **bfd per-link** command.

- Configuring BFD per-link with link-local is not supported.

Default Settings

The following table lists the default settings for BFD parameters.

Table 8: Default BFD Parameters

Parameters	Default
BFD feature	Disabled
Required minimum receive interval	50 milliseconds
Desired minimum transmit interval	50 milliseconds
Detect multiplier	3
Echo function	Enabled
Mode	Asynchronous
Port-channel	Logical mode (one session per source-destination pair address)
Slow timer	2000 milliseconds

Configuring BFD

Configuration Hierarchy

You can configure BFD at the global level and at the interface level. The interface configuration overrides the global configuration.

For physical ports that are members of a port channel, the member port inherits the primary port channel BFD configuration.

Task Flow for Configuring BFD

Follow these steps in the following sections to configure BFD:

- Enabling the BFD Feature.
- Configuring Global BFD Parameters or Configuring BFD on an Interface.

Enabling the BFD Feature

You must enable the BFD feature before you can configure BFD on an interface and protocol.

Note Use the **no feature bfd** command to disable the BFD feature and remove all associated configuration.

Command	Purpose
no feature bfd **Example:** `switch(config)# no feature bfd`	Disables the BFD feature and removes all associated configuration.

SUMMARY STEPS

1. **configure terminal**
2. **feature bfd**
3. **show feature | include bfd**
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**feature bfd** **Example:** `switch(config)# feature bfd`	Enables the BFD feature.
Step 3	**show feature | include bfd** **Example:** `switch(config)# show feature | include bfd`	(Optional) Displays enabled and disabled features.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the configuration change.

Configuring Global BFD Parameters

You can configure the BFD session parameters for all BFD sessions on the device. The BFD session parameters are negotiated between the BFD peers in a three-way handshake.

See the Configuring BFD on an Interface section to override these global session parameters on an interface.

Before you begin

Enable the BFD feature.

SUMMARY STEPS

1. **configure terminal**
2. **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*
3. **bfd slow-timer** [*interval*]
4. **bfd echo-interface loopback** *interface number*
5. **show running-config bfd**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**bfd interval** *mintx* **min_rx** *msec* **multiplier** *value* **Example:** `switch(config)# bfd interval 50 min_rx 50 multiplier 3`	Configures the BFD session parameters for all BFD sessions on the device. This command overrides these values by configuring the BFD session parameters on an interface. The *mintx* and *msec* range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.
Step 3	**bfd slow-timer** [*interval*] **Example:** `switch(config)# bfd slow-timer 2000`	Configures the slow timer used in the echo function. This value determines how fast BFD starts up a new session and at what speed the asynchronous sessions use for BFD control packets when the echo function is enabled. The slow-timer value is used as the new control packet interval, while the echo packets use the configured BFD intervals. The echo packets are used for link failure detection, while the control packets at the slower rate maintain the BFD session. The range is from 1000 to 30000 milliseconds. The default is 2000.
Step 4	**bfd echo-interface loopback** *interface number* **Example:** `switch(config)# bfd echo-interface loopback 1 3`	Configures the interface used for Bidirectional Forwarding Detection (BFD) echo frames. This command changes the source address for the echo packets to the one configured on the specified loopback interface. The interface number range is from 0 to 1023.

	Command or Action	Purpose
Step 5	**show running-config bfd** **Example:** `switch(config)# show running-config bfd`	(Optional) Displays the BFD running configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the configuration change.

Configuring BFD on an Interface

You can configure the BFD session parameters for all BFD sessions on an interface. The BFD session parameters are negotiated between the BFD peers in a three-way handshake.

This configuration overrides the global session parameters for the configured interface.

Before you begin

Ensure that Internet Control Message Protocol (ICMP) redirect messages are disabled on BFD-enabled interfaces. Use the **no ip redirects** command or the **no ipv6 redirects** command on the interface.

Enable the BFD feature. See the Enabling the BFD Feature section.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *int-if*
3. **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*
4. **bfd authentication keyed-sha1 keyid** *id* **key** *ascii_key*
5. **show running-config bfd**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *int-if*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode. Use the ? keyword to display the supported interfaces.

	Command or Action	Purpose
Step 3	**bfd interval** *mintx* **min_rx** *msec* **multiplier** *value* **Example:** `switch(config-if)# bfd interval 50 min_rx 50 multiplier 3`	Configures the BFD session parameters for all BFD sessions on the device. This command overrides these values by configuring the BFD session parameters on an interface. The *mintx* and *msec* range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3. Beginning with Cisco NX-OS Release 9.3(5), configuring BFD session parameters under interface with default timer values using the **bfd interval 50 min_rx 50 multiplier 3** command is functionally equivalent to **no bfd interval** command. Once BFD session parameters under interface are set to default values, those BFD sessions running on that interface will inherit global session parameters, if present.
Step 4	**bfd authentication keyed-sha1 keyid** *id* **key** *ascii_key* **Example:** `switch(config-if)# bfd authentication keyed-sha1 keyid 1 ascii_key cisco123`	(Optional) Configures SHA-1 authentication for all BFD sessions on the interface. The *ascii_key* string is a secret key shared among BFD peers. The *id* value, a number between 0 and 255, is assigned to this particular *ascii_key*. BFD packets specify the key by *id*, allowing the use of multiple active keys. To disable SHA-1 authentication on the interface, use the **no** form of the command.
Step 5	**show running-config bfd** **Example:** `switch(config-if)# show running-config bfd`	(Optional) Displays the BFD running configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Configuring BFD on a Port Channel

You can configure the BFD session parameters for all BFD sessions on a port channel. If per-link mode is used for Layer 3 port channels, BFD creates a session for each link in the port channel and provides an aggregate result to client protocols. For example, if the BFD session for one link on a port channel is up, BFD informs client protocols, such as OSPF, that the port channel is up. The BFD session parameters are negotiated between the BFD peers in a three-way handshake.

This configuration overrides the global session parameters for the configured port channel. The member ports of the port channel inherit the port channel BFD session parameters.

Before you begin

Ensure that you enable LACP on the port channel before you enable BFD.

Ensure that Internet Control Message Protocol (ICMP) redirect messages are disabled on BFD-enabled interfaces. Use the **no ip redirects** command on the interface.

Enable the BFD feature. See the Enabling the BFD Feature section.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*
3. **bfd per-link**
4. **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*
5. **bfd authentication keyed-sha1 keyid** *id* **key** *ascii_key*
6. **show running-config bfd**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 2`
	`switch(config-if)#`
		Enters port-channel configuration mode. Use the **?** keyword to display the supported number range.

Step 3	**bfd per-link** **Example:** `switch(config-if)# bfd per-link`	Configures the BFD sessions for each link in the port channel.
Step 4	**bfd interval** *mintx* **min_rx** *msec* **multiplier** *value* **Example:** `switch(config-if)# bfd interval 50 min_rx 50 multiplier 3`	(Optional) Configures the BFD session parameters for all BFD sessions on the port channel. This command overrides these values by configuring the BFD session parameters. The *mintx* and *msec* range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.
Step 5	**bfd authentication keyed-sha1 keyid** *id* **key** *ascii_key* **Example:** `switch(config-if)# bfd authentication keyed-sha1 keyid 1 ascii_key cisco123`	(Optional) Configures SHA-1 authentication for all BFD sessions on the interface. The *ascii_key* string is a secret key shared among BFD peers. The *id* value, a number between 0 and 255, is assigned to this particular *ascii_key*. BFD packets specify the key by *id*, allowing the use of multiple active keys. To disable SHA-1 authentication on the interface, use the **no** form of the command.

	Command or Action	Purpose
Step 6	**show running-config bfd** **Example:** `switch(config-if)# show running-config bfd`	(Optional) Displays the BFD running configuration.
Step 7	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Configuring the BFD Echo Function

You can configure the BFD echo function on one or both ends of a BFD-monitored link. The echo function slows down the required minimum receive interval, based on the configured slow timer. The RequiredMinEchoRx BFD session parameter is not set to zero if the echo function is disabled in compliance with RFC 5880. The slow timer becomes the required minimum receive interval if the echo function is enabled.

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section on or the Configuring BFD on an Interface section.

Ensure that Internet Control Message Protocol (ICMP) redirect messages are disabled on BFD-enabled interfaces. Use the **no ip redirects** command on the interface.

Ensure that the IP packet verification check for identical IP source and destination addresses is disabled. Use the **no hardware ip verify address identical** command. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about this command.

SUMMARY STEPS

1. **configure terminal**
2. **bfd slow-timer** *echo-interval*
3. **interface** *int-if*
4. **bfd echo**
5. **show running-config bfd**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**bfd slow-timer** *echo-interval* **Example:** `switch(config)# bfd slow-timer 2000`	Configures the slow timer used in the echo function. This value determines how fast BFD starts up a new session and is used to slow down the asynchronous sessions when the BFD echo function is enabled. This value overwrites the required minimum receive interval when the echo function is enabled. The range is from 1000 to 30000 milliseconds. The default is 2000.

Step 3	**interface** *int-if*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode. Use the ? keyword to display the supported interfaces.

Step 4	**bfd echo** **Example:** `switch(config-if)# bfd echo`	Enables the echo function. The default is enabled.
Step 5	**show running-config bfd** **Example:** `switch(config-if)# show running-config bfd`	(Optional) Displays the BFD running configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Configuring BFD Support for Routing Protocols

Configuring BFD on BGP

You can configure BFD for the Border Gateway Protocol (BGP).

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the BGP feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*

3. **neighbor** (*ip-address* | *ipv6-address*) **remote-as** *as-number*
4. **bfd**
5. **update-source** *interface*
6. **show running-config bgp**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 3	**neighbor** (*ip-address* | *ipv6-address*) **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 209.165.201.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		Configures the IPv4 or IPv6 address and AS number for a remote BGP peer. The *ip-address* format is x.x.x.x. The *ipv6-address* format is A:B::C:D.

Step 4	**bfd** **Example:** `switch(config-router-neighbor)# bfd`	Configures BFD on the neighbor.
Step 5	**update-source** *interface* **Example:** `switch(config-router-neighbor)# update-source ethernet 2/1`	Allows BGP sessions to use the primary IP address from a particular interface as the local address when forming a BGP session with a neighbor and enables BGP to register as a client with BFD.
Step 6	**show running-config bgp** **Example:** `switch(config-router-neighbor)# show running-config bgp`	(Optional) Displays the BGP running configuration.
Step 7	**copy running-config startup-config** **Example:** `switch(config-router-neighbor)# copy running-config startup-config`	(Optional) Saves the configuration change.

Configuring BFD on EIGRP

You can configure BFD for the Enhanced Interior Gateway Routing Protocol (EIGRP).

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the EIGRP feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **bfd** [**ipv4** | **ipv6**]
4. **interface** *int-if*
5. **ip eigrp** *instance-tag* **bfd**
6. **show ip eigrp** [**vrf** *vrf-name*] [**interfaces** *if*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**router eigrp** *instance-tag*
	Example:
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
		If you configure an instance-tag that does not qualify as an AS number, you must use the **autonomous-system** to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.

Step 3	**bfd** [**ipv4**	**ipv6**] **Example:** `switch(config-router-neighbor)# bfd ipv4`	(Optional) Enables BFD for all EIGRP interfaces.

Step 4	**interface** *int-if*
	Example:
	`switch(config-router-neighbor)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.

Step 5	**ip eigrp** *instance-tag* **bfd** **Example:** `switch(config-if)# ip eigrp Test1 bfd`	(Optional) Enables or disables BFD on an EIGRP interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The default is disabled.

	Command or Action	Purpose
Step 6	**show ip eigrp** [**vrf** *vrf-name*] [**interfaces** *if*] **Example:** `switch(config-if)# show ip eigrp`	(Optional) Displays information about EIGRP. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters.
Step 7	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Configuring BFD on OSPF

You can configure BFD for the Open Shortest Path First.

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the OSPF feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **bfd** [**ipv4** | **ipv6**]
4. **interface** *int-if*
5. **ip ospf bfd**
6. **show ip ospf** [**vrf** *vrf-name*] [**interfaces** *if*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 200`
	`switch(config-router)#`
		Creates a new OSPF instance with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.

	Command or Action	Purpose
Step 3	**bfd** [**ipv4** | **ipv6**] **Example:** `switch(config-router)# bfd`	(Optional) Enables BFD for all OSPF interfaces.

Step 4	**interface** *int-if*
	Example:
	`switch(config-router)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.

Step 5	**ip ospf bfd** **Example:** `switch(config-if)# ip ospf bfd`	(Optional) Enables or disables BFD on an OSPF interface. The default is disabled.
Step 6	**show ip ospf** [**vrf** *vrf-name*] [**interfaces** *if*] **Example:** `switch(config-if)# show ip ospf`	(Optional) Displays information about OSPF. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters.
Step 7	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Example Configurations for BFD on OSPF

Example configuration where BFD is enabled under a non-default VRF (OSPFv3 neighbors in vrf3).

```
configure terminal
  router ospfv3 10
    vrf vrf3
    bfd
```

Configuring BFD on IS-IS

You can configure BFD for the Intermediate System-to-Intermediate System (IS-IS) protocol.

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the IS-IS feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

SUMMARY STEPS

1. **configure terminal**
2. **router isis** *instance-tag*
3. **bfd** [**ipv4** | **ipv6**]
4. **interface** *int-if*
5. **isis bfd**
6. **show isis** [**vrf** *vrf-name*] [**interface** *if*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**router isis** *instance-tag*
	Example:
	`switch(config)# router isis 100`
	`switch(config-router)# net 49.0001.1720.1600.1001.00`
	`switch(config-router)# address-family ipv6 unicast`
		Creates a new IS-IS instance with the configured *instance tag*.

Step 3	**bfd** [**ipv4**	**ipv6**] **Example:** `switch(config-router)# bfd`	(Optional) Enables BFD for all OSPF interfaces.

Step 4	**interface** *int-if*
	Example:
	`switch(config-router)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.

Step 5	**isis bfd** **Example:** `switch(config-if)# isis bfd`	(Optional) Enables or disables BFD on an IS-IS interface. The default is disabled.
Step 6	**show isis** [**vrf** *vrf-name*] [**interface** *if*] **Example:** `switch(config-if)# show isis`	(Optional) Displays information about IS-IS. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters.
Step 7	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

	Command or Action	Purpose

Example Configurations for BFD on IS-IS

Example configuration for IS-IS where BFD is enabled under IPv4 and an IPv6 address family.

```
configure terminal
  router isis isis-1
    bfd
    address-family ipv6 unicast
    bfd
```

Configuring BFD on HSRP

You can configure BFD for the Hot Standby Router Protocol (HSRP). The active and standby HSRP routers track each other through BFD. If BFD on the standby HSRP router detects that the active HSRP router is down, the standby HSRP router treats this event as an active time rexpiry and takes over as the active HSRP router.

The **show hsrp detail** command shows this event as BFD@Act-down or BFD@Sby-down.

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the HSRP feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

SUMMARY STEPS

1. **configure terminal**
2. **hsrp bfd all-interfaces**
3. **interface** *int-if*
4. **hsrp bfd**
5. **show running-config hsrp**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**hsrp bfd all-interfaces** **Example:** `switch# hsrp bfd all-interfaces`	(Optional) Enables or disables BFD on all HSRP interfaces. The default is disabled.

Step 3	**interface** *int-if*
	Example:
	`switch(config-router)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.

Step 4	**hsrp bfd** **Example:** `switch(config-if)# hsrp bfd`	(Optional) Enables or disables BFD on an HSRP interface. The default is disabled.
Step 5	**show running-config hsrp** **Example:** `switch(config-if)# show running-config hsrp`	(Optional) Displays the HSRP running configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Configuring BFD on VRRP

You can configure BFD for the Virtual Router Redundancy Protocol (VRRP). The active and standby VRRP routers track each other through BFD. If BFD on the standby VRRP router detects that the active VRRP router is down, the standby VRRP router treats this event as an active time rexpiry and takes over as the active VRRP router.

The **show vrrp detail** command shows this event as BFD@Act-down or BFD@Sby-down.

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the VRRP feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *int-if*
3. **vrrp** *group-no*
4. **vrrp bfd** *address*

5. **show running-config vrrp**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *int-if*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.

Step 3	**vrrp** *group-no* **Example:** `switch(config-if)# vrrp 2`	Specifies the VRRP group number.
Step 4	**vrrp bfd** *address* **Example:** `switch(config-if)# vrrp bfd`	Enables or disables BFD on a VRRP interface. The default is disabled.
Step 5	**show running-config vrrp** **Example:** `switch(config-if)# show running-config vrrp`	(Optional) Displays the VRRP running configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Configuring BFD on PIM

You can configure BFD for the Protocol Independent Multicast (PIM) protocol.

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Enable the PIM feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

SUMMARY STEPS

1. **configure terminal**

2. **ip pim bfd**
3. **interface** *int-if*
4. **ip pim bfd-instance** [**disable**]
5. **show running-config pim**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**ip pim bfd** **Example:** `switch(config)# ip pim bfd`	Enables BFD for PIM.

Step 3	**interface** *int-if*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.

Step 4	**ip pim bfd-instance** [**disable**] **Example:** `switch(config-if)# ip pim bfd-instance`	(Optional) Enables or disables BFD on a PIM interface. The default is disabled.
Step 5	**show running-config pim** **Example:** `switch(config)# show running-config pim`	(Optional) Displays the PIM running configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the configuration change.

Configuring BFD on Static Routes

You can configure BFD for static routes on an interface. You can optionally configure BFD on a static route within a virtual routing and forwarding (VRF) instance.

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **ip route** *route interface* {*nh-address* | *nh-prefix*}
4. **ip route static bfd** *interface* {*nh-address* | *nh-prefix*}
5. **show ip route static** [**vrf** *vrf-name*]
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vrf context** *vrf-name*
	Example:
	`switch(config)# vrf context Red`
	`switch(config-vrf)#`
		(Optional) Enters VRF configuration mode.

Step 3	**ip route** *route interface* {*nh-address* | *nh-prefix*} **Example:** `switch(config-vrf)# ip route 192.0.2.1 ethernet 2/1 192.0.2.4`	Creates a static route Use the **?** keyword to display the supported interfaces.
Step 4	**ip route static bfd** *interface* {*nh-address* | *nh-prefix*} **Example:** `switch(config-vrf)# ip route static bfd ethernet 2/1 192.0.2.4`	Enables BFD for all static routes on an interface. Use the**?** keyword to display the supported interfaces.
Step 5	**show ip route static** [**vrf** *vrf-name*] **Example:** `switch(config-vrf)# show ip route static vrf Red`	(Optional) Displays the static routes.
Step 6	**copy running-config startup-config** **Example:** `switch(config-vrf)# copy running-config startup-config`	(Optional) Saves the configuration change.

Disabling BFD on an Interface

You can selectively disable BFD on an interface for a routing protocol that has BFD enabled at the global or VRF level.

To disable BFD on an interface, use one of the following commands in interface configuration mode:

Command	Purpose
ip eigrp *instance-tag* **bfd disable** **Example:** `switch(config-if)# ip eigrp Test1 bfd disable`	Disables BFD on an EIGRP interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
ip ospf bfd disable **Example:** `switch(config-if)# ip ospf bfd disable`	Disables BFD on an OSPFv2 interface.
isis bfd disable **Example:** `switch(config-if)# isis bfd disable`	Disables BFD on an IS-IS interface.

Disabling BFD on an Interface

Example configuration where BFD is disabled per interface.

```
configure terminal
  interface port-channel 10
    no ip redirects
    ip address 22.1.10.1/30
    ipv6 address 22:1:10::1/120
    no ipv6 redirects
    ip router ospf 10 area 0.0.0.0
    ip ospf bfd disable        /*** disables IPv4 BFD session for OSPF
    ospfv3 bfd disable         /*** disables IPv6 BFD session for OSPFv3
```

Configuring BFD Interoperability

Configuring BFD Interoperability in Cisco NX-OS Devices in a Point-to-Point Link

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *int-if*
3. **ip ospf bfd**
4. **no ip redirects**
5. **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *int-if* **Example:** `switch(config-if)# interface ethernet 2/1`	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 3	**ip ospf bfd** **Example:** `switch(config-if)# ip ospf bfd`	Enables BFD on an OSPFv2 interface. The default is disabled. OSPF is used as an example. You can enable BFD of any of the supported protocols.
Step 4	**no ip redirects** **Example:** `switch(config-if)# no ip redirects`	Prevents the device from sending redirects.
Step 5	**bfd interval** *mintx* **min_rx** *msec* **multiplier** *value* **Example:** `switch(config-if)# bfd interval 50 min_rx 50 multiplier 3`	Configures the BFD session parameters for all BFD sessions on the port channel. This command overrides these values by configuring the BFD session parameters. The *mintx* and *msec* range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.
Step 6	**exit** **Example:** `switch(config-if)# exit`	Exits interface configuration mode and returns to EXEC mode.

Configuring BFD Interoperability in Cisco NX-OS Devices in a Switch Virtual Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *vlan vlan-id*
3. **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*
4. **no ip redirects**
5. **ip address** *ip-address/length*
6. **ip ospf bfd**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *vlan vlan-id*
	Example:
	`switch(config)# interface vlan 998`
	`switch(config-if)#`
		Creates a dynamic Switch Virtual Interface (SVI).

Step 3	**bfd interval** *mintx* **min_rx** *msec* **multiplier** *value* **Example:** `switch(config-if)# bfd interval 50 min_rx 50 multiplier 3`	Configures the BFD session parameters for all BFD sessions on the device. The *mintx* and *msec* range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.
Step 4	**no ip redirects** **Example:** `switch(config-if)# no ip redirects`	Prevents the device from sending redirects.
Step 5	**ip address** *ip-address/length* **Example:** `switch(config-if)# ip address 10.1.0.253/24`	Configures an IP address for this interface.
Step 6	**ip ospf bfd** **Example:** `switch(config-if)# ip ospf bfd`	Enables BFD on an OSPFv2 interface. The default is disabled.
Step 7	**exit** **Example:** `switch(config-if)# exit`	Exits interface configuration mode and returns to EXEC mode.

Configuring BFD Interoperability in Cisco NX-OS Devices in Logical Mode

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *type number.subinterface-id*
3. **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*
4. **no ip redirects**
5. **ip ospf bfd**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *type number.subinterface-id* **Example:** `switch(config-if)# interface port-channel 50.2`	Enters port channel configuration mode. Use the **?** keyword to display the supported number range.
Step 3	**bfd interval** *mintx* **min_rx** *msec* **multiplier** *value* **Example:** `switch(config-if)# bfd interval 50 min_rx 50 multiplier 3`	Configures the BFD session parameters for all BFD sessions on the port channel. The *mintx* and *msec* range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.
Step 4	**no ip redirects** **Example:** `switch(config-if)# no ip redirects`	Prevents the device from sending redirects.
Step 5	**ip ospf bfd** **Example:** `switch(config-if)# ip ospf bfd`	Enables BFD on an OSPFv2 interface. The default is disabled. OSPF is used as an example. You can enable BFD of any of the supported protocols.
Step 6	**exit** **Example:** `switch(config-if)# exit`	Exits interface configuration mode and returns to EXEC mode.

Verifying BFD Interoperability in a Cisco Nexus 9000 Series Device

The following example shows how to verify BFD interoperability in a Cisco Nexus 9000 Series device.

```
switch# show bfd neighbors details
OurAddr NeighAddr LD/RD RH/RS Holdown(mult) State Int
Vrf
10.1.1.1 10.1.1.2 1140850707/2147418093 Up 6393(4) Up Vlan2121
default
Session state is Up and using echo function with 50 ms interval
Local Diag: 0, Demand mode: 0, Poll bit: 0, Authentication: None
MinTxInt: 50000 us, MinRxInt: 2000000 us, Multiplier: 3
Received MinRxInt: 2000000 us, Received Multiplier: 4
Holdown (hits): 8000 ms (0), Hello (hits): 2000 ms (108)
Rx Count: 92, Rx Interval (ms) min/max/avg: 347/1996/1776 last: 1606 ms ago
Tx Count: 108, Tx Interval (ms) min/max/avg: 1515/1515/1515 last: 1233 ms ago
Registered protocols: ospf
Uptime: 0 days 0 hrs 2 mins 44 secs
Last packet: Version: 1 - Diagnostic: 0
State bit: Up - Demand bit: 0
```

```
Poll bit: 0 - Final bit: 0
Multiplier: 4 - Length: 24
My Discr.: 2147418093 - Your Discr.: 1140850707
Min tx interval: 2000000 - Min rx interval: 2000000
Min Echo interval: 1000 - Authentication bit: 0
Hosting LC: 10, Down reason: None, Reason not-hosted: None


switch# show bfd neighbors details
OurAddr NeighAddr LD/RD RH/RS Holdown(mult) State Int
Vrf
10.0.2.1 10.0.2.2 1140850695/131083 Up 270(3) Up Po14.121
default
Session state is Up and not using echo function
Local Diag: 0, Demand mode: 0, Poll bit: 0, Authentication: None
MinTxInt: 50000 us, MinRxInt: 50000 us, Multiplier: 3
Received MinRxInt: 100000 us, Received Multiplier: 3
Holdown (hits): 300 ms (0), Hello (hits): 100 ms (3136283)
Rx Count: 2669290, Rx Interval (ms) min/max/avg: 12/1999/93 last: 29 ms ago
Tx Count: 3136283, Tx Interval (ms) min/max/avg: 77/77/77 last: 76 ms ago
Registered protocols: ospf
Uptime: 2 days 21 hrs 41 mins 45 secs
Last packet: Version: 1 - Diagnostic: 0
State bit: Up - Demand bit: 0
Poll bit: 0 - Final bit: 0
Multiplier: 3 - Length: 24
My Discr.: 131083 - Your Discr.: 1140850695
Min tx interval: 100000 - Min rx interval: 100000
Min Echo interval: 0 - Authentication bit: 0
Hosting LC: 8, Down reason: None, Reason not-hosted: None
```

Verifying the BFD Configuration

To display BFD configuration information, perform one of the following:

Command	Purpose
show running-config bfd	Displays the running BFD configuration.
show startup-config bfd	Displays the BFD configuration that will be applied on the next system startup.

Monitoring BFD

Use the following commands to display BFD:

Command	Purpose
show bfd neighbors [**application** *name*] [**details**]	Displays information about BFD for a supported application, such as BGP or OSPFv2.
show bfd neighbors [**interface** *int-if*] [**details**]	Displays information about BGP sessions on an interface.

Command	Purpose
show bfd neighbors [**dest-ip** *ip-address*] [**src-ip** *ip-address*][**details**]	Displays information about the specified BGP session on an interface.
show bfd neighbors [**vrf** *vrf-name*] [**details**]	Displays information about BFD for a VRF.
show bfd [**ipv4** | **ipv6**] [**neighbors**]	Displays information about IPv4 neighbors or IPv6 neighbors.

Configuration Examples for BFD

This example shows how to configure BFD for OSPFv2 on Ethernet 2/1, using the default BFD session parameters:

```
feature bfd
feature ospf
router ospf Test1
interface ethernet 2/1
ip ospf bfd
no shutdown
```

This example shows how to configure BFD for all EIGRP interfaces, using the default BFD session parameters:

```
feature bfd
feature eigrp
bfd interval 100 min_rx 100 multiplier 4
router eigrp Test2
bfd
```

This example shows how to configure BFDv6:

```
feature bfd
feature ospfv3
router ospfv3 Test1
interface Ethernet2/7
  ipv6 router ospfv3 Test1 area 0.0.0.0
  ospfv3 bfd
  no shutdown
```

Show Example for BFD

This example shows results of the **show bfd ipv6 neighbors details** command.

```
#show bfd ipv6 neighbors details

OurAddr                          NeighAddr
 LD/RD                 RH/RS           Holdown(mult)     State       Int
 Vrf
cc:10::2                         cc:10::1
 1090519335/1090519260 Up              5692(3)           Up          Po1
 default
```

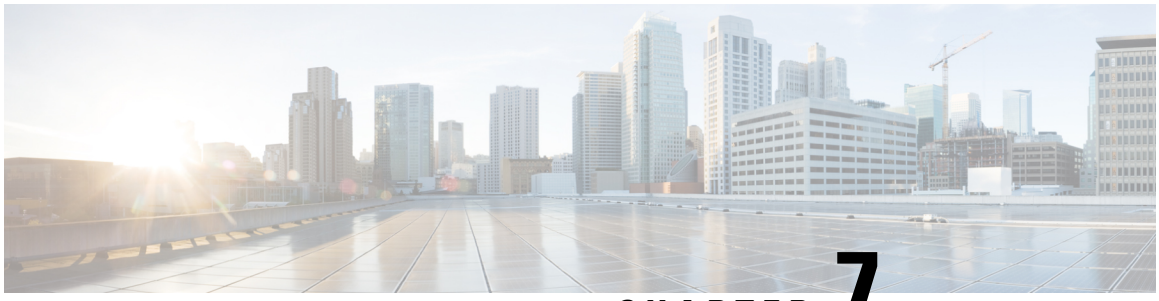
```
Session state is Up and using echo function with 250 ms interval
Local Diag: 0, Demand mode: 0, Poll bit: 0, Authentication: None
MinTxInt: 250000 us, MinRxInt: 2000000 us, Multiplier: 3
Received MinRxInt: 2000000 us, Received Multiplier: 3
Holdown (hits): 6000 ms (4), Hello (hits): 2000 ms (205229)
Rx Count: 227965, Rx Interval (ms) min/max/avg: 124/1520/1510 last: 307 ms ago
Tx Count: 205229, Tx Interval (ms) min/max/avg: 1677/1677/1677 last: 587 ms ago
Registered protocols:  bgp
Uptime: 3 days 23 hrs 31 mins 13 secs
Last packet: Version: 1               - Diagnostic: 0
             State bit: Up            - Demand bit: 0
             Poll bit: 0              - Final bit: 0
             Multiplier: 3            - Length: 24
             My Discr.: 1090519260    - Your Discr.: 1090519335
             Min tx interval: 250000  - Min rx interval: 2000000
             Min Echo interval: 250000 - Authentication bit: 0
Hosting LC: 1, Down reason: None, Reason not-hosted: None
```

Related Documents

Related Topic	Document Title
BFD commands	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*

RFCs

RFC	Title
RFC 5880	*Bidirectional Forwarding Detection (BFD)*
RFC 5881	*BFD for IPv4 and IPv6 (Single Hop)*
RFC 7130	*Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces*



CHAPTER 7

Configuring Port Channels

This chapter describes how to configure port channels and to apply and configure the Link Aggregation Control Protocol (LACP) for more efficient use of port channels in the Cisco NX-OS devices.

On a single switch, the port-channel compatibility parameters must be the same among all the port-channel members on the physical switch.

About Port Channels

A port channel is an aggregation of multiple physical interfaces that creates a logical interface. You can bundle up to 32 individual active links into a port channel to provide increased bandwidth and redundancy. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational as long as at least one physical interface within the port channel is operational.

You can create a Layer 2 port channel by bundling compatible Layer 2 interfaces, or you can create Layer 3 port channels by bundling compatible Layer 3 interfaces. You cannot combine Layer 2 and Layer 3 interfaces in the same port channel.

You can also change the port channel from Layer 3 to Layer 2. See the Configuring Layer 2 Interfaces chapter for information about creating Layer 2 interfaces.

A Layer 2 port channel interface and it's member ports can have different STP parameters. Changing the STP parameters of the port channel does not impact the STP parameters of the member ports because a port channel interface takes precedence if the member ports are bundled.

Note After a Layer 2 port becomes part of a port channel, all switchport configurations must be done on the port channel; you can no longer apply switchport configurations to individual port-channel members. You cannot apply Layer 3 configurations to an individual port-channel member either; you must apply the configuration to the entire port channel.

You can use static port channels, with no associated aggregation protocol, for a simplified configuration.

For more flexibility, you can use the Link Aggregation Control Protocol (LACP), which is defined in IEEE 802.3ad. When you use LACP, the link passes protocol packets. You cannot configure LACP on shared interfaces.

See the LACP Overview section for information about LACP.

Port Channels

A port channel bundles physical links into a channel group to create a single logical link that provides the aggregate bandwidth of up to 32 physical links. If a member port within a port channel fails, the traffic previously carried over the failed link switches to the remaining member ports within the port channel.

However, you can enable the LACP to use port channels more flexibly. Configuring port channels with LACP and static port channels require a slightly different procedure (see the “Configuring Port Channels” section).

Note The device does not support Port Aggregation Protocol (PAgP) for port channels.

Each port can be in only one port channel. All the ports in a port channel must be compatible; they must use the same speed and duplex mode (see the “Compatibility Requirements” section). When you run static port channels with no aggregation protocol, the physical links are all in the on channel mode; you cannot change this mode without enabling LACP (see the “Port-Channel Modes” section).

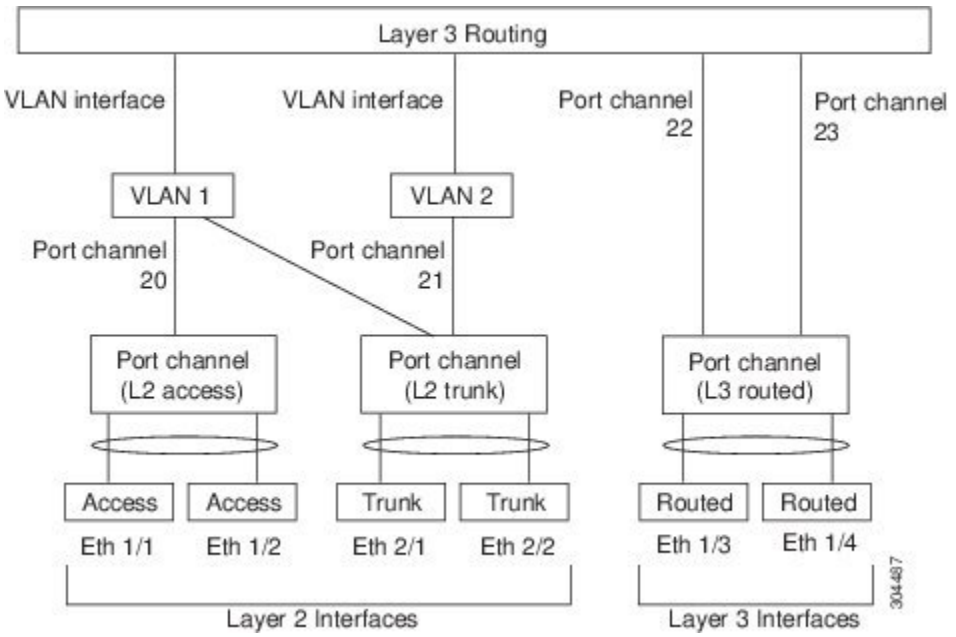
You can create port channels directly by creating the port-channel interface, or you can create a channel group that acts to aggregate individual ports into a bundle. When you associate an interface with a channel group, the software creates a matching port channel automatically if the port channel does not already exist. In this instance, the port channel assumes the Layer 2 or Layer 3 configuration of the first interface. You can also create the port channel first. In this instance, the Cisco NX-OS software creates an empty channel group with the same channel number as the port channel and takes the default Layer 2 or Layer 3 configuration, as well as the compatibility configuration (see the “Compatibility Requirements” section).

Note The port channel is operationally up when at least one of the member ports is up and that port’s status is channeling. The port channel is operationally down when all member ports are operationally down.

Port-Channel Interfaces

The following shows port-channel interfaces.

Figure 8: Port-Channel Interfaces



You can classify port-channel interfaces as Layer 2 or Layer 3 interfaces. In addition, you can configure Layer 2 port channels in either access or trunk mode. Layer 3 port-channel interfaces have routed ports as channel members.

You can configure a Layer 3 port channel with a static MAC address. If you do not configure this value, the Layer 3 port channel uses the router MAC of the first channel member to come up. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for information about configuring static MAC addresses on Layer 3 port channels.

See the "Configuring Layer 2 Interfaces" chapter for information about configuring Layer 2 ports in access or trunk mode and the "Configuring Layer 3 Interfaces" chapter for information about configuring Layer 3 interfaces and subinterfaces.

Basic Settings

You can configure the following basic settings for the port-channel interface:

- Bandwidth—Use this setting for informational purposes only; this setting is to be used by higher-level protocols.
- Delay—Use this setting for informational purposes only; this setting is to be used by higher-level protocols.
- Description
- Duplex
- IP addresses
- Maximum Transmission Unit (MTU)
- Shutdown

- Speed

Compatibility Requirements

When you add an interface to a channel group, the software checks certain interface attributes to ensure that the interface is compatible with the channel group. For example, you cannot add a Layer 3 interface to a Layer 2 channel group. The Cisco NX-OS software also checks a number of operational attributes for an interface before allowing that interface to participate in the port-channel aggregation.

The compatibility check includes the following operational attributes:

- Network layer
- (Link) speed capability
- Speed configuration
- Duplex capability
- Duplex configuration
- Port mode
- Access VLAN
- Trunk native VLAN
- Tagged or untagged
- Allowed VLAN list
- MTU size
- SPAN—Cannot be a SPAN source or a destination port
- Storm control
- Flow-control capability
- Flow-control configuration
- Media type, either copper or fiber

Use the **show port-channel compatibility-parameters** command to see the full list of compatibility checks that the Cisco NX-OS uses.

You can only add interfaces configured with the channel mode set to on to static port channels, and you can only add interfaces configured with the channel mode as active or passive to port channels that are running LACP. You can configure these attributes on an individual member port. If you configure a member port with an incompatible attribute, the software suspends that port in the port channel.

Alternatively, you can force ports with incompatible parameters to join the port channel if the following parameters are the same:

- (Link) speed capability
- Speed configuration

- Duplex capability
- Duplex configuration
- Flow-control capability
- Flow-control configuration

When the interface joins a port channel, some of its individual parameters are removed and replaced with the values on the port channel as follows:

- Bandwidth
- Delay
- Extended Authentication Protocol over UDP
- VRF
- IP address
- MAC address
- Spanning Tree Protocol
- NAC
- Service policy
- Access control lists (ACLs)

Many interface parameters remain unaffected when the interface joins or leaves a port channel as follows:

- Beacon
- Description
- CDP
- LACP port priority
- Debounce
- UDLD
- MDIX
- Rate mode
- Shutdown
- SNMP trap

Note When you delete the port channel, the software sets all member interfaces as if they were removed from the port channel.

See the “LACP Marker Responders” section for information about port-channel modes.

Load Balancing Using Port Channels

The Cisco NX-OS software load balances traffic across all operational interfaces in a port channel by hashing the addresses in the frame to a numerical value that selects one of the links in the channel. Port channels provide load balancing by default. Port-channel load balancing uses MAC addresses, IP addresses, or Layer 4 port numbers to select the link. Port-channel load balancing uses either source or destination addresses or ports, or both source and destination addresses or ports.

You can configure the load- balancing mode to apply to all port channels that are configured on the entire device. You can configure one load-balancing mode for the entire device. You cannot configure the load-balancing method per port channel.

You can configure the type of load-balancing algorithm used. You can choose the load-balancing algorithm that determines which member port to select for egress traffic by looking at the fields in the frame.

The default load-balancing mode for Layer 3 interfaces is the source and destination IP L4 ports, and the default load-balancing mode for non-IP traffic is the source and destination MAC address. Use the **port-channel load-balance** command to set the load-balancing method among the interfaces in the channel-group bundle. The default method for Layer 2 packets is src-dst-mac. The default method for Layer 3 packets is src-dst ip-l4port.

You can configure the device to use one of the following methods to load balance across the port channel:

- Destination MAC address
- Source MAC address
- Source and destination MAC address
- Destination IP address
- Source IP address
- Source and destination IP address
- Source TCP/UDP port number
- Destination TCP/UDP port number
- Source and destination TCP/UDP port number
- GRE inner IP headers with source, destination and source-destination

Non-IP and Layer 3 port channels both follow the configured load-balancing method, using the source, destination, or source and destination parameters. For example, when you configure load balancing to use the source IP address, all non-IP traffic uses the source MAC address to load balance the traffic while the Layer 3 traffic load balances the traffic using the source IP address. Similarly, when you configure the destination MAC address as the load-balancing method, all Layer 3 traffic uses the destination IP address while the non-IP traffic load balances using the destination MAC address.

The unicast and multicast traffic is load-balanced across port-channel links based on configured load-balancing algorithm displayed in **show port-channel load-balancing** command output.

The multicast traffic uses the following methods for load balancing with port channels:

- Multicast traffic with Layer 4 information—Source IP address, source port, destination IP address, destination port

- Multicast traffic without Layer 4 information—Source IP address, destination IP address
- Non-IP multicast traffic—Source MAC address, destination MAC address

Note Devices that run Cisco IOS can optimize the behavior of the member ports ASICs if a failure of a single member occurred by running the port-channel hash-distribution command. The Cisco Nexus 9000 Series device performs this optimization by default and does not require or support this command. Cisco NX-OS does support the customization of the load-balancing criteria on port channels through the port-channel load-balance command for the entire device.

Symmetric Hashing

To be able to effectively monitor traffic on a port channel, it is essential that each interface connected to a port channel receives both forward and reverse traffic flows. Normally, there is no guarantee that the forward and reverse traffic flows will use the same physical interface. However, when you enable symmetric hashing on the port channel, bidirectional traffic is forced to use the same physical interface and each physical interface in the port channel is effectively mapped to a set of flows.

When symmetric hashing is enabled, the parameters used for hashing, such as the source and destination IP address, are normalized before they are entered into the hashing algorithm. This process ensures that when the parameters are reversed (the source on the forward traffic becomes the destination on the reverse traffic), the hash output is the same. Therefore, the same interface is chosen.

Only the following load-balancing algorithms support symmetric hashing:

- src-dst ip
- src-dst ip-l4port

Resilient Hashing

With the exponential increase in the number of physical links used in data centers, there is also the potential for an increase in the number of failed physical links. In static hashing systems that are used for load balancing flows across members of port channels or Equal Cost Multipath (ECMP) groups, each flow is hashed to a link. If a link fails, all flows are rehashed across the remaining working links. This rehashing of flows to links results in some packets being delivered out of order even for those flows that were not hashed to the failed link.

This rehashing also occurs when a link is added to the port channel or Equal Cost Multipath (ECMP) group. All flows are rehashed across the new number of links, which results in some packets being delivered out of order.

Resilient hashing maps flows to physical ports and it is supported for both ECMP groups and port channel interfaces.

If a physical link fails, the flows originally assigned to the failed link are redistributed uniformly among the remaining working links. The existing flows through the working links are not rehashed and hence are not impacted.

Resilient hashing supports IPv4 and IPv6 unicast traffic, but it does not support IPv4 multicast traffic.

Resilient hashing is supported on all the Cisco Nexus 9000 Series platforms. . Beginning Cisco NX-OS Release 9.3(3), resilient hashing is supported on Cisco Nexus 92160YC-X, 92304QC, 9272Q, 9232C, 9236C, 92300YC switches.

LACP

LACP allows you to configure up to 16 interfaces into a port channel.

LACP Overview

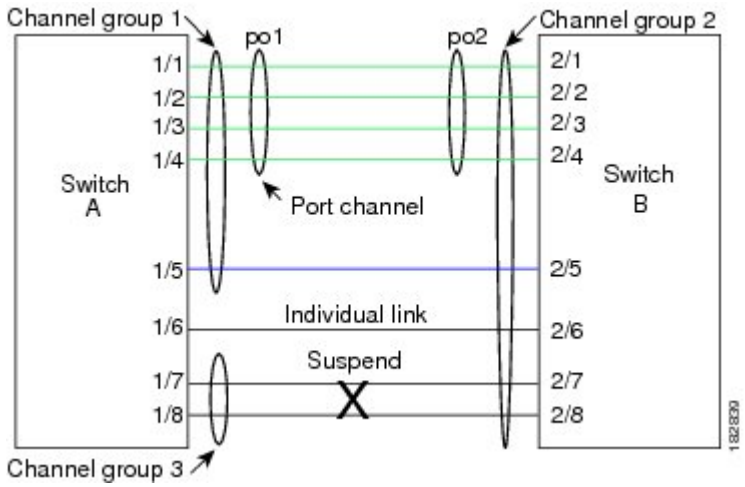
The Link Aggregation Control Protocol (LACP) for Ethernet is defined in IEEE 802.1AX and IEEE 802.3ad. This protocol controls how physical ports are bundled together to form one logical channel.

Note You must enable LACP before you can use LACP. By default, LACP is disabled. See the “Enabling LACP” section for information about enabling LACP.

The system automatically takes a checkpoint before disabling the feature, and you can roll back to this checkpoint. See the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide for information about rollbacks and checkpoints.

The following figure shows how individual links can be combined into LACP port channels and channel groups as well as function as individual links.

Figure 9: Individual Links Combined into a Port Channel



With LACP, you can bundle up to 32 interfaces in a channel group.

Note When you delete the port channel, the software automatically deletes the associated channel group. All member interfaces revert to their original configuration.

You cannot disable LACP while any LACP configurations are present.

Port-Channel Modes

Individual interfaces in port channels are configured with channel modes. When you run static port channels with no aggregation protocol, the channel mode is always set to **on**. After you enable LACP globally on the device, you enable LACP for each channel by setting the channel mode for each interface to either **active** or **passive**. You can configure channel mode for individual links in the LACP channel group when you are adding the links to the channel group

Note You must enable LACP globally before you can configure an interface in either the **active** or **passive** channel mode.

The following table describes the channel modes.

Table 9: Channel Modes for Individual Links in a Port Channel

Channel Mode	Description
passive	The LACP is enabled on this port channel and the ports are in a passive negotiating state. Ports responds to LACP packets that it receives but does not initiate LACP negotiation.
active	The LACP is enabled on this port channel and the ports are in an active negotiating state. Ports initiate negotiations with other ports by sending LACP packets.
on	The LACP is disabled on this port channel and the ports are in a non-negotiating state. The **on** state of the port channel represents the static mode. The port will not verify or negotiate port channel memberships. If you attempt to change the channel mode to active or passive before enabling LACP, the device displays an error message. When an LACP attempts to negotiate with an interface in the **on** state, it does not receive any LACP packets and becomes an individual link with that interface, it does not join the LACP channel group. The **on** state is the default port-channel mode

Both the passive and active modes allow LACP to negotiate between ports to determine if they can form a port channel based on criteria such as the port speed and the trunking state.The passive mode is useful when you do not know whether the remote system, or partner, supports LACP.

Two devices can form an LACP port channel when their ports are in different LACP modes if the modes are compatible as in the following example:

Table 10: Channel Modes Compatibility

Device 1 > Port-1	Device 2 > Port-2	Result
Active	Active	Can form a port channel.
Active	Passive	Can form a port channel.
Passive	Passive	Cannot form a port channel because no ports can initiate negotiation.
On	Active	Cannot form a port channel because LACP is enabled only on one side.
On	Passive	Cannot form a port channel because LACP is not enabled.

LACP ID Parameters

This section describes the LACP parameters.

LACP System Priority

Each system that runs LACP has an LACP system priority value. You can accept the default value of 32768 for this parameter, or you can configure a value between 1 and 65535. LACP uses the system priority with the MAC address to form the system ID and also uses the system priority during negotiation with other devices. A higher system priority value means a lower priority.

Note The LACP system ID is the combination of the LACP system priority value and the MAC address.

LACP Port Priority

Each port that is configured to use LACP has an LACP port priority. You can accept the default value of 32768 for the LACP port priority, or you can configure a value between 1 and 65535. LACP uses the port priority with the port number to form the port identifier.

LACP uses the port priority to decide which ports should be put in standby mode when there is a limitation that prevents all compatible ports from aggregating and which ports should be put into active mode. A higher port priority value means a lower priority for LACP. You can configure the port priority so that specified ports have a lower priority for LACP and are most likely to be chosen as active links, rather than hot-standby links.

LACP Administrative Key

LACP automatically configures an administrative key value equal to the channel-group number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port's ability to aggregate with other ports is determined by these factors:

- Port physical characteristics, such as the data rate and the duplex capability
- Configuration restrictions that you establish

LACP Marker Responders

You can dynamically redistribute the data traffic by using port channels. This redistribution might result from a removed or added link or a change in the load-balancing scheme. Traffic redistribution that occurs in the middle of a traffic flow can cause misordered frames.

LACP uses the Marker Protocol to ensure that frames are not duplicated or reordered due to this redistribution. The Marker Protocol detects when all the frames of a given traffic flow are successfully received at the remote end. LACP sends Marker PDUs on each of the port-channel links. The remote system responds to the Marker PDU once it receives all the frames received on this link prior to the Marker PDU. The remote system then sends a Marker Responder. Once the Marker Responders are received by the local system on all member links of the port channel, the local system can redistribute the frames in the traffic flow with no chance of misordering. The software supports only Marker Responders.

LACP-Enabled and Static Port Channels Differences

The following table summarizes the major differences between port channels with LACP enabled and static port channels.

Table 11: Port Channels with LACP Enabled and Static Port Channels

Configurations	Port Channels with LACP Enabled	Static Port Channels
Protocol applied	Enable globally	Not applicable
Channel mode of links	Can be either: • Active • Passive	Can only be On
Maximum number of links in channel	32	32

LACP Compatibility Enhancements

When a Cisco Nexus 9000 Series device is connected to a non-Nexus peer, its graceful failover defaults may delay the time that is taken to bring down a disabled port or cause traffic from the peer to be lost. To address these conditions, the **lacp graceful-convergence** command was added.

By default, LACP sets a port to suspended state if it does not receive an LACP PDU from the peer. **lacp suspend-individual** is a default configuration on Cisco Nexus 9000 series switches. This command puts the port in suspended state if it does not receive any LACP PDUs. In some cases, although this feature helps in preventing loops created due to misconfigurations, it can cause servers fail to boot up because they require LACP to logically bring up the port. You can put a port into an individual state by using the **no lacp suspend-individual**. Port in individual sate takes attributes of the individual port based on the port configuration.

LACP port-channels exchange LACP PDUs for quick bundling of links when connecting a server and a switch. However, the links go into suspended state when the PDUs are not received.

The **delayed LACP** feature enables one port-channel member, the delayed-LACP port, to come up first as a member of a regular port-channel before LACP PDUs are received. After it is connected in LACP mode,

other members, the auxiliary LACP ports, are brought up. This avoids having the links becoming suspended when PDUs are not received.

Which port in the port-channel comes up first depends on the port-priority value of the ports. A member link in a port channel with lowest priority value, will come come up first as a LACP delayed port. Regardless of the operational status of the links, the configured priority of a LACP port is used to select the delayed-lacp port

This feature supports Layer 2 port channels, trunk mode spanning tree, and vPC and has the following limitations:

- Using **no lacp suspend-individual** and **lacp mode delay** on a same port channel is not recommended because it can put non-lacp delayed ports in individual state. As a best practice, you must avoid combining these two configurations.
- Not supported on Layer 3 port channels.
- Not supported on Cisco Nexus 9500 Switches and FEX HIF and FEX fabric ports.

LACP Port-Channel Minimum Links and MaxBundle

A port channel aggregates similar ports to provide increased bandwidth in a single manageable interface.

The introduction of the minimum links and maxbundle feature further refines LACP port-channel operation and provides increased bandwidth in one manageable interface.

The LACP port-channel minimum links feature does the following:

- Configures the minimum number of ports that must be linked up and bundled in the LACP port channel.
- Prevents the low-bandwidth LACP port channel from becoming active.
- Causes the LACP port channel to become inactive if there are few active members ports to supply the required minimum bandwidth.

The LACP MaxBundle defines the maximum number of bundled ports allowed in a LACP port channel.

The LACP MaxBundle feature does the following:

- Defines an upper limit on the number of bundled ports in an LACP port channel.
- Allows hot-standby ports with fewer bundled ports. (For example, in an LACP port channel with five ports, you can designate two of those ports as hot-standby ports.)

Note The minimum links and maxbundle feature works only with LACP port channels. However, the device allows you to configure this feature in non-LACP port channels, but the feature is not operational.

LACP Fast Timers

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the lacp rate command to set the rate at which LACP control packets are sent to an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only

on LACP-enabled interfaces. To configure the LACP fast time rate, see the "Configuring the LACP Fast Timer Rate" section.

ISSU and ungraceful switchovers are not supported with LACP fast timers.

Virtualization Support

You must configure the member ports and other port channel-related configuration from the virtual device context (VDC) that contains the port channel and member ports. You can use the numbers from 1 to 4096 in each VDC to number the port channels.

All ports in one port channel must be in the same VDC. When you are using LACP, all possible 8 active ports and all possible 8 standby ports must be in the same VDC.

Note You must configure load balancing using port channels in the default VDC. See the "Load Balancing Using Port Channels" section for more information about load balancing.

High Availability

Port channels provide high availability by load balancing traffic across multiple ports. If a physical port fails, the port channel is still operational if there is an active member in the port channel. You can bundle ports from different modules and create a port channel that remains operational even if a module fails because the settings are common across the module.

Port channels support stateful and stateless restarts. A stateful restart occurs on a supervisor switchover. After the switchover, the Cisco NX-OS software applies the runtime configuration after the switchover.

The port channel goes down if the operational ports fall below the configured minimum links number.

Note See the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide* for complete information about high-availability features.

Prerequisites for Port Channeling

Port channeling has the following prerequisites:

- You must be logged onto the device.
- All ports for a single port channel must be either Layer 2 or Layer 3 ports.
- All ports for a single port channel must meet the compatibility requirements. See the "Compatibility Requirements" section for more information about the compatibility requirements.
- You must configure load balancing from the default VDC.

Guidelines and Limitations

Port channeling has the following configuration guidelines and limitations:

- For scaled port-channel deployments on Cisco Nexus 9516 switch with Gen 1 line cards, you need to use the **port-channel scale-fanout** command followed by **copy run start** and **reload** commands.
- **show** commands with the **internal** keyword are not supported.
- The LACP port-channel minimum links and maxbundle feature is not supported for host interface port channels.
- You must enable LACP before you can use that feature.
- You can configure multiple port channels on a device.
- Do not put shared and dedicated ports into the same port channel. (See the "Configuring Basic Interface Parameters" chapter for information about shared and dedicated ports.)
- For Layer 2 port channels, ports with different STP port path costs can form a port channel if they are compatibly configured with each other. See the "Compatibility Requirements" section for more information about the compatibility requirements.
- In STP, the port-channel cost is based on the aggregated bandwidth of the port members.
- After you configure a port channel, the configuration that you apply to the port channel interface affects the port channel member ports. The configuration that you apply to the member ports affects only the member port where you apply the configuration.
- LACP does not support half-duplex mode. Half-duplex ports in LACP port channels are put in the suspended state.
- You must remove the port-security information from a port before you can add that port to a port channel. Similarly, you cannot apply the port-security configuration to a port that is a member of a channel group.
- Do not configure ports that belong to a port channel group as private VLAN ports. While a port is part of the private VLAN configuration, the port channel configuration becomes inactive.
- Channel member ports cannot be a source or destination SPAN port.
- The port channel might be affected by the limitations of the Application Leaf Engine (ALE) uplink ports on Cisco Nexus 9300 and 9500 Series devices:Limitations for ALE Uplink Ports

Default Settings

The following table lists the default settings for port-channel parameters.

Table 12: Default Port-Channel Parameters

Parameters	Default
Port channel	Admin up
Load balancing method for Layer 3 interfaces	Source and destination IP address

Parameters	Default
Load balancing method for Layer 2 interfaces	Source and destination MAC address
Load balancing per module	Disabled
LACP	Disabled
Channel mode	on
LACP system priority	32768
LACP port priority	32768
Minimum links for LACP	1
Maxbundle	32
Minimum links for FEX fabric port channel	1

Configuring Port Channels

Note See the "Configuring Basic Interface Parameters" chapter for information about configuring the maximum transmission unit (MTU) for the port-channel interface. See the "Configuring Layer 3 Interfaces" chapter for information about configuring IPv4 and IPv6 addresses on the port-channel interface.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Creating a Port Channel

You can create a port channel before you create a channel group. The software automatically creates the associated channel group.

Note When the port channel is created before the channel group, the port channel should be configured with all of the interface attributes that the member interfaces are configured with. Use the **switchport mode trunk** {*allowed vlan vlan-id* | *native vlan-id*} command to configure the members.

This is required only when the channel group members are Layer 2 ports (switchport) and trunks (switchport mode trunk).

Note Use the **no interface port-channel** command to remove the port channel and delete the associated channel group.

Command	Purpose
no interface port-channel *channel-number* **Example:** `switch(config)# no interface port-channel 1`	Removes the port channel and deletes the associated channel group.

Before you begin

Enable LACP if you want LACP-based port channels.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel-number*
3. **show port-channel summary**
4. **no shutdown**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)`
		Specifies the port-channel interface to configure, and enters the interface configuration mode. The range is from 1 to 4096. The Cisco NX-OS software automatically creates the channel group if it does not already exist.

Step 3	**show port-channel summary** **Example:** `switch(config-router)# show port-channel summary`	(Optional) Displays information about the port channel.

Step 4	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

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

Example

This example shows how to create a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
```

See the "Compatibility Requirements" section for details on how the interface configuration changes when you delete the port channel.

Adding a Layer 2 Port to a Port Channel

You can add a Layer 2 port to a new channel group or to a channel group that already contains Layer 2 ports. The software creates the port channel associated with this channel group if the port channel does not already exist.

Note Use the **no channel-group** command to remove the port from the channel group.

Command	Purpose
no channel-group **Example:** `switch(config)# no channel-group`	Removes the port from the channel group.

Before you begin

Enable LACP if you want LACP-based port channels.

All Layer 2 member ports must run in full-duplex mode and at the same speed

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **switchport**
4. **switchport mode trunk**
5. **switchport trunk** {**allowed vlan** *vlan-id* | **native** *vlan-id*}
6. **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]
7. **show interface** *type slot/port*
8. **no shutdown**

9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.

Step 3	**switchport** **Example:** `switch(config)# switchport`	Configures the interface as a Layer 2 access port.
Step 4	**switchport mode trunk** **Example:** `switch(config)# switchport mode trunk`	(Optional) Configures the interface as a Layer 2 trunk port.

Step 5	**switchport trunk** {**allowed vlan** *vlan-id* | **native** *vlan-id*}
	Example:
	`switch(config)# switchport trunk native 3`
	`switch(config-if)#`
		(Optional) Configures necessary parameters for a Layer 2 trunk port.

Step 6	**channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}] **Example:** • `switch(config-if)# channel-group 5` • `switch(config-if)# channel-group 5 force`	Configures the port in a channel group and sets the mode. The channel-number range is from 1 to 4096. This command creates the port channel associated with this channel group if the port channel does not already exist. All static port-channel interfaces are set to mode **on**. You must set all LACP-enabled port-channel interfaces to **active** or **passive**. The default mode is **on**. (Optional) Forces an interface with some incompatible configurations to join the channel. The forced interface must have the same speed, duplex, and flow control settings as the channel group. **Note** The **force** option fails if the port has a QoS policy mismatch with the other members of the port channel.
Step 7	**show interface** *type slot/port* **Example:** `switch# show interface port channel 5`	(Optional) Displays interface information.

	Command or Action	Purpose

Step 8	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 9	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to add a Layer 2 Ethernet interface 1/4 to channel group 5:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# switchport
switch(config-if)# channel-group 5
```

Adding a Layer 3 Port to a Port Channel

You can add a Layer 3 port to a new channel group or to a channel group that is already configured with Layer 3 ports. The software creates the port channel associated with this channel group if the port channel does not already exist.

If the Layer 3 port that you are adding has a configured IP address, the system removes that IP address before adding the port to the port channel. After you create a Layer 3 port channel, you can assign an IP address to the port-channel interface.

Note Use the **no channel-group** command to remove the port from the channel group. The port reverts to its original configuration. You must reconfigure the IP addresses for this port.

Command	Purpose
no channel-group **Example:** `switch(config)# no channel-group`	Removes the port from the channel group.

Before you begin

Enable LACP if you want LACP-based port channels.

Remove any IP addresses configured on the Layer 3 interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **no switchport**
4. **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]
5. **show interface** *type slot/port*
6. **no shutdown**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.

Step 3	**no switchport** **Example:** `switch(config-if)# no switchport`	Configures the interface as a Layer 3 port.
Step 4	**channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}] **Example:** • `switch(config-if)# channel-group 5` • `switch(config-if)# channel-group 5 force`	Configures the port in a channel group and sets the mode. The channel-number range is from 1 to 4096. The Cisco NX-OS software creates the port channel associated with this channel group if the port channel does not already exist. (Optional) Forces an interface with some incompatible configurations to join the channel. The forced interface must have the same speed, duplex, and flow control settings as the channel group.
Step 5	**show interface** *type slot/port* **Example:** `switch# show interface ethernet 1/4`	(Optional) Displays interface information.

Step 6	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:**	(Optional) Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

Example

This example shows how to add a Layer 3 Ethernet interface 1/5 to channel group 6 in on mode:

```
switch# configure terminal
switch (config)# interface ethernet 1/5
switch(config-if)# switchport
switch(config-if)# channel-group 6
```

This example shows how to create a Layer 3 port-channel interface and assign the IP address:

```
switch# configure terminal
switch (config)# interface port-channel 4
switch(config-if)# ip address 192.0.2.1/8
```

Configuring the Bandwidth and Delay for Informational Purposes

The bandwidth of the port channel is determined by the number of total active links in the channel.

You configure the bandwidth and delay on port-channel interfaces for informational purposes.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel-number*
3. **bandwidth** *value*
4. **delay** *value*
5. **exit**
6. **show interface port-channel** *channel-number*
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 2`
	`switch(config-if)#`
		Specifies the port-channel interface that you want to configure, and enters the interface mode.

	Command or Action	Purpose

Step 3	**bandwidth** *value*
	Example:
	`switch(config-if)# bandwidth 60000000`
	`switch(config-if)#`
		Specifies the bandwidth, which is used for informational purposes. The range is from 1 to 3,200,000,000 kbs. The default value depends on the total active interfaces in the channel group.

Step 4	**delay** *value*
	Example:
	`switch(config-if)# delay 10000`
	`switch(config-if)#`
		Specifies the throughput delay, which is used for informational purposes. The range is from 1 to 16,777,215 tens of microseconds. The default value is 10 microseconds.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode and returns to the configuration mode.

Step 6	**show interface port-channel** *channel-number* **Example:** `switch# show interface port-channel 2`	(Optional) Displays interface information for the specified port channel.
Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure the informational parameters of the bandwidth and delay for port channel 5:

```
switch# configure terminal
switch (config)# interface port-channel 5
switch(config-if)# bandwidth 60000000
switch(config-if)# delay 10000
switch(config-if)#
```

Shutting Down and Restarting the Port-Channel Interface

You can shut down and restart the port-channel interface. When you shut down a port-channel interface, no traffic passes and the interface is administratively down.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel-number*
3. **shutdown**
4. **exit**
5. **show interface port-channel** *channel-number*

6. **no shutdown**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 2`
	`switch(config-if)#`
		Specifies the port-channel interface that you want to configure, and enters the interface mode.

Step 3	**shutdown**
	Example:
	`switch(config-if)# shutdown`
	`switch(config-if)#`
		Shuts down the interface. No traffic passes and the interface displays as administratively down. The default is no shutdown.
		Note Use the **no shutdown** command to open the interface.
		The interface displays as administratively up. If there are no operational problems, traffic passes. The default is no shutdown.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode and returns to the configuration mode.

Step 5	**show interface port-channel** *channel-number* **Example:** `switch(config-router)# show interface port-channel 2`	(Optional) Displays interface information for the specified port channel.

Step 6	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e3/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to bring up the interface for port channel 2:

```
switch# configure terminal
switch (config)# interface port-channel 2
switch(config-if)# no shutdown
```

Configuring a Port-Channel Description

You can configure a description for a port channel.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel-number*
3. **description**
4. **exit**
5. **show interface port-channel** *channel-number*
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 2`
	`switch(config-if)#`
		Specifies the port-channel interface that you want to configure, and enters the interface mode.

Step 3	**description**
	Example:
	`switch(config-if)# description engineering`
	`switch(config-if)#`
		Allows you to add a description to the port-channel interface. You can use up to 80 characters in the description. By default, the description does not display; you must configure this parameter before the description displays in the output.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode and returns to the configuration mode.

Step 5	**show interface port-channel** *channel-number* **Example:** `switch# show interface port-channel 2`	(Optional) Displays interface information for the specified port channel.

	Command or Action	Purpose
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to add a description to port channel 2:

```
switch# configure terminal
switch (config)# interface port-channel 2
switch(config-if)# description engineering
```

Configuring the Speed and Duplex Settings for a Port-Channel Interface

You can configure the speed and duplex settings for a port-channel interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel-number*
3. **speed** {**10** | **100** | **1000** | **auto**}
4. **duplex** {**auto** | **full** | **half**}
5. **exit**
6. **show interface port-channel** *channel-number*
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 2`
	`switch(config-if)#`
		Specifies the port-channel interface that you want to configure, and enters the interface mode.

Step 3	**speed** {**10**	**100**	**1000**	**auto**}
				Example:
				`switch(config-if)# speed auto`
				`switch(config-if)#`
					Sets the speed for the port-channel interface. The default is auto for autonegotiation.

	Command or Action	Purpose

Step 4	**duplex** {**auto** | **full** | **half**}
	Example:
	`switch(config-if)# speed auto`
	`switch(config-if)#`
		Sets the duplex for the port-channel interface. The default is auto for autonegotiation.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode and returns to the configuration mode.

Step 6	**show interface port-channel** *channel-number* **Example:** `switch# show interface port-channel 2`	(Optional) Displays interface information for the specified port channel.
Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set port channel 2 to 100 Mb/s:

```
switch# configure terminal
switch (config)# interface port-channel 2
switch(config-if)# speed 100
```

Configuring Load Balancing Using Port Channels

You can configure the load-balancing algorithm for port channels that applies to the entire device.

Note Use the **no port-channel load-balance** command to restore the default load-balancing algorithm of source-dest-mac for non-IP traffic and source-dest-ip for IP traffic.

Command	Purpose
no port-channel load-balance **Example:** `switch(config)# no port-channel load-balance`	Restores the default load-balancing algorithm.

Before you begin

Enable LACP if you want LACP-based port channels.

SUMMARY STEPS

1. **configure terminal**
2. **port-channel load-balance** *method* {**dst ip** | **dst ip-gre** | **dst ip-l4port** | **dst ip-l4port-vlan** | **dst ip-vlan** | **dst l4port** | **dst mac** | **src ip** | **src ip-gre** | **src ip-l4port** | **src ip-l4port-vlan** | **src ip-vlan** | **src l4port** | **src mac** | **src-dst ip** | **src-dst ip-gre** | **src-dst ip-l4port** [**symmetric**] | **src-dst ip-l4port-vlan** | **src-dst ip-vlan** | **src-dst l4port** | **src-dst mac**} [**fex** {*fex-range* | *all*}] [**rotate** *rotate*]
3. **show port-channel load-balance**
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**port-channel load-balance** *method* {**dst ip** | **dst ip-gre** | **dst ip-l4port** | **dst ip-l4port-vlan** | **dst ip-vlan** | **dst l4port** | **dst mac** | **src ip** | **src ip-gre** | **src ip-l4port** | **src ip-l4port-vlan** | **src ip-vlan** | **src l4port** | **src mac** | **src-dst ip** | **src-dst ip-gre** | **src-dst ip-l4port** [**symmetric**] | **src-dst ip-l4port-vlan** | **src-dst ip-vlan** | **src-dst l4port** | **src-dst mac**} [**fex** {*fex-range* | *all*}] [**rotate** *rotate*] **Example:** • `switch(config)# port-channel load-balance src-dst mac` `switch(config)#` • `switch(config)# no port-channel load-balance src-dst mac` `switch(config)#`	Specifies the load-balancing algorithm for the device. The range depends on the device. The default for Layer 3 is **src-dst ip-l4port** for both IPv4 and IPv6, and the default for non-IP is **src-dst mac**. **Note** Only the following load-balancing algorithms support symmetric hashing: • src-dst ip • src-dst ip-l4port
Step 3	**show port-channel load-balance** **Example:** `switch(config-router)# show port-channel load-balance`	(Optional) Displays the port-channel load-balancing algorithm.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling LACP

LACP is disabled by default; you must enable LACP before you begin LACP configuration. You cannot disable LACP while any LACP configuration is present.

LACP learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once LACP identifies correctly matched Ethernet links, it group the links into a port channel. The port channel is then added to the spanning tree as a single bridge port.

To configure LACP, you must do the following:

- Enable LACP globally by using the **feature lacp** command.
- You can use different modes for different interfaces within the same LACP-enabled port channel. You can change the mode between **active** and **passive** for an interface only if it is the only interface that is designated to the specified channel group.

SUMMARY STEPS

1. **configure terminal**
2. **feature lacp**
3. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**feature lacp** **Example:** `switch(config)# feature lacp`	Enables LACP on the device.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to enable LACP:

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

Configuring LACP Port-Channel Port Modes

After you enable LACP, you can configure the channel mode for each individual link in the LACP port channel as **active** or **passive**. This channel configuration mode allows the link to operate with LACP.

When you configure port channels with no associated aggregation protocol, all interfaces on both sides of the link remain in the **on** channel mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **channel-group** *number* **mode** {**active** | **on** | **passive**}
4. **show port-channel summary**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.

Step 3	**channel-group** *number* **mode** {**active** | **on** | **passive**} **Example:** `switch(config-if)# channel-group 5 mode active`	Specifies the port mode for the link in a port channel. After LACP is enabled, you configure each link or the entire channel as active or passive. When you run port channels with no associated aggregation protocol, the port-channel mode is always on. The default port-channel mode is **on**.
Step 4	**show port-channel summary** **Example:** `switch(config-if)# show port-channel summary`	(Optional) Displays summary information about the port channels.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the LACP-enabled interface to the active port-channel mode for Ethernet interface 1/4 in channel group 5:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# channel-group 5 mode active
```

Configuring LACP Port-Channel Minimum Links

You can configure the LACP minimum links feature. Although minimum links and maxbundles work only in LACP, you can enter the CLI commands for these features for non-LACP port channels, but these commands are nonoperational.

Note Use the **no lacp min-links** command to restore the default port-channel minimum links configuration.

Command	Purpose
no lacp min-links **Example:** `switch(config)# no lacp min-links`	Restores the default port-channel minimum links configuration.

Before you begin

Ensure that you are in the correct port-channel interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*
3. **lacp min-links** *number*
4. **show running-config interface port-channel** *number*

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 3`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**lacp min-links** *number* **Example:** `switch(config-if)# lacp min-links 3`	Specifies the port-channel interface to configure the number of minimum links. The range is from 1 to 16.
Step 4	**show running-config interface port-channel** *number* **Example:** `switch(config-if)# show running-config interface port-channel 3`	(Optional) Displays the port-channel minimum links configuration.

Example

This example shows how to configure the minimum number of port-channel member interfaces to be up/active for the port-channel to be up/active:

```
switch# configure terminal
switch(config)# interface port-channel 3
switch(config-if)# lacp min-links 3
```

Configuring the LACP Port-Channel MaxBundle

You can configure the LACP maxbundle feature. Although minimum links and maxbundles work only in LACP, you can enter the CLI commands for these features for non-LACP port channels, but these commands are nonoperational.

Note Use the **no lacp max-bundle** command to restore the default port-channel max-bundle configuration.

Command	Purpose
no lacp max-bundle **Example:** `switch(config)# no lacp max-bundle`	Restores the default port-channel max-bundle configuration.

Before you begin

Ensure that you are in the correct port-channel interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*
3. **lacp max-bundle** *number*
4. **show running-config interface port-channel** *number*

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 3`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

	Command or Action	Purpose
Step 3	**lacp max-bundle** *number* **Example:** `switch(config-if)# lacp max-bundle`	Specifies the port-channel interface to configure max-bundle. The default value for the port-channel max-bundle is 16. The allowed range is from 1 to 32. **Note** Even if the default value is 16, the number of active members in a port channel is the minimum of the pc_max_links_config and pc_max_active_members that is allowed in the port channel.
Step 4	**show running-config interface port-channel** *number* **Example:** `switch(config-if)# show running-config interface port-channel 3`	(Optional) Displays the port-channel max-bundle configuration.

Example

This example shows how to configure the port channel interface max-bundle:

```
switch# configure terminal
switch(config)# interface port-channel 3
switch(config-if)# lacp max-bundle 3
```

Configuring the LACP Fast Timer Rate

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the **lacp rate** command to set the rate at which LACP control packets are sent to an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces.

Note We do not recommend changing the LACP timer rate. HA and SSO are not supported when the LACP fast rate timer is configured.

Note Configuring **lacp rate fast** is not recommended on the vPC Peer-Links. When **lacp rate fast** is configured on the vPC Peer-Link member interfaces, an alert is displayed in the syslog messages only when the LACP logging level is set to 5.

Before you begin

Ensure that you have enabled the LACP feature.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **lacp rate fast**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure and enters the interface configuration mode.

Step 3	**lacp rate fast** **Example:** `switch(config-if)# lacp rate fast`	Configures the fast rate (one second) at which LACP control packets are sent to an LACP-supported interface. To reset the timeout rate to its default, use the **no** form of the command.

Example

This example shows how to configure the LACP fast rate on Ethernet interface 1/4:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# lacp rate fast
```

This example shows how to restore the LACP default rate (30 seconds) on Ethernet interface 1/4.

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

Configuring the LACP System Priority

The LACP system ID is the combination of the LACP system priority value and the MAC address.

Before you begin

Enable LACP.

SUMMARY STEPS

1. **configure terminal**
2. **lacp system-priority** *priority*

3. **show lacp system-identifier**
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**lacp system-priority** *priority* **Example:** `switch(config)# lacp system-priority 40000`	Configures the system priority for use with LACP. Valid values are from 1 through 65535, and higher numbers have a lower priority. The default value is 32768. **Note** Each VDC has a different LACP system ID because the software adds the MAC address to this configured value.
Step 3	**show lacp system-identifier** **Example:** `switch(config-if)# show lacp system-identifier`	(Optional) Displays the LACP system identifier.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the LACP system priority to 2500:

```
switch# configure terminal
switch(config)# lacp system-priority 2500
```

Configuring the LACP Port Priority

When you enable LACP, you can configure each link in the LACP port channel for the port priority.

Before you begin

Enable LACP.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **lacp port-priority** *priority*
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.

Step 3	**lacp port-priority** *priority* **Example:** `switch(config-if)# lacp port-priority 40000`	Configures the port priority for use with LACP. Valid values are from 1 through 65535, and higher numbers have a lower priority. The default value is 32768.
Step 4	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the LACP port priority for Ethernet interface 1/4 to 40000:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# lacp port-priority 40000
```

Disabling LACP Graceful Convergence

By default, LACP graceful convergence is enabled. In situations where you need to support LACP interoperability with devices where the graceful failover defaults may delay the time taken for a disabled port to be brought down or cause traffic from the peer to be lost, you can disable convergence. If the downstream access switch is not a Cisco Nexus device, disable the LACP graceful convergence option.

Note The port channel has to be in the administratively down state before the command can be run.

Before you begin

Enable LACP.

SUMMARY STEPS

1. **configure terminal**

2. **interface port-channel** *number*
3. **shutdown**
4. **no lacp graceful-convergence**
5. **no shutdown**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)#`
		Specifies the port channel interface to configure and enters the interface configuration mode.

Step 3	**shutdown** **Example:** `switch(config-if) shutdown`	Administratively shuts down the port channel.
Step 4	**no lacp graceful-convergence** **Example:** `switch(config-if)# no lacp graceful-convergence`	Disables LACP graceful convergence on the port channel.
Step 5	**no shutdown** **Example:** `switch(config-if) no shutdown`	Brings the port channel administratively up.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to disable LACP graceful convergence on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# no lacp graceful-convergence
switch(config-if)# no shutdown
```

Reenabling LACP Graceful Convergence

If the default LACP graceful convergence is once again required, you can reenable convergence.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*
3. **shutdown**
4. **lacp graceful-convergence**
5. **no shutdown**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)#`
		Specifies the port channel interface to configure and enters the interface configuration mode.

Step 3	**shutdown** **Example:** `switch(config-if) shutdown`	Administratively shuts down the port channel.
Step 4	**lacp graceful-convergence** **Example:** `switch(config-if)# lacp graceful-convergence`	Enables LACP graceful convergence on the port channel.
Step 5	**no shutdown** **Example:** `switch(config-if) no shutdown`	Brings the port channel administratively up.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to enable LACP graceful convergence on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# lacp graceful-convergence
switch(config-if)# no shutdown
```

Disabling LACP Suspend Individual

LACP sets a port to the suspended state if it does not receive an LACP PDU from the peer. This process can cause some servers to fail to boot up as they require LACP to logically bring up the port.

Note You should only enter the **lacp suspend-individual** command on edge ports. The port channel has to be in the administratively down state before you can use this command.

Before you begin

Enable LACP.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*
3. **shutdown**
4. **no lacp suspend-individual**
5. **no shutdown**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)#`
		Specifies the port channel interface to configure and enters the interface configuration mode.

Step 3	**shutdown** **Example:** `switch(config-if) shutdown`	Administratively shuts down the port channel.
Step 4	**no lacp suspend-individual** **Example:**	Disables LACP individual port suspension behavior on the port channel.

	Command or Action	Purpose
	`switch(config-if)# no lacp suspend-individual`	
Step 5	**no shutdown** **Example:** `switch(config-if) no shutdown`	Brings the port channel administratively up.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to disable LACP individual port suspension on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# no lacp suspend-individual
switch(config-if)# no shutdown
```

Reenabling LACP Suspend Individual

You can reenable the default LACP individual port suspension.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*
3. **shutdown**
4. **lacp suspend-individual**
5. **no shutdown**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)#`
		Specifies the port channel interface to configure and enters the interface configuration mode.

	Command or Action	Purpose
Step 3	**shutdown** **Example:** `switch(config-if) shutdown`	Administratively shuts down the port channel.
Step 4	**lacp suspend-individual** **Example:** `switch(config-if)# lacp suspend-individual`	Enables LACP individual port suspension behavior on the port channel.
Step 5	**no shutdown** **Example:** `switch(config-if) no shutdown`	Brings the port channel administratively up.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to reenable the LACP individual port suspension on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# lacp suspend-individual
switch(config-if)# no shutdown
```

Configuring Port Channel Hash Distribution

Cisco NX-OS supports the adaptive and fixed hash distribution configuration for both global and port-channel levels. This option minimizes traffic disruption by minimizing Result Bundle Hash (RBH) distribution changes when members come up or go down so that flows that are mapped to unchange RBH values continue to flow through the same links. The port-channel level configuration overrules the global configuration. The default configuration is adaptive globally, and there is no configuration for each port channel, so there is no change during an ISSU. No ports are flapped when the command is applied, and the configuration takes effect at the next member link change event. Both modes work with RBH module or non-module schemes.

During an ISSD to a lower version that does not support this feature, you must disable this feature if the fixed mode command is being used globally or if there is a port-channel level configuration.

Configuring Port Channel Hash Distribution at the Global Level

SUMMARY STEPS

1. **configure terminal**
2. **no port-channel hash-distribution** {**adaptive** | **fixed**}
3. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**no port-channel hash-distribution** {**adaptive** | **fixed**}
	Example:
	`switch(config)# port-channel hash-distribution adaptive`
	`switch(config)#`
		Specifies the port-channel hash distribution at the global level.
		The default is adaptive mode.
		The command does not take effect until the next member link event (link down/up/no shutdown/shutdown). (`Do you still want to continue(y/n)? [yes]`)

Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure hash distribution at the global level:

```
switch# configure terminal
switch(config)# no port-channel hash-distribution fixed
```

Configuring Port Channel Hash Distribution at the Port Channel Level

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** {*channel-number* | *range*}
3. **no port-channel port hash-distribution** {**adaptive** | **fixed**}
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** {*channel-number* | *range*} **Example:**	Specifies the interface to configure, and enters the interface configuration mode.

	Command or Action	Purpose
	`switch# interface port-channel 4` `switch(config-if)#`	

Step 3	**no port-channel port hash-distribution** {**adaptive** | **fixed**}
	Example:
	`switch(config-if)# port-channel port hash-distribution adaptive`
	`switch(config-if)`
		Specifies the port-channel hash distribution at the port channel level.
		There is no default.
		The command does not take effect until the next member link event (link down/up/no shutdown/shutdown). (`Do you still want to continue(y/n)? [yes]`)

Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure hash distribution as a global-level command:

```
switch# configure terminal
switch(config)# no port-channel hash-distribution fixed
```

Verifying the Port-Channel Configuration

To display port-channel configuration information, perform one of the following tasks:

Command	Purpose
show interface port-channel *channel-number*	Displays the status of a port-channel interface.
show feature	Displays enabled features.
load- interval {**interval** *seconds* {**1** | **2** | **3**}}	Sets three different sampling intervals to bit-rate and packet-rate statistics.
show port-channel compatibility-parameters	Displays the parameters that must be the same among the member ports in order to join a port channel.
show port-channel database [**interface port-channel** *channel-number*]	Displays the aggregation state for one or more port-channel interfaces.
show port-channel load-balance	Displays the type of load balancing in use for port channels.
show port-channel summary	Displays a summary for the port-channel interfaces.
show port-channel traffic	Displays the traffic statistics for port channels.
show port-channel usage	Displays the range of used and unused channel numbers.

Command	Purpose
show lacp {**counters** [**interface port-channel** *channel-number*] | [**interface** *type/slot*] | **neighbor** [**interface port-channel** *channel-number*] | **port-channel** [**interface port-channel** *channel-number*] | **system-identifier**]]}	Displays information about LACP.
show running-config interface port-channel *channel-number*	Displays information about the running configuration of the port-channel.

Monitoring the Port-Channel Interface Configuration

Use the following commands to display port-channel interface configuration information.

Command	Purpose
clear counters interface port-channel *channel-number*	Clears the counters.
clear lacp counters [**interface port-channel** *channel-number*]	Clears the LACP counters.
load- interval {**interval** *seconds* {**1** | **2** | **3**}}	Sets three different sampling intervals to bit-rate and packet-rate statistics.
show interface counters [**module** *module*]	Displays input and output octets unicast packets, multicast packets, and broadcast packets.
show interface counters detailed [**all**]	Displays input packets, bytes, and multicast and output packets and bytes.
show interface counters errors [**module** *module*]	Displays information about the number of error packets.
show lacp counters	Displays statistics for LACP.

Example Configurations for Port Channels

This example shows how to create an LACP port channel and add two Layer 2 interfaces to that port channel:

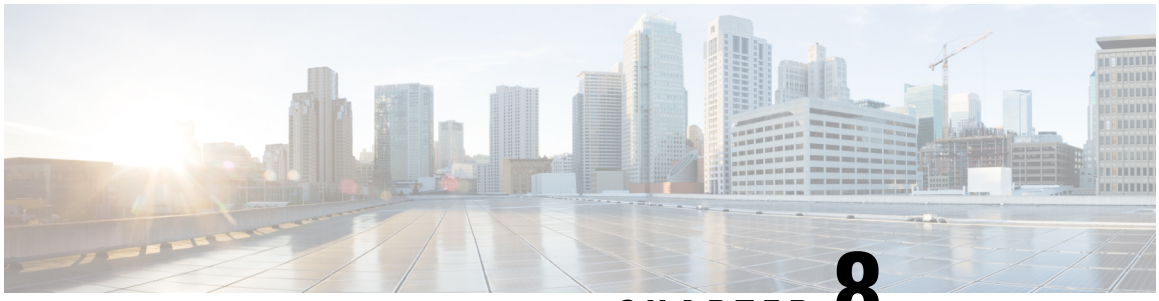
```
switch# configure terminal
switch (config)# feature lacp
switch (config)# interface port-channel 5
switch (config-if)# interface ethernet 1/4
switch(config-if)# switchport
switch(config-if)# channel-group 5 mode active
switch(config-if)# lacp port priority 40000
switch(config-if)# interface ethernet 1/7
switch(config-if)# switchport
switch(config-if)# channel-group 5 mode
```

This example shows how to add two Layer 3 interfaces to a channel group. The Cisco NX-OS software automatically creates the port channel:

```
switch# configure terminal
switch (config)# interface ethernet 1/5
switch(config-if)# no switchport
switch(config-if)# no ip address
switch(config-if)# channel-group 6 mode active
switch (config)# interface ethernet 2/5
switch(config-if)# no switchport
switch(config-if)# no ip address
switch(config-if)# channel-group 6 mode active
switch (config)# interface port-channel 6
switch(config-if)# ip address 192.0.2.1/8
```

Related Documents

Related Topic	Document Title
System management	*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*
High availability	*Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*
Licensing	*Cisco NX-OS Licensing Guide*



CHAPTER 8

Configuring vPCs

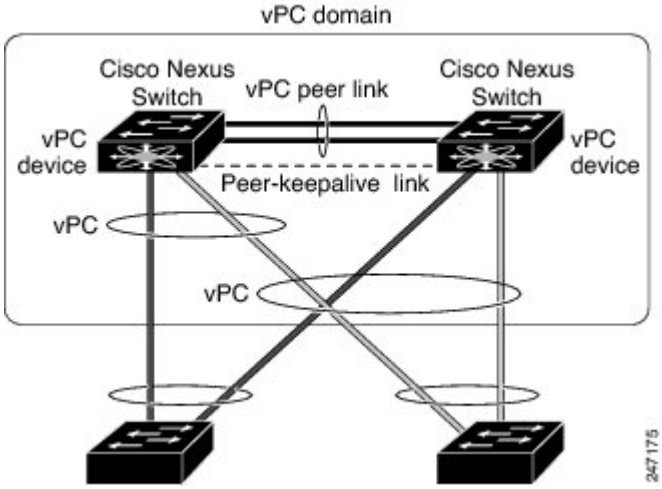
This chapter describes how to configure virtual port channels (vPCs) on Cisco NX-OS devices.

Information About vPCs

vPC Overview

A virtual port channel (vPC) allows links that are physically connected to two Cisco Nexus 9000 Series devices to appear as a single port channel by a third device (see figure). The third device can be a switch, server, or any other networking device that supports port channels. A vPC can provide Layer 2 multipathing, which allows you to create redundancy and increase the bisectional bandwidth by enabling multiple parallel paths between nodes and allowing load balancing traffic.

Figure 10: vPC Architecture



You can use only Layer 2 port channels in the vPC.You configure the port channels by using one of the following:

- No protocol
- Link Aggregation Control Protocol (LACP)

When you configure the port channels in a vPC—including the vPC Peer-Link channel—without using LACP, each device can have up to 32 active links in a single port channel. When you configure the port channels in a vPC—including the vPC Peer-Link channels—using LACP, each device can have 32 active links and eight standby links in a single port channel. (See the “vPC Interactions with Other Features” section for more information on using LACP and vPCs.)

Note You must enable the vPC feature before you can configure or run the vPC functionality.

After you enable the vPC functionality, you create the peer-keepalive link, which sends heartbeat messages between the two vPC peer devices.

You can create a vPC Peer-Link by configuring a port channel on one Cisco Nexus 9000 Series chassis by using two or more Ethernet ports higher speed than 1-Gigabit Ethernet. To ensure that you have the correct hardware to enable and run a vPC, enter the **show hardware feature-capability** command. If you see an X across from the vPC in your command output, your hardware cannot enable the vPC feature.

We recommend that you configure the vPC Peer-Link Layer 2 port channels as trunks. On another Cisco Nexus 9000 Series chassis, you configure another port channel again using two or more Ethernet ports withspeed higher than 1-Gigabit in the dedicated port mode. Connecting these two port channels creates a vPC Peer-Link in which the two linked Cisco Nexus devices appear as one device to a third device. The third device, or downstream device, can be a switch, server, or any other networking device that uses a regular port channel to connect to the vPC.

For modular Cisco Nexus 9500 switches, we recommend that you configure the vPC Peer-Links on dedicated ports of different modules to reduce the possibility of a failure. For the best resiliency scenario, use at least two modules.

You can use any of the interfaces of the Nexus 9000 device for the vPC Peer-Link.If you must configure all the vPC Peer-Links and core-facing interfaces on a single module, you should configure a track object that is associated with the Layer 3 link to the core and on all the links on the vPC Peer-Link on both vPC peer devices.

The vPC domain includes both vPC peer devices, the vPC peer-keepalive link, the vPC Peer-Link, and all of the port channels in the vPC domain connected to the downstream device. You can have only one vPC domain ID on each device.

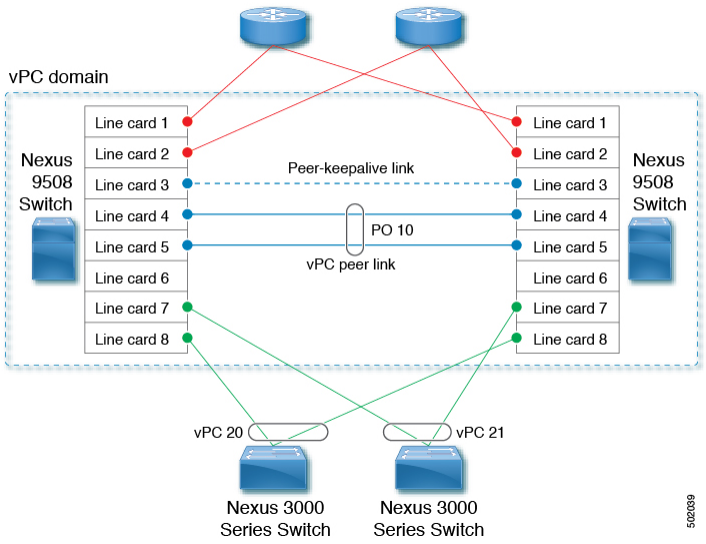
In this version, you can connect each downstream device to a single vPC domain ID using a single port channel.

Note Devices attached to a vPC domain using port channels should be connected to both of vPC peers.

A vPC (see figure) provides the following benefits:

- Allows a single device to use a port channel across two upstream devices
- Eliminates Spanning Tree Protocol (STP) blocked ports
- Provides a loop-free topology
- Uses all available uplink bandwidth
- Provides fast convergence if either the link or a device fails
- Provides link-level resiliency
- Assures high availability

Figure 11: vPC Interfaces



Hitless vPC Role Change

A virtual port channel (vPC) allows links that are physically connected to two different Cisco Nexus 9000 Series devices to appear as a single port channel. The vPC role change feature enables you switch vPC roles

between vPC peers without impacting traffic flow. The vPC role switching is done based on the role priority value of the device under the vPC domain. A vPC peer device with lower role priority is selected as the primary vPC device during the vPC Role switch. You can use the vpc role preempt command to switch vPC role between peers.

For information about how to configure Hitless vPC Role Change, see Configuring Hitless vPC Role Change, on page 223.

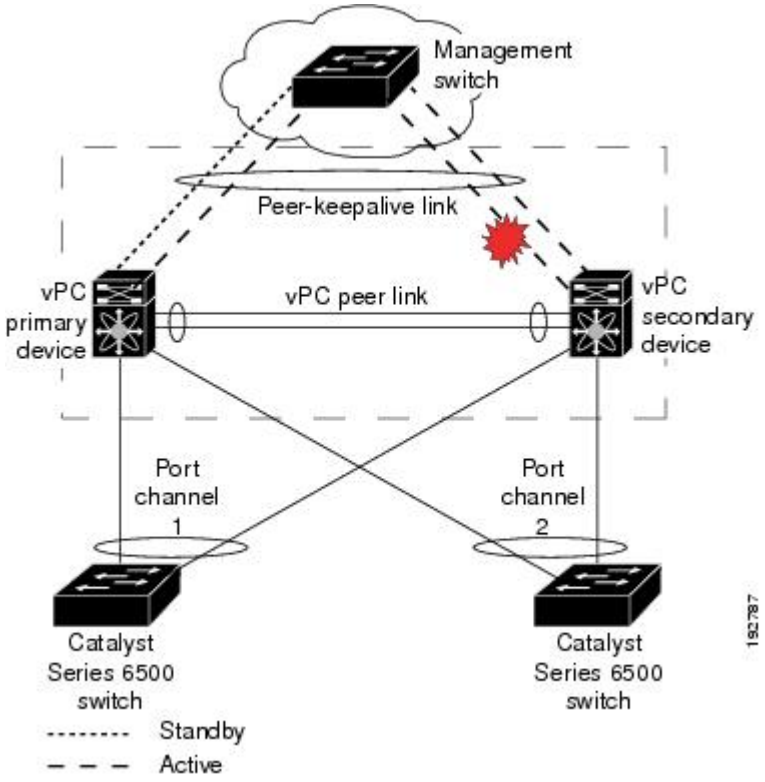
vPC Terminology

The terminology used in vPCs is as follows:

- vPC—The combined port channel between the vPC peer devices and the downstream device.
- vPC peer device—One of a pair of devices that are connected with the special port channel known as the vPC Peer-Link.
- vPC Peer-Link—The link used to synchronize state between the vPC peer devices. This link must use a 10-Gigabit Ethernet interface at a minimum. Higher-bandwidth interfaces (such as 25-Gigabit Ethernet, 40-Gigabit Ethernet, 100-Gigabit Ethernet, and so on) may also be used.
- vPC member port—An interface that belongs to a vPC.
- Host vPC port—A Fabric Extender host interfaces that belongs to a vPC.
- vPC domain—This domain includes both vPC peer devices, the vPC peer-keepalive link, and all of the port channels in the vPC connected to the downstream devices. It is also associated to the configuration mode that you must use to assign vPC global parameters.
- vPC peer-keepalive link—The peer-keepalive link monitors the vitality of a vPC peer Cisco Nexus 9000 Series device. The peer-keepalive link sends configurable, periodic keepalive messages between vPC peer devices.

  We recommend that you associate a peer-keepalive link to a separate virtual routing and forwarding (VRF) instance that is mapped to a Layer 3 interface in each vPC peer device. If you do not configure a separate VRF, the system uses the management VRF by default. However, if you use the management interfaces for the peer-keepalive link, you must put a management switch connected to both the active and standby management ports on each vPC peer device (see figure).

Figure 12: Separate Switch Required to Connect Management Ports for vPC Peer-Keepalive Link



No data or synchronization traffic moves over the vPC peer-keepalive link; the only traffic on this link is a message that indicates that the originating switch is operating and running a vPC.

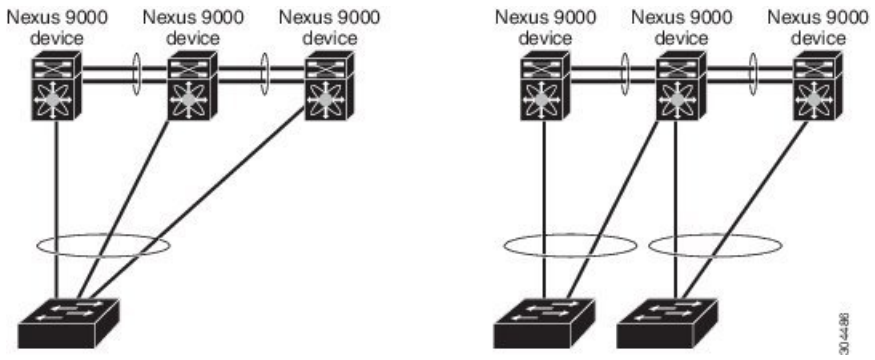
- vPC member port—Interfaces that belong to the vPCs.
- Dual-active— Both vPC peers act as primary. This situation occurs when the peer-keepalive and vPC Peer-Link go down when both the peers are still active. In this case, the secondary vPC assumes that the primary vPC is inactive and acts as the primary vPC.
- Recovery—When the peer-keepalive and the vPC Peer-Link come up, one switch becomes the secondary vPC. On the switch that becomes the secondary vPC, the vPC links go down and come back up.

vPC Peer-Link Overview

You can have only two devices as vPC peers; each device can serve as a vPC peer to only one other vPC peer. The vPC peer devices can also have non-vPC links to other devices.

See the following figure for invalid vPC peer configurations.

Figure 13: vPC Peer Configurations That Are Not Allowed



To make a valid configuration, you first configure a port channel on each device and then configure the vPC domain. You assign the port channel on each device as a vPC Peer-Link, using the same vPC domain ID. For redundancy, we recommend that you should configure at least two of the dedicated ports into the port channel because if one of the interfaces in the vPC Peer-Link fails, the device automatically falls back to use another interface in the vPC Peer-Link.

Note We recommend that you configure the Layer 2 port channels in trunk mode.

Many operational parameters and configuration parameters must be the same in each device connected by a vPC Peer-Link (see the Compatibility Parameters for vPC Interfaces section). Because each device is completely independent on the management plane, you must ensure that the devices are compatible on the critical parameters. vPC peer devices have separate control planes. After configuring the vPC Peer-Link, you should display the configuration on each vPC peer device to ensure that the configurations are compatible.

Note You must ensure that the two devices connected by the vPC Peer-Link have certain identical operational and configuration parameters. For more information on required configuration consistency, see the Compatibility Parameters for vPC Interfaces section.

When you configure the vPC Peer-Link, the vPC peer devices negotiate that one of the connected devices is the primary device and the other connected device is the secondary device (see the “Configuring vPCs” section). The Cisco NX-OS software uses the lowest MAC address to elect the primary device. The software takes different actions on each device—that is, the primary and secondary—only in certain failover conditions. If the primary device fails, the secondary device becomes the new primary device when the system recovers, and the previously primary device is now the secondary device.

You can also configure which of the vPC devices is the primary device. Changing the priority of the vPC peer devices can cause the interfaces in your network to go up and down. If you want to configure the role priority again to make one vPC device the primary device, configure the role priority on both the primary vPC device with a lower priority value and the secondary vPC device with the higher value. Then, shut down the port channel that is the vPC Peer-Link on both devices by entering the **shutdown** command, and finally reenable the port channel on both devices by entering the **no shutdown** command.

Note We recommend that you use two different modules for redundancy on each vPC peer device on each vPC Peer-Link.

The software keeps all traffic that forwards across the vPC peer devices as local traffic. A packet that ingresses the port channel uses one of the local links rather than moving across the vPC Peer-Link. Unknown unicast, multicast, and broadcast traffic (including STP BPDUs) are flooded across the vPC Peer-Link. The software keeps the multicast forwarding state synchronized on both of the vPC peer devices.

You can configure any of the standard load-balancing schemes on both the vPC Peer-Link devices and the downstream device (see the *Configuring Port Channels* chapter for information about load balancing).

Configuration information flows across the vPC Peer-Links using the Cisco Fabric Services over Ethernet (CFSoE) protocol. (See the CFSoE, on page 183 section for more information about CFSoE.)

All MAC addresses for those VLANs configured on both devices are synchronized between vPC peer devices. The software uses CFSoE for this synchronization. (See the CFSoE, on page 183 section for information about CFSoE.)

If the vPC Peer-Link fails, the software checks the status of the remote vPC peer device using the peer-keepalive link, which is a link between vPC peer devices that ensures that both devices are up. If the vPC peer device is up, the secondary vPC device disables all vPC ports on its device, to prevent loops and disappearing or flooding traffic. The data then forwards down the remaining active links of the port channel.

The software learns of a vPC peer device failure when the keepalive messages are not returned over the peer-keepalive link.

Use a separate link (vPC peer-keepalive link) to send configurable keepalive messages between the vPC peer devices. The keepalive messages on the vPC peer-keepalive link determines whether a failure is on the vPC Peer-Link only or on the vPC peer device. The keepalive messages are used only when all the links in the vPC Peer-Link fail. See the “Peer-Keepalive Link and Messages” section for information about the keepalive message.

Features That You Must Manually Configure on the Primary and Secondary Devices

You must manually configure the following features to conform to the primary/secondary mapping of each of the vPC peer devices:

- STP root—Configure the primary vPC peer device as the STP primary root device and configure the vPC secondary device to be the STP secondary root device. See the “vPC Peer-Links and STP” section for more information about vPCs and STP.
  - We recommend that you configure the vPC Peer-Link interfaces as STP network ports so that Bridge Assurance is enabled on all vPC Peer-Links.
  - We recommend that you configure Rapid per VLAN Spanning Tree plus (PVST+) so that the primary device is the root for all VLANs and configure Multiple Spanning Tree (MST) so that the primary device is the root for all instances.
- Layer 3 VLAN network interface—Configure Layer 3 connectivity from each vPC peer device by configuring a VLAN network interface for the same VLAN from both devices.
- HSRP active—If you want to use Hot Standby Router Protocol (HSRP) and VLAN interfaces on the vPC peer devices, configure the primary vPC peer device with the HSRP active highest priority. Configure the secondary device to be the HSRP standby and ensure that you have VLAN interfaces on each vPC

device that are in the same administrative and operational mode. (See the "vPC Peer-Links and Routing" section for more information on vPC and HSRP.)

While you configure Unidirectional Link Detection (UDLD), note the following recommendations:

Configuring Layer 3 Backup Routes on a vPC Peer-Link

You can use VLAN network interfaces on the vPC peer devices to link to Layer 3 of the network for such applications as HSRP and PIM. Ensure that you have a VLAN network interface configured on each peer device and that the interface is connected to the same VLAN on each device. Also, each VLAN interface must be in the same administrative and operational mode. For more information about configuring VLAN network interfaces, see the "Configuring Layer 3 Interfaces" chapter.

If a failover occurs on the vPC Peer-Link, the VLAN interfaces on the vPC peer devices are also affected. If a vPC Peer-Link fails, the system brings down associated VLAN interfaces on the secondary vPC peer device.

You can ensure that specified VLAN interfaces do not go down on the vPC secondary device when the vPC Peer-Link fails.

Peer-Keepalive Link and Messages

The Cisco NX-OS software uses the peer-keepalive link between the vPC peers to transmit periodic, configurable keepalive messages. You must have Layer 3 connectivity between the peer devices to transmit these messages; the system cannot bring up the vPC Peer-Link unless the peer-keepalive link is already up and running.

Note We recommend that you associate the vPC peer-keepalive link to a separate VRF mapped to a Layer 3 interface in each vPC peer device. If you do not configure a separate VRF, the system uses the management VRF and management ports by default. Do not use the vPC Peer-Link itself to send and receive vPC peer-keepalive messages.

If one of the vPC peer devices fails, the vPC peer device on the other side of the vPC Peer-Link senses the failure by not receiving any peer-keepalive messages. The default interval time for the vPC peer-keepalive message is 1 second, and you can configure the interval between 400 milliseconds and 10 seconds.

You can configure a hold-timeout value with a range of 3 to 10 seconds; the default hold-timeout value is 3 seconds. This timer starts when the vPC Peer-Link goes down. During this hold-timeout period, the secondary vPC peer device ignores vPC peer-keepalive messages, which ensures that network convergence occurs before a vPC action takes place. The purpose of the hold-timeout period is to prevent false-positive cases.

You can also configure a timeout value with a range of 3 to 20 seconds; the default timeout value is 5 seconds. This timer starts at the end of the hold-timeout interval. During the timeout period, the secondary vPC peer device checks for vPC peer-keepalive hello messages from the primary vPC peer device. If the secondary vPC peer device receives a single hello message, that device disables all vPC interfaces on the secondary vPC peer device.

The difference between the hold-timeout and the timeout parameters is as follows:

- During the hold-timeout, the vPC secondary device does not take any action based on any keepalive messages received, which prevents the system taking action when the keepalive might be received just temporarily, such as if a supervisor fails a few seconds after the vPC Peer-Link goes down.

- During the timeout, the vPC secondary device takes action to become the vPC primary device if no keepalive message is received by the end of the configured interval.

See the "Configuring vPC Keepalive Link and Messages" section for information about configuring the timer for the keepalive messages.

Note Ensure that both the source and destination IP addresses used for the peer-keepalive messages are unique in your network and these IP addresses are reachable from the VRF associated with the vPC peer-keepalive link.

Peer-keepalive IP addresses must be global unicast addresses. Link-local addresses are not supported.

Use the command-line interface (CLI) to configure the interfaces you are using the vPC peer-keepalive messages as trusted ports. Leave the precedence at the default (6) or configure it higher.

vPC Peer-Gateway

You can configure vPC peer devices to act as the gateway even for packets that are destined to the vPC peer device's MAC address.

Use the **peer-gateway** command to configure this feature.

Note The **peer-gateway exclude-vlan** command that is used when configuring a VLAN interface for Layer 3 backup routing on vPC peer devices is not supported.

Some network-attached storage (NAS) devices or load balancers might have features that help to optimize the performances of particular applications. These features enable the device to avoid a routing-table lookup when responding to a request that originated from a host that is not locally attached to the same subnet. Such devices might reply to traffic using the MAC address of the sender Cisco Nexus 9000 Series device rather than the common HSRP gateway. This behavior is noncomplaint with some basic Ethernet RFC standards. Packets that reach a vPC device for the nonlocal router MAC address are sent across the vPC Peer-Link and could be dropped by the built in vPC loop avoidance mechanism if the final destination is behind another vPC.

The vPC peer-gateway capability allows a vPC switch to act as the active gateway for packets that are addressed to the router MAC address of the vPC peer. This feature enables local forwarding of packets without the need to cross the vPC Peer-Link. In this scenario, the feature optimizes use of the vPC Peer-Link and avoids potential traffic loss.

Configuring the peer-gateway feature must be done on both primary and secondary vPC peers and is nondisruptive to the operations of the device or to the vPC traffic. The vPC peer-gateway feature can be configured globally under the vPC domain submode.

When you enable this feature, Cisco NX-OS automatically disables IP redirects on all interface VLANs mapped over a vPC VLAN to avoid generation of IP redirect messages for packets switched through the peer gateway router.

Packets that arrive at the peer-gateway vPC device have their Time to Live (TTL) decremented, so that packets carrying a TTL of 1 might get dropped in transit due to TTL expiration. You should take this situation into account when the peer-gateway feature is enabled and particular network protocols that source packets with a TTL of 1 operate on a vPC VLAN.

vPC Domain

You can use the vPC domain ID to identify the vPC Peer-Links and the ports that are connected to the vPC downstream devices.

The vPC domain is also a configuration mode that you use to configure the keepalive messages and other vPC Peer-Link parameters rather than accept the default values. See the “Configuring vPCs” section for more information about configuring these parameters.

To create a vPC domain, you must first create a vPC domain ID on each vPC peer device using a number from 1 to 1000. You can have only one vPC domain per vPC peer.

You must explicitly configure the port channel that you want to act as the vPC Peer-Link on each device. You associate the port channel that you made a vPC Peer-Link on each device with the same vPC domain ID to form a single vPC domain. Within this domain, the system provides a loop-free topology and Layer 2 multipathing.

You can only configure these port channels and vPC Peer-Links statically. You can configure the port channels and vPC Peer-Links either using LACP or no protocol. We recommend that you use LACP with the interfaces in active mode to configure port channels in each vPC, which ensures an optimized, graceful recovery in a port-channel failover scenario and provides configuration checks against configuration mismatches among the port channels themselves.

The vPC peer devices use the vPC domain ID that you configure to automatically assign a unique vPC system MAC address. Each vPC domain has a unique MAC address that is used as a unique identifier for the specific vPC-related operations, although the devices use the vPC system MAC addresses only for link-scope operations, such as LACP. We recommend that you create each vPC domain within the contiguous Layer 2 network with a unique domain ID. You can also configure a specific MAC address for the vPC domain, rather than having the Cisco NX-OS software assign the address.

See the “vPC and Orphan Ports” section for more information about displaying the vPC MAC table.

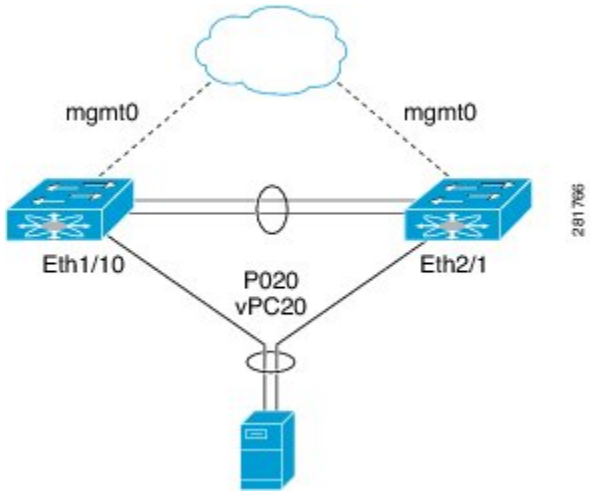
After you create a vPC domain, the Cisco NX-OS software creates a system priority for the vPC domain. You can also configure a specific system priority for the vPC domain.

Note When manually configuring the system priority, you must ensure that you assign the same priority value on both vPC peer devices. If the vPC peer devices have different system priority values, vPC does not come up.

vPC Topology

The following figure shows a basic configuration in which the Cisco Nexus 9000 Series device ports are directly connected to another switch or host and are configured as part of a port channel that becomes part of a vPC.

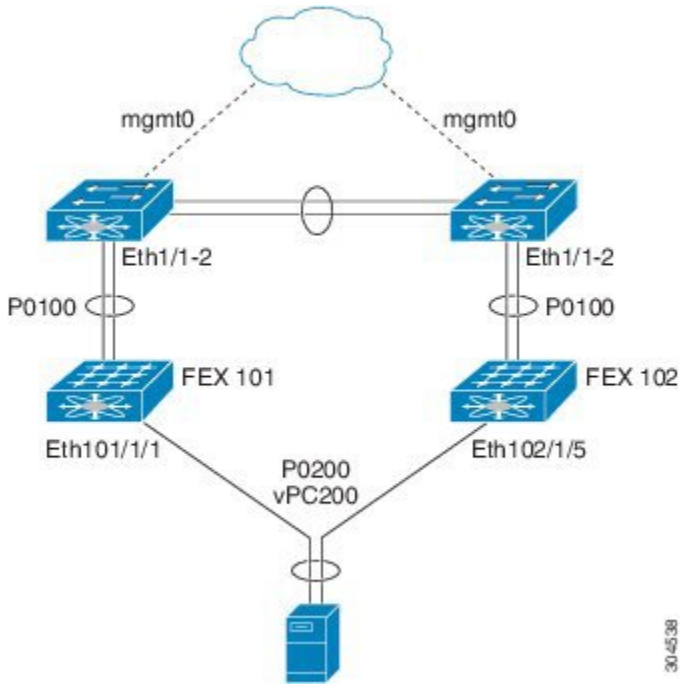
Figure 14: Switch vPC Topology



In the figure, vPC 20 is configured on port channel 20, which has Eth1/10 on the first device and Eth2/1 on the second as member ports.

You can configure a vPC from the peer devices through Fabric Extenders (FEXs) as shown in the figure.

Figure 15: FEX Straight-Through Topology (Host vPC)



In the figure, each FEX is single-homed (straight-through FEX topology) with a Cisco Nexus 9000 Series device. The host interfaces on this FEX are configured as port channels and those port channels are configured as vPCs. Eth101/1/1 and Eth102/1/5 are configured as members of PO200, and PO200 is configured for vPC 200.

In both topologies, port channels P020 and P0200 must be configured identically on the peer switches and configuration synchronization is used to synchronize the configurations of the vPC switches.

See the Cisco Nexus 2000 Series NX-OS Fabric Extender Configuration Guide for Cisco Nexus 9000 Series Switches for more information about configuring FEX ports.

Compatibility Parameters for vPC Interfaces

Many configuration and operational parameters must be identical on all interfaces in the vPC. We recommend that you configure the Layer 2 port channels that you use for the vPC Peer-Link in trunk mode.

After you enable the vPC feature and configure the vPC Peer-Link on both vPC peer devices, Cisco Fabric Services (CFS) messages provide a copy of the configuration on the local vPC peer device configuration to the remote vPC peer device. The system then determines whether any of the crucial configuration parameters differ on the two devices. (See the "vPC and Orphan Ports" section for more information about CFS.)

Note Enter the **show vpc consistency-parameters**command to display the configured values on all interfaces in the vPC. The displayed configurations are only those configurations that would limit the vPC Peer-Link and vPC from coming up.

Note The port channel compatibility parameters must be the same for all the port channel members on the physical switch. You cannot configure shared interfaces to be part of a vPC.

The compatibility check process for vPCs differs from the compatibility check for regular port channels.

See the "Configuring Port Channels" chapter for information about regular port channels.

Configuration Parameters That Must Be Identical

The configuration parameters in this section must be configured identically on both devices of the vPC Peer-Link; otherwise, the vPC moves fully or partially into a suspended mode.

Note You must ensure that all interfaces in the vPC have the identical operational and configuration parameters listed in this section.

Note Enter the **show vpc consistency-parameters** command to display the configured values on all interfaces in the vPC. The displayed configurations are only those configurations that would limit the vPC Peer-Link and vPC from coming up.

The devices automatically check for compatibility for some of these parameters on the vPC interfaces. The per-interface parameters must be consistent per interface, and the global parameters must be consistent globally:

- Port-channel mode: on, off, or active (port-channel mode can, however, be active/passive on each side of the vPC peer)

- Link speed per channel
- Duplex mode per channel
- Trunk mode per channel:
  - Native VLAN
  - VLANs allowed on trunk
  - Tagging of native VLAN traffic
- Spanning Tree Protocol (STP) mode
- STP region configuration for Multiple Spanning Tree
- Enable/disable state per VLAN
- STP global settings:
  - Bridge Assurance setting
  - Port type setting
  - Loop Guard settings
- STP interface settings:
  - Port type setting
  - Loop Guard
  - Root Guard
- Maximum Transmission Unit (MTU)

If any of these parameters are not enabled or defined on either device, the vPC consistency check ignores those parameters.

Note To ensure that none of the vPC interfaces are in the suspend mode, enter the **show vpc brief** and **show vpc consistency-parameters** commands and check the syslog messages.

Configuration Parameters That Should Be Identical

When any of the following parameters are not configured identically on both vPC peer devices, a misconfiguration might cause undesirable behavior in the traffic flow:

- MAC aging timers
- Static MAC entries
- VLAN interface—Each device on the end of the vPC Peer-Link must have a VLAN interface configured for the same VLAN on both ends and they must be in the same administrative and operational mode. Those VLANs configured on only one device of the vPC Peer-Link do not pass traffic using the vPC or vPC Peer-Link. You must create all VLANs on both the primary and secondary vPC devices, or the VLAN will be suspended.

- All ACL configurations and parameters
- Quality of Service (QoS) configuration and parameters
- STP interface settings:
  - BPDU Filter
  - BPDU Guard
  - Cost
  - Link type
  - Priority
  - VLANs (Rapid PVST+)
- Port security
- Cisco Trusted Security (CTS)
- Dynamic Host Configuration Protocol (DHCP) snooping
- Network Access Control (NAC)
- Dynamic ARP Inspection (DAI)
- IP source guard (IPSG)
- Internet Group Management Protocol (IGMP) snooping
- Hot Standby Routing Protocol (HSRP)
- Protocol Independent Multicast (PIM)
- All routing protocol configurations

To ensure that all the configuration parameters are compatible, we recommend that you display the configurations for each vPC peer device once you configure the vPC.

Consequences of Parameter Mismatches

You can configure the graceful consistency check feature, which suspends only the links on the secondary peer device when a mismatch is introduced in a working vPC. This feature is configurable only in the CLI and is enabled by default.

The graceful consistency-check command is configured by default.

As part of the consistency check of all parameters from the list of parameters that must be identical, the system checks the consistency of all VLANs.

The vPC remains operational, and only the inconsistent VLANs are brought down. This per-VLAN consistency check feature cannot be disabled and does not apply to Multiple Spanning Tree (MST) VLANs.

vPC Number

Once you have created the vPC domain ID and the vPC Peer-Link, you create port channels to attach the downstream device to each vPC peer device. That is, you create one port channel to the downstream device

from the primary vPC peer device and you create another port channel to the downstream device from the secondary peer device.

Note We recommend that you configure the ports on the downstream devices that connect to a host or a network device that is not functioning as a switch or a bridge as STP edge ports.

On each vPC peer device, you assign a vPC number to the port channel that connects to the downstream device. You will experience minimal traffic disruption when you are creating vPCs. To simplify the configuration, you can assign the vPC ID number to every port channel to be the same as the port channel itself (that is, vPC ID 10 for port channel 10).

Note The vPC number that you assign to the port channel that connects to the downstream device from the vPC peer device must be identical on both vPC peer devices.

Moving Other Port Channels into a vPC

Note You must attach a downstream device using a port channel to both vPC peer devices.

To connect to the downstream device, you create a port channel to the downstream device from the primary vPC peer device and you create another port channel to the downstream device from the secondary peer device. On each vPC peer device, you assign a vPC number to the port channel that connects to the downstream device. You will experience minimal traffic disruption when you are creating vPCs.

vPC Object Tracking

Note We recommend that you configure the vPC Peer-Links on dedicated ports of different modules on Cisco Nexus 9500 devices. This is recommended to reduce the possibility of a failure. For the best resiliency scenario, use at least two modules.

vPC object tracking is used to prevent traffic black-holing in case of failure of a module where both vPC Peer-Link and uplinks to the core resides. By tracking interface feature can suspend vPC on affected switch and prevent traffic black-holing.

If you must configure all the vPC Peer-Links and core-facing interfaces on a single module, you should configure, using the command-line interface, a track object and a track list that is associated with the Layer 3 link to the core and on all vPC Peer-Links on both vPC peer devices. You use this configuration to avoid dropping traffic if that particular module goes down because when all the tracked objects on the track list go down, the system does the following:

- Stops the vPC primary peer device sending peer-keepalive messages, which forces the vPC secondary peer device to take over.

- Brings down all the downstream vPCs on that vPC peer device, which forces all the traffic to be rerouted in the access switch toward the other vPC peer device.

Once you configure this feature and if the module fails, the system automatically suspends all the vPC links on the primary vPC peer device and stops the peer-keepalive messages. This action forces the vPC secondary device to take over the primary role and all the vPC traffic to go to this new vPC primary device until the system stabilizes.

You should create a track list that contains all the links to the core and all the vPC Peer-Links as its object. Enable tracking for the specified vPC domain for this track list. Apply this same configuration to the other vPC peer device. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for information about configuring object tracking and track lists.

Note This example uses Boolean OR in the track list and forces all traffic to the vPC peer device only for a complete module failure. If you want to trigger a switchover when any core interface or vPC Peer-Link goes down, use a Boolean AND in the torack list below.

To configure a track list to switch over a vPC to the remote peer when all related interfaces on a single module fail, follow these steps:

1. Configure track objects on an interface (Layer 3 to core) and on a port channel (vPC Peer-Link).

```
switch(config-if)# track 35 interface ethernet 8/35 line-protocol
switch(config-track)# track 23 interface ethernet 8/33 line-protocol
switch(config)# track 55 interface port-channel 100 line-protocol
```

2. Create a track list that contains all the interfaces in the track list using the Boolean OR to trigger when all objects fail.

```
switch(config)# track 44 list boolean OR
switch(config-track)# object 23
switch(config-track)# object 35
switch(config-track)# object 55
switch(config-track)# end
```

3. Add this track object to the vPC domain:

```
switch(config)# vpc domain 1
switch(config-vpc-domain)# track 44
```

4. Display the track object:

```
switch# show vpc brief
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 1
Peer status : peer adjacency formed ok
vPC keep-alive status : peer is alive
Configuration consistency status: success
vPC role : secondary
Number of vPCs configured : 52
Track object : 44
```

```
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ --------------------------------------------------
1 Po100 up 1-5,140
vPC status
----------------------------------------------------------------------
id Port Status Consistency Reason Active vlans
-- ---- ------ ----------- -------------------------- ------------
1 Po1 up success success 1-5,140
```

This example shows how to display information about the track objects:

```
switch# show track brief
Track Type Instance Parameter State Last
Change
23 Interface Ethernet8/33 Line Protocol UP 00:03:05
35 Interface Ethernet8/35 Line Protocol UP 00:03:15
44 List ----- Boolean
or UP 00:01:19
55 Interface port-channel100 Line Protocol UP 00:00:34
```

vPC Interactions with Other Features

vPC and LACP

LACP uses the system MAC address of the vPC domain to form the LACP Aggregation Group (LAG) ID for the vPC. (See the “Configuring Port Channels” chapter for information about LAG-ID and LACP.)

You can use LACP on all the vPC port channels, including those channels from the downstream device. We recommend that you configure LACP with active mode on the interfaces on each port channel on the vPC peer devices. This configuration allows you to more easily detect compatibility between devices, unidirectional links, and multihop connection, and provides dynamic reaction to run-time changes and link failures.

We recommend that you manually configure the system priority on the vPC Peer-Link devices to ensure that the vPC Peer-Link devices have a higher LACP priority than the downstream connected devices. A lower numerical value system priority means a higher LACP priority.

Note When manually configuring the system priority, you must ensure that you assign the same priority value on both vPC peer devices. If the vPC peer devices have different system priority values, vPC does not come up.

vPC Peer-Links and STP

Although vPCs provide a loop-free Layer 2 topology, STP is still required to provide a fail-safe mechanism to protect against any incorrect or defective cabling or possible misconfiguration. When you first bring up a vPC, STP reconverges. STP treats the vPC Peer-Link as a special link and always includes the vPC Peer-Link in the STP active topology.

We recommend that you set all the vPC Peer-Link interfaces to the STP network port type so that Bridge Assurance is automatically enabled on all vPC Peer-Links. We also recommend that you do not enable any

of the STP enhancement features on vPC Peer-Links. If the STP enhancements are already configured, they do not cause any problems for the vPC Peer-Links..

When you are running both MST and Rapid PVST+, ensure that the PVST simulation feature is correctly configured.

See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for information about STP enhancement features and PVST simulation.

Note You must configure a list of parameters to be identical on the vPC peer devices on both sides of the vPC Peer-Link. See the “Compatibility Parameters for vPC Interfaces” section for information about these required matched settings.

STP is distributed; that is, the protocol continues running on both vPC peer devices. However, the configuration on the vPC peer device elected as the primary device controls the STP process for the vPC interfaces on the secondary vPC peer device.

The primary vPC device synchronizes the STP state on the vPC secondary peer device using Cisco Fabric Services over Ethernet (CFSoE). See the “vPC and Orphan Ports” section for information about CFSoE.

The STP process for vPC also relies on the periodic keepalive messages to determine when one of the connected devices on the vPC Peer-Link fails. See the “Peer-Keepalive Link and Messages” section for information about these messages.

The vPC manager performs a proposal/handshake agreement between the vPC peer devices that set the primary and secondary devices and coordinates the two devices for STP. The primary vPC peer device then controls the STP protocol on both the primary and secondary devices. We recommend that you configure the primary vPC peer device as the STP primary root device and configure the secondary VPC device to be the STP secondary root device.

If the primary vPC peer device fails over to the secondary vPC peer device, there is no change in the STP topology.

The BPDUs uses the MAC address set for the vPC for the STP bridge ID in the designated bridge ID field. The vPC primary device sends these BPDUs on the vPC interfaces.

You must configure both ends of vPC Peer-Link with the identical STP configuration for the following parameters:

- STP global settings:
  - STP mode
  - STP region configuration for MST
  - Enable/disable state per VLAN
  - Bridge Assurance setting
  - Port type setting
  - Loop Guard settings
- STP interface settings:
  - Port type setting

- Loop Guard
- Root Guard

Note If any of these parameters are misconfigured, the Cisco NX-OS software suspends all interfaces in the vPC. Check the syslog and enter the **show vpc brief** command to see if the vPC interfaces are suspended.

Ensure that the following STP interface configurations are identical on both sides of the vPC Peer-Links or you may see unpredictable behavior in the traffic flow:

- BPDU Filter
- BPDU Guard
- Cost
- Link type
- Priority
- VLANs (PVRST+)

Note Display the configuration on both sides of the vPC Peer-Link to ensure that the settings are identical.

You can use the **show spanning-tree** command to display information about the vPC when that feature is enabled. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for an example.

Note We recommend that you configure the ports on the downstream devices as STP edge ports. You should configure all host ports connected to a switch as STP edge ports. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for more information about STP port types.

vPC Peer Switch

The vPC peer switch feature was added to Cisco NX-OS to address performance concerns around STP convergence. This feature allows a pair of Cisco Nexus 9000 Series devices to appear as a single STP root in the Layer 2 topology. This feature eliminates the need to pin the STP root to the vPC primary switch and improves vPC convergence if the vPC primary switch fails.

To avoid loops, the vPC Peer-Link is excluded from the STP computation. In vPC peer switch mode, STP BPDUs are sent from both vPC peer devices to avoid issues related to STP BPDU timeout on the downstream switches, which can cause traffic disruption.

This feature can be used with the pure peer switch topology in which the devices all belong to the vPC.

Note Peer-switch feature is supported on networks that use vPC and STP-based redundancy is not supported. If the vPC Peer-Link fail in a hybrid peer-switch configuration, you can lose traffic. In this scenario, the vPC peers use the same STP root ID as well as the same bridge ID. The access switch traffic is split in two with half going to the first vPC peer and the other half to the second vPC peer. With vPC Peer-Link failure, there is no impact to the north/south traffic but the east/west traffic is lost.

See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for information about STP enhancement features and Rapid PVST+.

vPC and ARP or ND

A feature was added to Cisco NX-OS to address table synchronization across vPC peers using the reliable transport mechanism of the Cisco Fabric Service over Ethernet (CFSoE) protocol. You must enable the **ip arp synchronize** and **ipv6 nd synchronize** commands to support faster convergence of address tables between the vPC peers. This convergence overcomes the delay that occurs in ARP table restoration for IPv4 or ND table restoration for IPv6 when the vPC Peer-Link port channel flaps or when a vPC peer comes back online.

vPC Multicast—PIM, IGMP, and IGMP Snooping

The Cisco NX-OS software for the Nexus 9000 Series devices supports the following on a vPC:

- PIM Any Source Multicast (ASM).
- PIM Source-Specific Multicast (SSM) .

Note The Cisco NX-OS software does not support Bidirectional (BIDR) on a vPC.

The software keeps the multicast forwarding state synchronized on both of the vPC peer devices. The IGMP snooping process on a vPC peer device shares the learned group information with the other vPC peer device through the vPC Peer-Link; the multicast states are always synchronized on both vPC peer devices. The PIM process in vPC mode ensures that only one of the vPC peer devices forwards the multicast traffic to the receivers.

Each vPC peer is a Layer 2 or Layer 3 device. Multicast traffic flows from only one of the vPC peer devices. You might see duplicate packets in the following scenarios:

- Orphan hosts
- When the source and receivers are in the Layer 2 vPC cloud in different VLANs with multicast routing enabled and a vPC member link goes down.

You might see negligible traffic loss in the following scenarios:

- When you reload the vPC peer device that is forwarding the traffic.
- When you restart PIM on the vPC peer device that is forwarding the traffic.

Overall multicast convergence times are scale and vPC role change / PIM restart duration dependent.

Ensure that you dual-attach all Layer 3 devices to both vPC peer devices. If one vPC peer device goes down, the other vPC peer device continues to forward all multicast traffic normally.

The following outlines vPC PIM and vPC IGMP/IGMP snooping:

- vPC PIM—The PIM process in vPC mode ensures that only one vPC peer device forwards multicast traffic. The PIM process in vPC mode synchronizes the source state with both vPC peer devices and elects which vPC peer device forwards the traffic.
- vPC IGMP/IGMP snooping—The IGMP process in vPC mode synchronizes the designated router (DR) information on both vPC peer devices. Dual DRs are available for IGMP when you are in vPC mode. Dual DRs are not available when you are not in vPC mode, because both vPC peer devices maintain the multicast group information between the peers.

Note A PIM adjacency between a Switched Virtual Interface (SVI) on a vPC VLAN (a VLAN that is carried on a vPC Peer-Link) and a downstream device is not supported; this configuration can result in dropped multicast packets. If a PIM neighbor relationship is required with a downstream device, a physical Layer 3 interface must be used on the Nexus switches instead of a vPC SVI.

For SVIs on vPC VLANs, only one PIM adjacency is supported, which is with the vPC peer switch. PIM adjacencies over the vPC Peer-Link with devices other than the vPC peer switch for the vPC-SVI are not supported.

You should enable or disable IGMP snooping identically on both vPC peer devices, and all the feature configurations should be identical. IGMP snooping is on by default.

Note The following commands are not supported in vPC mode:

- **ip pim spt-threshold infinity**
- **ip pim use-shared-tree-only**

See the *Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide* for more information about multicasting.

Multicast PIM Dual DR (Proxy DR)

By default, a multicast router sends PIM joins upstream only if it has interested receivers. These interested receivers can either be IGMP hosts (they communicate through IGMP reports) or other multicast routers (they communicate through PIM joins).

In the Cisco NX-OS vPC implementation, PIM works in dual designated router (DR) mode. That is, if a vPC device is a DR on a vPC SVI outgoing interface (OIF), its peer automatically assumes the proxy DR role. IGMP adds an OIF (the report is learned on that OIF) to the forwarding if the OIF is a DR. With dual DRs, both vPC devices have an identical (*,G) entry with respect to the vPC SVI OIFs as shown in this example:

```
VPC Device1:
------------
(*,G)
oif1 (igmp)
VPC Device2:
------------
(*,G)
```

```
oif1 (igmp)
```

IP PIM PRE-BUILD SPT

When the multicast source is in a Layer 3 cloud (outside the vPC domain), one vPC peer is elected as the forwarder for the source. This forwarder election is based on the metrics to reach the source. If there is a tie, the vPC primary is chosen as the forwarder. Only the forwarder has the vPC OIFs in its associated (S,G) and the nonforwarder (S,G) has 0 OIFs. Therefore, only the forwarder sends PIM (S,G) joins toward the source as shown in this example:

```
VPC Device1 (say this is Forwarder for Source 'S'):
------------
(*,G)
oif1 (igmp)
(S,G)
oif1 (mrib)
VPC Device2:
------------
(*,G)
oif1 (igmp)
(S,G)
NULL
```

In the case of a failure (for example, a Layer 3 Reverse Path Forwarding (RPF) link on the forwarder becomes inoperable or the forwarder gets reloaded), if the current nonforwarder ends up becoming the forwarder, it has to start sending PIM joins for (S,G) toward the source to pull the traffic. Depending upon the number of hops to reach the source, this operation might take some time (PIM is a hop-by-hop protocol).

To eliminate this issue and get better convergence, use the **ip pim pre-build-spt** command. This command enables PIM send joins even if the multicast route has 0 OIFs. In a vPC device, the nonforwarder sends PIM (S,G) joins upstream toward the source. The downside is that the link bandwidth upstream from the nonforwarder gets used for the traffic that is ultimately dropped by it. The benefits that result with better convergence far outweigh the link bandwidth usage. Therefore, we recommend that you use this command if you use vPCs.

vPC Peer-Links and Routing

The First Hop Redundancy Protocols (FHRPs) interoperate with vPCs. The Hot Standby Routing Protocol (HSRP), and Virtual Router Redundancy Protocol (VRRP) all interoperate with vPCs. We recommend that you dual-attach all Layer 3 devices to both vPC peer devices.

The primary FHRP device responds to ARP requests, even though the secondary vPC device forwards the data traffic.

To simplify initial configuration verification and vPC/HSRP troubleshooting, you can configure the primary vPC peer device with the FHRP active router highest priority.

In addition, you can use the priority command in the if-hsrp configuration mode to configure failover thresholds for when a group state enabled on a vPC Peer-Link is in standby or in listen state. You can configure lower and upper thresholds to prevent the interface from going up and down.

VRRP acts similarly to HSRP when running on vPC peer devices. You should configure VRRP the same way that you configure HSRP.

When the primary vPC peer device fails over to the secondary vPC peer device, the FHRP traffic continues to flow seamlessly.

We recommend that you configure routing adjacency between the two vPC peer devices to act as a backup routing path. If one vPC peer device loses Layer 3 uplinks, the vPC can redirect the routed traffic to the other vPC peer device and leverage its active Layer 3 uplinks.

You can configure the inter-switch link for a backup routing path in the following ways:

- Create a Layer 3 link between the two vPC peer devices.
- Use the non-VPC VLAN trunk with a dedicated VLAN interface.
- Use a vPC Peer-Link with a dedicated VLAN interface.

We do not recommend that you configure the burnt-in MAC address option (use-bia) for HSRP or manually configure virtual MAC addresses for any FHRP protocol in a vPC environment because these configurations can adversely affect vPC load balancing. The HSRP use-bia option is not supported on vPCs. When you are configuring custom MAC addresses, you must configure the same MAC address on both vPC peer devices.

You can use the **delay restore** command to configure a restore timer that delays the vPC coming back up until after the peer adjacency forms and the VLAN interfaces are back up. This feature enables you to avoid packet drops when the routing tables might not be converged before the vPC is once again passing traffic. Use the **delay restore** command to configure this feature.

To delay the VLAN interfaces on the restored vPC peer device from coming up, use the **interfaces-vlan** option of the **delay restore** command.

See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about FHRPs and routing.

CFSoE

The Cisco Fabric Services over Ethernet (CFSoE) is a reliable state transport mechanism that is used to synchronize the actions of the vPC peer devices. CFSoE carries messages and packets for many features linked with vPC, such as STP and IGMP. Information is carried in CFS/CFSoE protocol data units (PDUs).

When you enable the vPC feature, the device automatically enables CFSoE, and you do not have to configure anything. CFSoE distributions for vPCs do not need the capabilities to distribute over IP or the CFS regions. You do not need to configure anything for the CFSoE feature to work correctly on vPCs.

The CFSoE transport is local to each VDC.

You can use the **show mac address-table** command to display the MAC addresses that CFSoE synchronizes for the vPC Peer-Link.

Note Do not enter the **no cfs eth distribute** or the **no cfs distribute** command. You must enable CFSoE for vPC functionality. If you do enter either of these commands with vPC enabled, the system displays an error message.

When you enter the **show cfs application** command, the output displays “Physical-eth,” which shows the applications that are using CFSoE.

CFS also transports data over TCP/IP. See the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide for more information about CFS over IP.

Note The software does not support CFS regions.

vPC and Orphan Ports

When a device that is not vPC-capable connects to each peer, the connected ports are known as orphan ports because they are not members of a vPC. The device's link to one peer will be active (forwarding) and the other link will be standby (blocking) due to STP.

If a vPC Peer-Link failure or restoration occurs, an orphan port's connectivity might be bound to the vPC failure or restoration process. For example, if a device's active orphan port connects to the secondary vPC peer, the device loses any connections through the primary peer if a vPC Peer-Link failure occurs and the vPC ports are suspended by the secondary peer. If the secondary peer were to also suspend the active orphan port, the device's standby port becomes active, provides a connection to the primary peer, and restores connectivity. You can configure in the CLI that specific orphan ports are suspended by the secondary peer when it suspends its vPC ports and are restored when the vPC is restored.

Virtualization Support

All ports in a given vPC must be in the same VDC. This version of the software supports only one vPC domain per VDC. You can use the numbers from 1 to 4096 in each VDC to number the vPC.

vPC Recovery After an Outage

In a data center outage, both the vPC peer in vPC domain get reloaded. Occasionally only one peer can be restored. With no functioning peer-keepalive or vPC Peer-Link, the vPC cannot function normally, a method might be available to allow vPC services to use only the local ports of the functional peer.

Autorecovery

You can configure the Cisco Nexus 9000 Series device to restore vPC services when its peer fails to come online by using the **auto-recovery** command. You must save this setting in the startup configuration. On reload, if the vPC Peer-Link is down and three consecutive peer-keepalive messages are lost, the secondary device assumes the primary STP role and the primary LACP role. The software reinitializes the vPCs, bringing up its local ports. Because there are no peers, the consistency check is bypassed for the local vPC ports. The device elects itself to be the STP primary regardless of its role priority and also acts as the primary device for LACP port roles.

vPC Peer Roles After a Recovery

When the other peer device completes its reload and adjacency forms, the following process occurs:

1. The first vPC peer maintains its current role to avoid any transition reset to other protocols. The peer accepts the other available role.
2. When an adjacency forms, consistency checks are performed and appropriate actions are taken.

High Availability

During an In-Service Software Upgrade (ISSU), the software reload process on the first vPC device locks its vPC peer device by using CFS messaging over the vPC communications channel. Only one device at a time is upgraded. When the first device completes its upgrade, it unlocks its peer device. The second device then performs the upgrade process, locking the first device as it does so. During the upgrade, the two vPC devices

temporarily run different releases of Cisco NX-OS, however the system functions correctly because of its backward compatibility support.

Note See the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide for complete information about high-availability features.

vPC Forklift Upgrade Scenario

The following procedure describes a scenario of migrating pair of Cisco Nexus 9500 switches in a vPC domain to a different pair of Cisco Nexus 9500 switches with a same type of line cards. Migrating from Cisco Nexus 9504 switches to Cisco Nexus 9508 switches for the need of more interfaces is a typical example of such migration. The following migration scenarios are not supported:

- Migration of Cisco Nexus 9500 switches with a different set of line cards. For example, from a Cisco Nexus 9500 switches with N9K-X94xx line card to Cisco Nexus 9500 switches with N9K-X97xx line card.
- Migration between different generations of Cisco Nexus 9300 switches. For example, migration from Cisco Nexus N9K-C9372PX to Cisco Nexus N9K-93180YC-EX switches
- Having different generations of Cisco Nexus 9000 switches in a vPC domain is not supported

Considerations for a vPC forklift upgrade:

- vPC Role Election and Sticky-bit

  When the two vPC systems are joined to form a vPC domain, priority decides which device is the vPC primary and which is the vPC secondary. When the primary device is reloaded, the system comes back online and connectivity to the vPC secondary device (now the operational primary) is restored. The operational role of the secondary device (operational primary) does not change (to avoid unnecessary disruptions). This behavior is achieved with a sticky-bit, where the sticky information is not saved in the startup configuration. This method makes the device that is up and running win over the reloaded device. Hence, the vPC primary becomes the vPC operational secondary. Sticky-bit is also set when a vPC node comes up with vPC Peer-Link and peer-keepalive down and it becomes primary after the auto recovery period.

- vPC Delay Restore

  The delay restore timer is used to delay the vPC from coming up on the restored vPC peer device after a reload when the peer adjacency is already established.

  To delay the VLAN interfaces on the restored vPC peer device from coming up, use the **interfaces-vlan** option of the **delay restore** command.

- vPC Auto-Recovery

  During a data center power outage when both vPC peer switches go down, if only one switch is restored, the auto-recovery feature allows that switch to assume the role of the primary switch and the vPC links come up after the auto-recovery time period. The default auto-recovery period is 240 seconds.

The following example is a migration scenario that replaces vPC peer nodes Node1 and Node2 with New_Node1 and New_Node2.

	Migration Step	Expected Behavior	Node1 Configured role (Ex: role priority 100)	Node1 Operational role	Node2 Configured role (Ex: role priority 200)	Node2 Operational role
1	Initial state	Traffic is forwarded by both vPC peers – Node1 and Node2. Node1 is primary and Node2 is secondary.	primary	Primary Sticky bit: False	secondary	Secondary Sticky bit: False
2	Node2 replacement – Shut all vPCs and uplinks on Node2. vPC Peer-Link and vPC peer-keepalive are in administrative up state.	Traffic converged on Primary vPC peer Node1.	primary	Primary Sticky bit: False	secondary	Secondary Sticky bit: False
3	Remove Node2.	Node1 will continue to forward traffic.	primary	Primary Sticky bit: False	n/a	n/a
4	Configure New_Node2. Copy the configuration to startup config. vPC vPC Peer-Link and peer-keepalive in administrative up state. Power off New_Node2. Make all connections. Power on New_Node2.	New_Node2 will come up as secondary. Node1 continue to be primary. Traffic will continue to be forwarded on Node01.	primary	Primary Sticky bit: False	secondary	Secondary Sticky bit: False
5	Bring up all vPCs and uplink ports on New_Node2.	Traffic will be forwarded by both Node 1 and New_Node2.	primary	Primary Sticky bit: False	secondary	Secondary Sticky bit: False
6	Node1 replacement - Shut vPCs and uplinks on Node1.	Traffic will converge on New_Node2.	primary	Primary Sticky bit: False	secondary	Secondary Sticky bit: False

	Migration Step	Expected Behavior	Node1 Configured role (Ex: role priority 100)	Node1 Operational role	Node2 Configured role (Ex: role priority 200)	Node2 Operational role
7	Remove Node1.	New_Node2 will become secondary, operational primary and sticky bit will be set to True.	n/a	n/a	secondary	Primary Sticky bit: True
8	Configure New_Node1. Copy running to startup. Power off the new Node1. Make all connections. Power on New_Node1.	New_Node1 will come up as primary, operational secondary.	primary	Secondary Sticky bit: False	secondary	Primary Sticky bit: True
9	Bring up all vPCs and uplink ports on New_Node1.	Traffic will be forwarded by both New Node1 and new Node2.	primary	Secondary Sticky bit: False	secondary	Primary Sticky bit: True

Note If you prefer to have the configured secondary node as the operational secondary and the configured primary as the operational primary, then Node2 can be reloaded at the end of the migration. This is optional and does not have any functional impact.

Guidelines and Limitations

vPCs have the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- Cisco Nexus 9000 Series switches do not support NAT on vPC topology.
- The **spanning-tree pseudo-information** command is not available on Cisco Nexus 92160 and Cisco Nexus 93180 switches starting from Release 7.0(3)I4(1).
- vPC peers must run the same Cisco NX-OS release. During a software upgrade, make sure to upgrade the primary vPC peer first.
- All ports for a given vPC must be in the same VDC.
- You must enable vPCs before you can configure them.
- You must configure the peer-keepalive link and messages before the system can form the vPC Peer-Link.

- Only Layer 2 port channels can be in vPCs.
- You must configure both vPC peer devices; the configuration is not sent from one device to the other.
- To configure multilayer (back-to-back) vPCs, you must assign unique vPC domain ID for each respective vPC.
- Check that the necessary configuration parameters are compatible on both sides of the vPC Peer-Link. See the “Compatibility Parameters for vPC Interfaces” section for information about compatibility recommendations.
- You might experience minimal traffic disruption while configuring vPCs.
- The software does not support BIDR PIM on vPCs.
- The software does not support DHCP snooping, DAI, or IPSG in a vPC environment; DHCP Relay is supported.
- The software does not support CFS regions.
- Port security is not supported on port channels.
- When **peer-switch** features are configured under **vpc domain** configuration mode on two Cisco Nexus 9000 Series switches, the spanning-tree root changes even for VLANs that are not enabled on the vPC Peer-Link. Both the switches act as one system with one MAC address as the bridge address. This is true even for non-vPC mst-instance or VLANs. Therefore, a non vPC Peer-Link between the two switches gets blocked as a backup link. This is an expected behavior.
- We recommend that you configure all the port channels in the vPC using LACP with the interfaces in active mode.
- Back-to-back, multilayer vPC topologies require unique domain IDs on each respective vPC.
- Having the same Hot Standby Router Protocol (HSRP)/Virtual Router Redundancy Protocol (VRRP) group on all nodes on a double sided vPC is supported on Cisco NX-OS 7.0(3)I2(1) and later releases.
- When migrating from a pair of spine nodes to a pair of Cisco Nexus 9000 devices, the HSRP priority should be configured so that the Cisco Nexus 9000 vPC peers are in Active/Standby state. There is no support for Cisco Nexus 9000 vPC peers in HSRP state to be in Active/Listen state, or Standby/Listen state. (7.(0)I2(2) or later)
- When using vPCs, we recommend that you use default timers for FHRP (HSRP, VRRP), and PIM configurations. There is no advantage in convergence times when using aggressive timers in vPC configurations.
- If you configure open shortest path first (OSPF) in a vPC environment, use the following timer commands in router configuration mode on the core switch to ensure fast OSPF convergence when a vPC Peer-Link is shut down:

```
switch (config-router)# timers throttle spf 1 50 50
switch (config-router)# timers lsa-arrival 10
```

  See the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* for further details about OSPF.
- BFD for VRRP/HSRP is not supported in a vPC environment.

- The STP port cost is fixed to 200 in a vPC environment.
- Jumbo frames are enabled by default on the vPC Peer-Link.
- To accommodate increased traffic when the vPC goes down and traffic needs to cross the vPC Peer-Link, it is a best practice to use multiple high bandwidth interfaces (such as the 40G interfaces for the Cisco Nexus 9000) across linecards for the vPC Peer-Link.
- The **vpc orphan-ports suspend** command also applies to ports in non-vPC VLANs and Layer 3 ports. However, it is recommended to be used with ports in VPC VLANs.
- Starting in NX-OS 7.0(3)I5(2), FEX-AA (dual-homed FEX) and FEX-ST (FEX straight-thru) topologies (FEX-AA and FEX-ST) are supported. The following mixing is not supported as the parent switches:
  - Cisco Nexus 9300-EX and 9300 switches
  - Cisco Nexus 9300 and 9500 switches
  - Cisco Nexus 9300-EX and 9500 switches
- vPC STP hitless role change feature is supported only from Cisco Nexus 9000 Release 7.0(3)I7(1) onwards.
- vPC role change can be performed from either of the peer devices.
- When forming a vPC domain between two Cisco Nexus 9300 Series switches, both switches must be the exact same model to form a supported vPC domain. When forming a vPC domain between two Cisco Nexus 9500 Series switches, both switches must consist of the same models of line cards, fabric modules, supervisor modules, and system controllers inserted in the same slots of the chassis to form a supported vPC domain.
- If the original secondary device has higher role priority value than the original primary device, role swapping cannot be performed. Change the role priority on either vPC device so that the value of the original secondary device is lower than the original primary one. To view the existing role of a device, use the show vpc role command on local and peer switch.
- Always check the existing configured role priority before configuring vPC hitless role change feature
- In a vPC domain, enable the peer-switch command, where both vPC peers have same STP priorities, and ensure it is operational before issuing a role change. If you do not enable the peer-switch command, it can lead to convergence issues. Use **show spanning-tree summary | grep peer** command to verify whether the peer vPC switch is operational or not.
- All the devices that are attached to a vPC domain must be dual homed.
- The first generation Broadcom based Nexus 9300 series switches and Nexus 9500 series line-cards does not support policy based routing route map with a set ip next-hop statement where the egress interface is the vPC Peer-Link while the vPC convergence TCAM region is allocated. This limitation does not apply to cloud scale based Nexus 9000 series devices such as Cisco Nexus 9200 switches, 9300 switches with EX/FX/FX2 line-cards and Nexus 9500 platform switches with 9700-EX/FX line-cards.
- Beginning with Cisco NX-OS Release 7.0(3)I5(1), Layer 3 over vPC is supported on Cisco Nexus 9000 Series switches for Layer 3 unicast communication only. Layer 3 over vPC is not supported for Layer 3 multicast traffic. For more information see the *Best Practices for Layer 3 and vPC Configuration* section

- The default behavior with Layer 3 peer-router and TTL=1 packet destined to IP of vPC peer is to punt packet to CPU and then forward the software to vPC peer. This is applicable to the Cloud Scale based EOR switches.
- Starting with Cisco NX-OS Release 7.0(3)I7(9) Cloud Scale based TOR switches can forward TTL=1 packet destined to vPC peer in hardware/data plane. It is recommended to use one of these releases or later releases for a seamless operation of the feature.

Best Practices for Layer 3 and vPC Configuration

This section describes best practices for using and configuring Layer 3 with vPC.

Layer 3 and vPC Configuration Overview

When a Layer 3 device is connected to a vPC domain through a vPC, it has the following views:

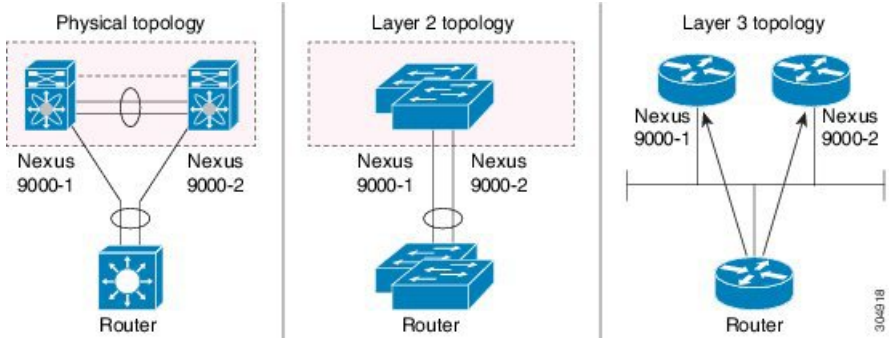
- At Layer 2, the Layer 3 device sees a unique Layer 2 switch presented by the vPC peer devices.
- At Layer 3, the Layer 3 device sees two distinct Layer 3 devices (one for each vPC peer device).

vPC is a Layer 2 virtualization technology, so at Layer 2, both vPC peer devices present themselves as a unique logical device to the rest of the network.

There is no virtualization technology at Layer 3, so each vPC peer device is seen as a distinct Layer 3 device by the rest of the network.

The following figure illustrates the two different Layer 2 and Layer 3 views with vPC.

Figure 16: Different Views for vPC Peer Devices

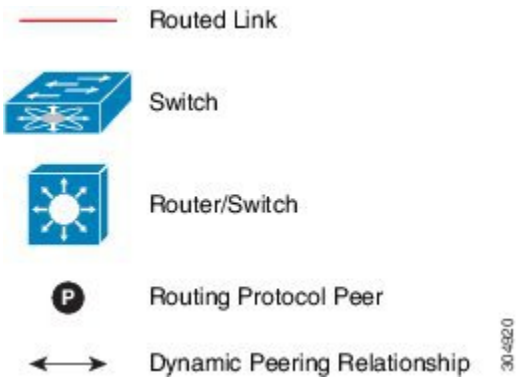


Supported Topologies for Layer 3 and vPC

This section contains examples of Layer 3 and vPC network topologies.

There are two approaches for Layer 3 and vPC interactions. The first one is by using dedicated Layer 3 links to connect the Layer 3 devices to each vPC peer device. The second one is by allowing the Layer 3 devices to peer with the SVIs defined on each of the vPC peer device, on a dedicated VLAN that is carried on the vPC connection. The following sections describe all the supported topologies leveraging the elements that are described in the legends in the following figure.

Figure 17: Legend

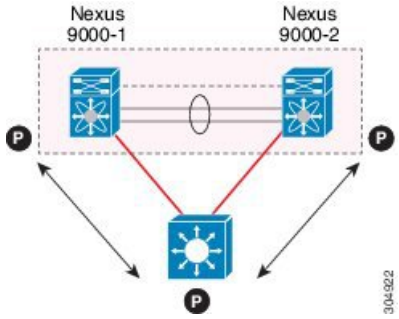


Peering with an External Router Using Layer 3 Links

This example shows a topology that uses Layer 3 links to connect a Layer 3 device to the Cisco Nexus 9000 switches that are part of the a vPC domain

Note Interconnecting the two entities together in this way allows to support Layer 3 unicast and multicast communication.

Figure 18: Peering with an External Router Using Layer 3 Links



Layer 3 devices can initiate Layer 3 routing protocol adjacencies with both vPC peer devices.

One or multiple Layer 3 links can be used to connect a Layer 3 device to each vPC peer device. Cisco Nexus 9000 series devices support Layer 3 Equal Cost Multipathing (ECMP) with up to 16 hardware load-sharing paths per prefix. Traffic from a vPC peer device to a Layer 3 device can be load-balanced across all the Layer 3 links interconnecting the two devices together.

Using Layer 3 ECMP on the Layer 3 device can effectively use all Layer 3 links from the device to the vPC domain. Traffic from a Layer 3 device to the vPC domain can be load-balanced across all the Layer 3 links interconnecting the two entities together.

Follow these guidelines when connecting a Layer 3 device to the vPC domain using Layer 3 links:

- Use separate Layer 3 links to connect Layer 3 devices to the vPCdomain. Each link represents a point-to-point Layer 3 connection and should get assigned an IP address taken from a small IP subnet (/30 or /31).

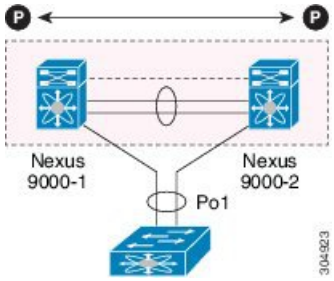
- If the Layer 3 peering is required for multiple VRFs, it is recommended to define multiple sub-interfaces, each mapped to an individual VRF.

Peering Between vPC Devices for a Backup Routing Path

This example shows peering between the two vPC peer devices with a Layer 3 backup routed path. If the Layer 3 uplinks on vPC peer device 1 or vPC peer device 2 fail, the path between the two peer devices is used to redirect traffic to the switch that has the Layer 3 uplinks in the up state.

The Layer 3 backup routing path can be implemented using a dedicated interface VLAN (such as SVI) over the vPC Peer-Link or by using dedicated Layer 2 or Layer 3 links across the two vPC peer devices.

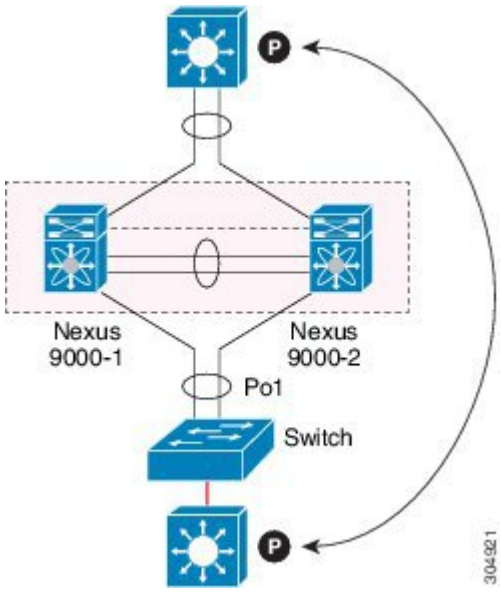
Figure 19: Peering Between vPC Devices for a Backup Routing Path



Direct Layer 3 Peering Between Routers

In this scenario, the Nexus 9000 devices part of the vPC domain are simply used as a Layer 2 transit path to allow the routers connected to them to establish Layer 3 peering and communication.

Figure 20: Peering Between Routers



The Layer 3 devices can peer with each other in following two methods. Peering also depends on the specific device deployed for this role.

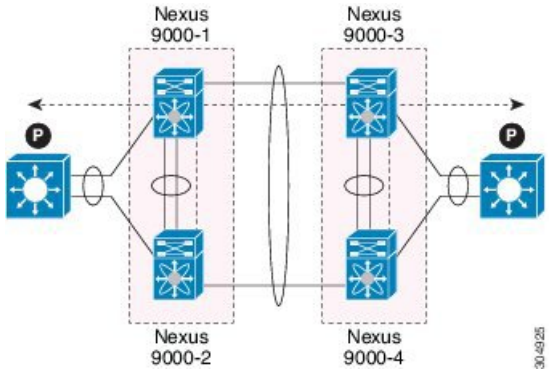
- Defining a VLAN network interface (SVI) for a VLAN that is extended between the Layer 3 devices through the intermediate Cisco Nexus 9000 vPC peer switches.
- Defining a Layer 3 port-channel interface on each Layer 3 device and establishing a point-to-point Layer 3 peering.

Note In deployments where the Layer 3 peering must be established for multiple VRFs, the first method require the definition on the Layer 3 devices of a VLAN (and SVI) per VRF. For the second method, it is possible to create a Layer 3 port-channel subinterface per VRF

Peering Between Two Routers with vPC Devices as Transit Switches

This example is similar to the peering between routers topology. In this case also, the Cisco Nexus 9000 devices that are part of the same vPC domain are only used as Layer 2 transit paths. The difference here is that there are two pairs of Cisco Nexus 9000 switches. Each switch that is connected with a Layer 3 device using a vPC connection, also establishes a back-to-back vPC connection between them. The difference is that the vPC domains are only used as Layer 2 transit paths.

Figure 21: Peering Between Two Routers with vPC Devices as Transit Switches



This topology is commonly used when you want to establish connectivity between separate data centers that are interconnected with direct links (dark fibers or DWDM circuits). The two pairs of Cisco Nexus 9000 switches, in this case, provide only Layer 2 extension services, allowing the Layer 3 devices to peer with each other at Layer 3.

Peering with an External Router on Parallel Interconnected Routed Ports

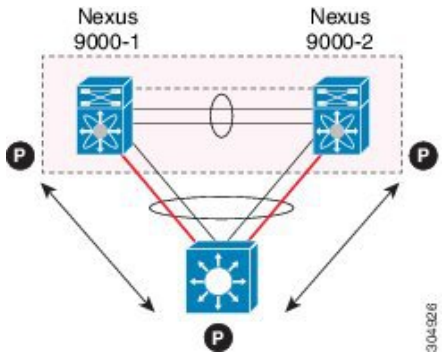
When you require both routed and bridged traffic, use individual Layer 3 links for routed traffic and a separate Layer 2 port-channel for bridged traffic, as shown in following figure.

The Layer 2 links are used for bridged traffic (traffic staying in the same VLAN) or inter-VLAN traffic (assuming vPC domain hosts the interface VLAN and associated HSRP configuration).

The Layer 3 links are used for routing protocol peering adjacency with each vPC peer device.

The purpose of this topology is to attract specific traffic to go through the Layer 3 device. Layer 3 links are also used to carry routed traffic from a Layer 3 device to the vPC domain.

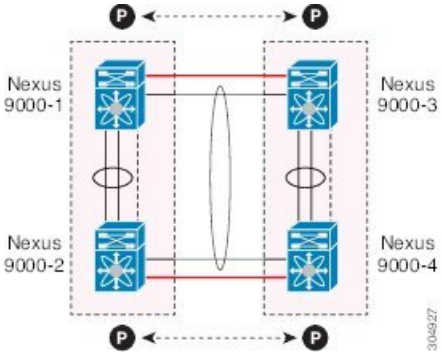
Figure 22: Peering with an External Router on Parallel Interconnected Routed Ports



Peering between vPC Switch Pairs on Parallel Interconnected Routed Ports

An alternative design to what is shown in the previous section (Peering Between Two Routers with vPC Devices as Transit Switches), uses two pairs of Cisco Nexus 9000 switches that are deployed in each data center for providing both Layer 2 and Layer 3 extension services. When routing protocol peering adjacency is required to be established between the two pairs of Cisco Nexus 9000 devices, the best practice is to add dedicated Layer 3 links between the two sites as shown in the following example.

Figure 23: Peering Over a vPC Interconnection on Parallel Interconnected Routed Ports



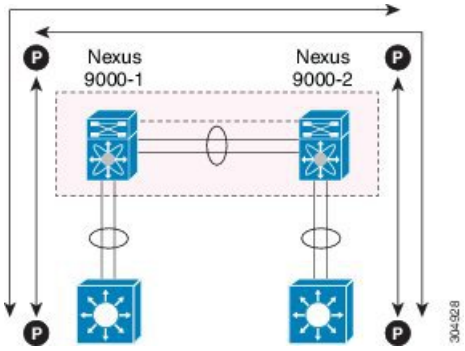
The back-to-back vPC connection between the two data centers carry bridged traffic or inter-VLAN traffic while the dedicated Layer 3 links carry the routed traffic across the two sites.

Peering Over a PC Interconnection and Dedicated Interswitch Link Using non-vPC VLAN

This example shows when the Layer 3 device is single-attached to the vPC domain, you can use a non-vPC VLAN with a dedicated inter-switch link to establish the routing protocol peering adjacency between the Layer 3 device and each vPC peer device. However, the non-vPC VLAN must be configured to use a static MAC that is different than the vPC VLAN.

Note Configuring the vPC VLAN (and vPC Peer-Link) for this purpose is not supported.

Figure 24: Peering Over a PC Interconnection and Dedicated Interswitch Link Using non-vPC VLAN

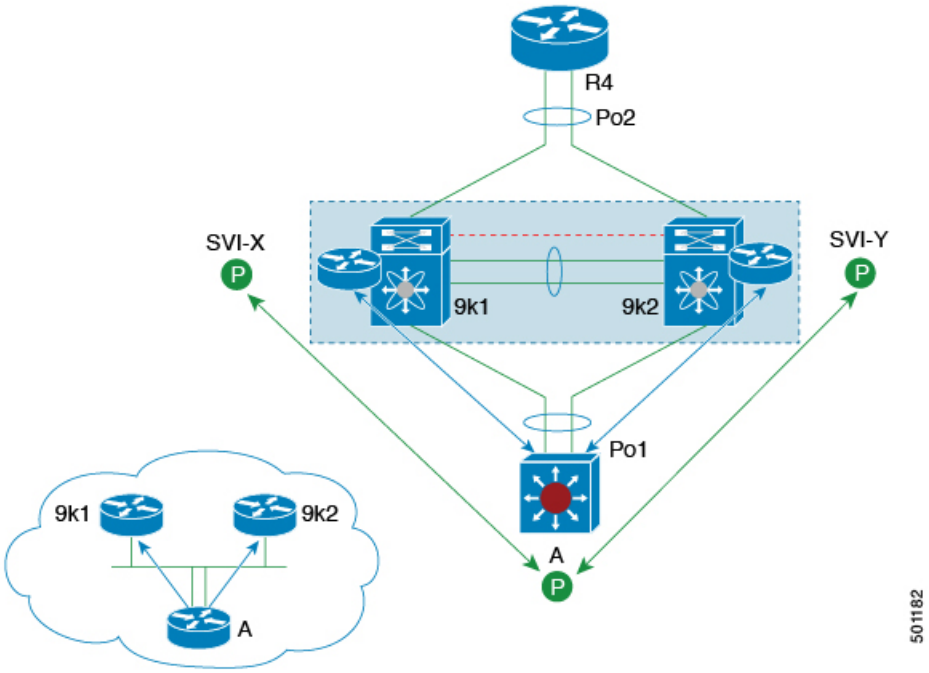


Peering Directly Over a vPC Connection

Beginning with Cisco NX-OS Release 7.0(3)I5(1), an alternative method has been introduced to establish Layer 3 peering between a Layer 3 router and a pair of Cisco Nexus 9000 vPC switches.

Note Peering directly over a vPC connection is supported only for Layer 3 unicast communication but not for Layer 3 multicast traffic. If you require Layer 3 multicast, you must establish peering over dedicated Layer 3 links

Figure 25: Supported: Peering Over a vPC Interconnection Where the Router Peers with Both the vPC Peers.

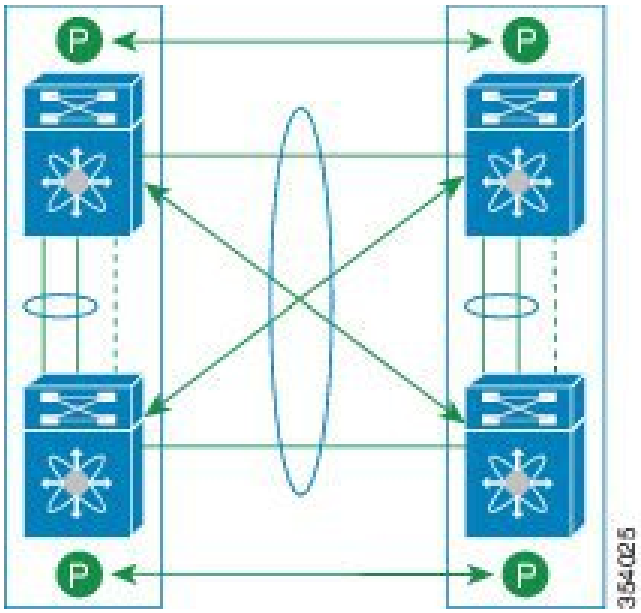


In this scenario, the Layer 3 peering between the external router and the Cisco Nexus 9000 switches that are part of a same vPC domain is established directly on a VLAN carried on the vPC connection. The external router in this case peers with SVI interfaces defined on each vPC device. As for the scenario shown in previous

figure 12, the external router could use an SVI or a Layer 3 Port-Channel to peer with the vPC devices (multiple SVIs or Port-Channel subinterfaces could be used for a multi-VRF deployment).

This deployment model requires configuring **layer3 peer-router** command as part of the vPC domain. You can adopt the same approach for establishing Layer 2 and Layer 3 connectivity on a vPC back-to-back connection established between two separate pairs of vPC switches.

Figure 26: Supported: Peering Over a vPC Interconnection Where Each Nexus Device Peers with Two vPC Peers.



In this deployment model, SVI interfaces in the same VLAN is configured on all the four Cisco Nexus 9000 switches to establish routing peering and connectivity between them.

Default Settings

The following table lists the default settings for vPC parameters.

Table 13: Default vPC Parameters

Parameters	Default
vPC system priority	32667
vPC peer-keepalive message	Disabled
vPC peer-keepalive interval	1 second
vPC peer-keepalive timeout	5 seconds
vPC peer-keepalive UDP port	3200

Configuring vPCs

Note You must use these procedures on both devices on both sides of the vPC Peer-Link. You configure both of the vPC peer devices using these procedures.

This section describes how to configure vPCs using the command-line interface (CLI).

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Enabling vPCs

You must enable the feature vPC before you can configure and use vPCs.

SUMMARY STEPS

1. **configure terminal**
2. **feature vpc**
3. **exit**
4. **show feature**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**feature vpc** **Example:** `switch(config)# feature vpc`	Enables vPCs on the device.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	**show feature** **Example:** `switch# show feature`	(Optional) Displays which features are enabled on the device.

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

Example

This example shows how to enable the vPC feature:

```
switch# configure terminal
switch(config)# feature vpc
switch(config)# exit
switch(config)#
```

Disabling vPCs

Note When you disable the vPC functionality, the device clears all the vPC configurations.

SUMMARY STEPS

1. **configure terminal**
2. **no feature vpc**
3. **exit**
4. **show feature**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**no feature vpc** **Example:** `switch(config)# no feature vpc`	Disables vPCs on the device.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

	Command or Action	Purpose
Step 4	**show feature** **Example:** `switch# show feature`	(Optional) Displays which features are enabled on the device.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to disable the vPC feature:

```
switch# configure terminal
switch(config)# no feature vpc
switch(config)# exit
switch#
```

Creating a vPC Domain and Entering vpc-domain Mode

You can create a vPC domain and put the vPC Peer-Link port channels into the identical vPC domain on both vPC peer devices. Use a unique vPC domain number throughout a single vPC domain . This domain ID is used to automatically to form the vPC system MAC address.

You can also use this command to enter vpc-domain command mode.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **exit**
4. **show vpc brief**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vpc domain** *domain-id*
	Example:
	`switch(config)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Creates a vPC domain on the device, and enters vpc-domain configuration mode for configuration purposes. There is no default; the range is from 1 to 1000.

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

Step 4	**show vpc brief** **Example:** `switch# show vpc brief`	(Optional) Displays brief information about each vPC domain.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to enter the vpc-domain command mode to configure an existing vPC domain:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# exit
switch(config)#
```

Configuring a vPC Keepalive Link and Messages

You can configure the destination IP for the peer-keepalive link that carries the keepalive messages. Optionally, you can configure other parameters for the keepalive messages.

Note You must configure the vPC peer-keepalive link before the system can form the vPC Peer-Link.

Note We recommend that you configure a separate VRF instance and put a Layer 3 port from each vPC peer device into that VRF for the vPC peer-keepalive link. Do not use the vPC Peer-Link itself to send vPC peer-keepalive messages. For information about creating and configuring VRFs, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide. Ensure that both the source and destination IP addresses use for the peer-keepalive message are unique in your network. The management port and management VRF are the defaults for these keepalive messages.

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **peer-keepalive destination** *ipaddress* [**hold-timeout** *secs* | **interval** *msecs* {**timeout** *secs*} | {**precedence** {*prec-value* | **network** | **internet** | **critical** | **flash-override** | **flash** | **immediate priority** | **routine**}} | **tos** {*tos-value* | **max-reliability** | **max-throughput** | **min-delay** | **min-monetary-cost** | **normal**}} |**tos-byte** *tos-byte-value*} | **source** *ipaddress* | **vrf** {*name* | **management vpc-keepalive**}]
4. **exit**
5. **show vpc statistics**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vpc domain** *domain-id*
	Example:
	`switch(config)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Creates a vPC domain on the device, and enters vpc-domain configuration mode.

Step 3	**peer-keepalive destination** *ipaddress* [**hold-timeout** *secs*	**interval** *msecs* {**timeout** *secs*}	{**precedence** {*prec-value*	**network**	**internet**	**critical**	**flash-override**	**flash**	**immediate priority**	**routine**}}	**tos** {*tos-value*	**max-reliability**	**max-throughput**	**min-delay**	**min-monetary-cost**	**normal**}} |**tos-byte** *tos-byte-value*}	**source** *ipaddress*	**vrf** {*name*	**management vpc-keepalive**}] **Example:** `switch(config-vpc-domain)# peer-keepalive destination 172.28.230.85` `switch(config-vpc-domain)#`	Configures the IPv4 and IPv6 addresses for the remote end of the vPC peer-keepalive link. **Note** The system does not form the vPC Peer-Link until you configure a vPC peer-keepalive link. **Note** You may get the following error message if you do not specify the source IP address when you configure an IPv6 address for the remote end of the vPC peer-keepalive link. `Cannot configure IPV6 peer-keepalive without source IPV6 address` The management ports and VRF are the defaults. **Note** We recommend that you configure a separate VRF and use a Layer 3 port from each vPC peer device in that VRF for the vPC peer-keepalive link. For more information about creating and configuring VRFs, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.
Step 4	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
Step 5	**show vpc statistics** **Example:** `switch# show vpc statistics`	(Optional) Displays information about the configuration for the keepalive messages.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

For more information about configuring VRFs, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

This example shows how to configure the destination and source IP address and VRF for the vPC-peer-keepalive link:

```
switch# configure terminal
switch(config)# vpc domain 100
switch(config-vpc-domain)# peer-keepalive destination 172.168.1.2 source 172.168.1.1 vrf
vpc-keepalive
switch(config-vpc-domain)# exit
switch#
```

Creating a vPC Peer-Link

You create the vPC Peer-Link by designating the port channel that you want on each device as the vPC Peer-Link for the specified vPC domain. We recommend that you configure the Layer 2 port channels that you are designating as the vPC Peer-Link in trunk mode and that you use two ports on separate modules on each vPC peer device for redundancy.

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel-number*
3. **switchport mode trunk**
4. **switchport trunk allowed vlan** *vlan-list*
5. **vpc peer-link**
6. **exit**
7. **show vpc brief**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 20`
	`switch(config-if)#`
		Selects the port channel that you want to use as the vPC Peer-Link for this device, and enters interface configuration mode.

Step 3	**switchport mode trunk** **Example:** `switch(config-if)# switchport mode trunk`	(Optional) Configures this interface in trunk mode.
Step 4	**switchport trunk allowed vlan** *vlan-list* **Example:** `switch(config-if)# switchport trunk allowed vlan 1-120,201-3967`	(Optional) Configures the permitted VLAN list.

Step 5	**vpc peer-link**
	Example:
	`switch(config-if)# vpc peer-link`
	`switch(config-vpc-domain)#`
		Configures the selected port channel as the vPC Peer-Link, and enters vpc-domain configuration mode.

Step 6	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

Step 7	**show vpc brief** **Example:** `switch# show vpc brief`	(Optional) Displays information about each vPC, including information about the vPC Peer-Link.
Step 8	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure a vPC Peer-Link:

```
switch# configure terminal
switch(config)# interface port-channel 20
switch(config-if)# switchport mode
switch(config-if)# switchport mode trunk
```

```
switch(config-if)# switchport trunk allowed vlan 1-120,201-3967
switch(config-if)# vpc peer-link
switch(config-vpc-domain)# exit
switch(config)#
```

Configuring a vPC Peer-Gateway

You can configure vPC peer devices to act as the gateway for packets that are destined to the vPC peer device's MAC address.

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **peer-gateway**
4. **exit**
5. **show vpc brief**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vpc domain** *domain-id*
	Example:
	`switch(config-if)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Creates a vPC domain if it does not already exist, and enters vpc-domain configuration mode.

Step 3	**peer-gateway** **Example:** `switch(config-vpc-domain)# peer-gateway` **Note** Disable IP redirects on all interface-vlans of this vPC domain for correct operation of this feature.	Enables Layer 3 forwarding for packets destined to the peer's gateway MAC address.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

	Command or Action	Purpose
Step 5	**show vpc brief** **Example:** `switch# show vpc brief`	(Optional) Displays information about each vPC, including information about the vPC Peer-Link.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring a Graceful Consistency Check

You can configure the graceful consistency check feature, which is enabled by default. Unless this feature is enabled, the vPC is completely suspended when a mismatch in a mandatory compatibility parameter is introduced in a working vPC. When this feature is enabled, only the links on the secondary peer device are suspended. See the "Compatibility Parameters for vPC Interfaces" section for information about consistent configurations on the vPCs.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **graceful consistency-check**
4. **exit**
5. **show vpc brief**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vpc domain** *domain-id*
	Example:
	`switch(config-if)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Creates a vPC domain if it does not already exist, and enters vpc-domain configuration mode.

Step 3	**graceful consistency-check** **Example:** `switch(config-vpc-domain)# graceful consistency-check`	Specifies that only the links on the secondary peer device are suspended when a mismatch is detected in a mandatory compatibility parameter. Use the **no** form of this command to disable the feature.
Step 4	**exit** **Example:**	Exits vpc-domain configuration mode.

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
Step 5	**show vpc brief** **Example:** `switch# show vpc brief`	(Optional) Displays information on the vPCs.

Example

This example shows how to enable the graceful consistency check feature:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# graceful consistency-check
switch(config-vpc-domain)# exit
switch(config)#
```

Checking the Configuration Compatibility on a vPC Peer-Link

After you have configured the vPC Peer-Link on both vPC peer devices, check that the configurations are consistent on all vPC interfaces. See the "Compatibility Parameters for vPC Interfaces" section for information about consistent configurations on the vPCs.

SUMMARY STEPS

1. **configure terminal**
2. **show vpc consistency-parameters** {**global** | **interface port-channel** *channel-number*}

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**show vpc consistency-parameters** {**global** | **interface port-channel** *channel-number*}
	Example:
	`switch(config)# show vpc consistency-parameters global`
	`switch(config)#`
		(Optional) Displays the status of those parameters that must be consistent across all vPC interfaces.

Example

This example shows how to check that the required configurations are compatible across all the vPC interfaces:

```
switch# configure terminal
switch(config)# show vpc consistency-parameters global
switch(config)#
```

Note Messages regarding the vPC interface configuration compatibility are also logged to the syslog.

Moving Other Port Channels into a vPC

We recommend that you attach the vPC domain downstream port channel to two devices for redundancy.

To connect to the downstream device, you create a port channel from the downstream device to the primary vPC peer device and you create another port channel from the downstream device to the secondary peer device. On each vPC peer device, you assign a vPC number to the port channel that connects to the downstream device. You will experience minimal traffic disruption when you are creating vPCs.

Before you begin

Ensure that you have enabled the vPC feature.

Ensure that you are using a Layer 2 port channel.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel-number*
3. **vpc** *number*
4. **exit**
5. **show vpc brief**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 20`
	`switch(config-if)#`
		Selects the port channel that you want to put into the vPC to connect to the downstream device, and enters interface configuration mode.

	Command or Action	Purpose

Step 3	**vpc** *number*
	Example:
	`switch(config-if)# vpc 5`
	`switch(config-vpc-domain)#`
		Configures the selected port channel into the vPC to connect to the downstream device. You can use any module in the device for these port channels. The range is from 1 and 4096.
		Note The vPC number that you assign to the port channel connecting to the downstream device from the vPC peer device must be identical on both vPC peer devices.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

Step 5	**show vpc brief** **Example:** `switch# show vpc brief`	(Optional) Displays information on the vPCs.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure a port channel to connect to the downstream device:

```
switch# configure terminal
switch(config)# interface port-channel 20
switch(config-if)# vpc 5
switch(config-if)# exit
switch(config)#
```

Manually Configuring a vPC Domain MAC Address

When you create a vPC domain, the Cisco NX-OS software automatically creates a vPC system MAC address, which is used for operations that are confined to the link-scope, such as LACP. However, you might choose to configure the vPC domain MAC address manually.

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **system-mac** *mac-address*

4. **exit**
5. **show vpc role**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vpc domain** *domain-id*
	Example:
	`switch(config)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Enters the vPC domain number that you want to configure. The system enters vpc-domain configuration mode.

Step 3	**system-mac** *mac-address*
	Example:
	`switch(config-vpc-domain)# system-mac 23fb.4ab5.4c4e`
	`switch(config-vpc-domain)#`
		Enters the MAC address that you want for the specified vPC domain in the following format: aaaa.bbbb.cccc.

Step 4	**exit**
	Example:
	`switch(config-vpc-domain)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

Step 5	**show vpc role** **Example:** `switch# show vpc brief`	(Optional) Displays the vPC system MAC address.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to manually configure a vPC domain MAC address:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# system-mac 13gb.4ab5.4c4e
switch(config-vpc-domain)# exit
switch(config)#
```

Manually Configuring the System Priority

When you create a vPC domain, the system automatically creates a vPC system priority. However, you can also manually configure a system priority for the vPC domain.

Note We recommend that you manually configure the vPC system priority when you are running LACP to ensure that the vPC peer devices are the primary devices on LACP. When you manually configure the system priority, ensure that you configure the same priority value on both vPC peer devices. If these values do not match, vPC does not come up.

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **system-priority** *priority*
4. **exit**
5. **show vpc role**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vpc domain** *domain-id*
	Example:
	`switch(config)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Enters the vPC domain number that you want to configure. The system enters vpc-domain configuration mode.

Step 3	**system-priority** *priority*
	Example:
	`switch(config-vpc-domain)# system-priority 4000`
	`switch(config-vpc-domain)#`
		Enters the system priority that you want for the specified vPC domain. The range of values is from 1 to 65535. The default value is 32667.

Step 4	**exit**
	Example:
	`switch(config-vpc-domain)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

	Command or Action	Purpose
Step 5	**show vpc role** **Example:** `switch# show vpc role`	(Optional) Displays the vPC system priority.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to manually configure the vPC domain system priority:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# system-priority 4000
switch(config-vpc-domain)# exit
switch(config)#
```

Manually Configuring the vPC Peer Device Role

By default, the Cisco NX-OS software elects a primary and secondary vPC peer device after you configure the vPC domain and both sides of the vPC Peer-Link. However, you might want to elect a specific vPC peer device as the primary device for the vPC. Then, you would manually configure the role value for the vPC peer device that you want as the primary device to be lower than the other vPC peer device.

vPCs do not support role preemption. If the primary vPC peer device fails, the secondary vPC peer device takes over to become operationally the vPC primary device. However, the original operational roles are not restored if the formerly primary vPC comes up again.

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **role priority** *priority*
4. **exit**
5. **show vpc role**
6. **copy running-config startup-config**

DETAILED STEPS

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

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

Step 2	**vpc domain** *domain-id*
	Example:
	`switch(config)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Enters the vPC domain number that you want to configure. The system enters vpc-domain configuration mode.

Step 3	**role priority** *priority*
	Example:
	`switch(config-vpc-domain)# role priority 4`
	`switch(config-vpc-domain)#`
		Enters the role priority that you want for the vPC system priority. The range of values is from 1 to 65636, and the default value is 32667. A lower value means that this switch has a better chance of being the primary vPC.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

Step 5	**show vpc role** **Example:** `switch# show vpc role`	(Optional) Displays the vPC system priority.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to manually configure the role priority of the vPC peer device:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# role priority 4
switch(config-vpc-domain)# exit
switch(config)#
```

Configuring vPC Object Tracking Tracking Feature on a Single-Module vPC

If you must configure all the vPC Peer-Links and core-facing interfaces on a single module, you should configure a track object and a track list that is associated with the Layer 3 link to the core and on all the links on the vPC Peer-Link on both primary vPC peer devices. Once you configure this feature and if the primary vPC peer device fails, the system automatically suspends all the vPC links on the primary vPC peer device. This action forces all the vPC traffic to the secondary vPC peer device until the system stabilizes.

You must put this configuration on both vPC peer devices. Additionally, you should put the identical configuration on both vPC peer devices because either device can become the operationally primary vPC peer device.

Before you begin

Ensure that you have enabled the vPC feature.

Ensure that you have configured the track object and the track list. Ensue that you assign all interfaces that connect to the core and to the vPC Peer-Link to the track-list object on both vPC peer devices.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **track** *track-object-id*
4. **exit**
5. **show vpc brief**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vpc domain** *domain-id*
	Example:
	`switch(config)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Enters the vPC domain number that you want to configure, and enters vpc-domain configuration mode.

Step 3	**track** *track-object-id*
	Example:
	`switch(config-vpc-domain)# track object 23`
	`switch(config-vpc-domain)#`
		Adds the previously configured track-list object with its associated interfaces to the vPC domain. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for information about configuring object tracking and track lists.

Step 4	**exit**
	Example:
	`switch(config-vpc-domain)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

Step 5	**show vpc brief** **Example:** `switch# show vpc brief`	(Optional) Displays information about the tracked objects.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to put the previously configured track-list object into the vPC domain on the vPC peer device:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# track object 5
switch(config-vpc-domain)# exit
switch(config)#
```

Configuring for Recovery After an Outage

If an outage occurs, the vPC waits for a peer adjacency to form on a switch reload. This situation can result in an unacceptably long service disruption. You can configure the Cisco Nexus 9000 Series device to restore vPC services when its peer fails to come on line.

Configuring Reload Restore

The **reload restore** command and procedure described in this section is deprecated. We recommend that you use the **auto-recovery** command and procedure described in the "Configuring an Autorecovery" section.

You can configure the Cisco Nexus 9000 Series device to restore vPC services when its peer fails to come online by using the **reload restore** command.

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **reload restore** [**delay** *time-out*]
4. **exit**
5. **show running-config vpc**
6. **show vpc consistency-parameters interface port-channel** *number*
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vpc domain** *domain-id* **Example:**	Enters the vPC domain number that you want to configure, and enters vpc-domain configuration mode.

	Command or Action	Purpose
	`switch(config)# vpc domain 5` `switch(config-vpc-domain)#`	
Step 3	**reload restore** [**delay** *time-out*] **Example:** `switch(config-vpc-domain)# reload restore`	Configures the vPC to assume its peer is not functional and to bring up the vPC. The default delay is 240 seconds. You can configure a time-out delay from 240 to 3600 seconds. Use the **no** form of the command to reset the vPC to its default settings.

Step 4	**exit**
	Example:
	`switch(config-vpc-domain)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

Step 5	**show running-config vpc** **Example:** `switch# show running-config vpc`	(Optional) Displays information about the vPC, specifically the reload status.
Step 6	**show vpc consistency-parameters interface port-channel** *number* **Example:** `switch# show vpc consistency-parameters interface port-channel 1`	(Optional) Displays information about the vPC consistency parameters for the specified interface.
Step 7	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration. **Note** To ensure the reload feature is enabled, you should perform this step.

Example

This example shows how to set the vPC reload restore feature and save it in the switch startup configuration:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vpc domain 5
switch(config-vpc-domain)# reload restore

Warning:
Enables restoring of vPCs in a peer-detached state after reload, will wait for 240
seconds (by default) to determine if peer is un-reachable

switch(config-vpc-domain)# exit
switch(config)# exit
switch# copy running-config startup-config
switch# show running-config vpc

!Command: show running-config vpc
!Time: Wed Mar 24 18:43:54 2010
version 5.0(2)
```

```
feature vpc
logging level vpc 6
vpc domain 5
reload restore
```

This example shows how to examine the consistency parameters:

```
switch# show vpc consistency-parameters interface port-channel 1

Legend:
Type 1 : vPC will be suspended in case of mismatch
Name Type Local Value Peer Value
------------- ---- ----------- ---------------
STP Port Type 1 Default -
STP Port Guard 1 None -
STP MST Simulate PVST 1 Default -
mode 1 on -
Speed 1 1000 Mb/s -
Duplex 1 full -
Port Mode 1 trunk -
Native Vlan 1 1 -
MTU 1 1500 -
Allowed VLANs - 1-3967,4048-4093
Local suspended VLANs
```

Configuring an Autorecovery

You can configure the Cisco Nexus 9000 Series device to restore vPC services when its peer fails to come online by using the auto-recovery command.

You can configure the Cisco Nexus 9000 Series device to restore vPC services on the secondary vPC peer when its vPC primary peer fails and bringing down peer-keepalive and vPC Peer-Link, by using the **auto-recovery** command. In case of failure of primary switch where both peer-keepalive and vPC Peer-Links are down secondary switch will suspend vPC member. However, after 3 missed keepalive heartbeats secondary switch resumes the role of a primary switch and bring up vPC member ports. The **auto-recovery reload restore** command can be used in scenarios when vPC primary switch reloads, where secondary switch resumes the role of the vPC primary and bring ip VPC member ports.

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **auto-recovery** [**reload-delay** *time*]
4. **exit**
5. **show running-config vpc**
6. **show vpc consistency-parameters interface port-channel** *number*
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vpc domain** *domain-id*
	Example:
	`switch(config)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Enters the vPC domain number that you want to configure, and enters vpc-domain configuration mode.

Step 3	**auto-recovery** [**reload-delay** *time*] **Example:** `switch(config-vpc-domain)# auto-recovery`	Configures the vPC to assume its peer is not functional and to bring up the vPC, and specifies the time to wait after a reload to restore the vPC. The default delay is 240 seconds. You can configure a delay from 240 to 3600 seconds. Use the **no** form of the command to reset the vPC to its default settings.

Step 4	**exit**
	Example:
	`switch(config-vpc-domain)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

Step 5	**show running-config vpc** **Example:** `switch# show running-config vpc`	(Optional) Displays information about the vPC, specifically the reload status.
Step 6	**show vpc consistency-parameters interface port-channel** *number* **Example:** `switch# show vpc consistency-parameters interface port-channel 1`	(Optional) Displays information about the vPC consistency parameters for the specified interface.
Step 7	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration. **Note** To ensure the autorecovery feature is enabled, you should perform this step.

Example

This example shows how to set the vPC autorecovery feature and save it in the switch startup configuration:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vpc domain 5
```

```
switch(config-vpc-domain)# auto-recovery
switch(config-vpc-domain)# auto-recovery auto-recovery reload-delay 100

Warning:
Enables restoring of vPCs in a peer-detached state after reload, will wait for 240
seconds to determine if peer is un-reachable

switch(config-vpc-domain)# exit
switch(config)# exit
switch# copy running-config startup-config
```

Configuring the Suspension of Orphan Ports

When a device that is not vPC-capable connects to each peer, the connected ports are known as orphan ports because they are not members of a vPC. You can explicitly declare physical interfaces as orphan ports to be suspended (shut down) by the secondary peer when it suspends its vPC ports in response to a vPC Peer-Link or peer-keepalive failure. The orphan ports are restored when the vPC is restored.

Note You can configure vPC orphan port suspension only on physical ports, portchannels. However, you cannot configure the same on individual port channel member ports.

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. **configure terminal**
2. **show vpc orphan-ports**
3. **interface** *type slot/port*
4. **vpc orphan-port suspend**
5. **exit**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**show vpc orphan-ports** **Example:** `switch# show vpc orphan-ports`	(Optional) Displays a list of the orphan ports.
Step 3	**interface** *type slot/port* **Example:**	Specifies an interface to configure, and enters interface configuration mode.

	Command or Action	Purpose
	`switch(config)# interface ethernet 3/1` `switch(config-if)#`	
Step 4	**vpc orphan-port suspend** **Example:** `switch(config-if)# vpc orphan-ports suspend`	Configures the selected interface as a vPC orphan port to be suspended by the secondary peer in the case of a vPC failure.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch#`
		Exits interface configuration mode.

Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure an interface as a vPC orphan port to be suspended by the secondary peer in the case of a vPC failure:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# vpc orphan-ports suspend
switch(config-if)# exit
switch(config)#
```

Configuring the vPC Peer Switch

You can configure the Cisco Nexus 9000 Series device to make a pair of vPC devices appear as a single STP root in the Layer 2 topology.

Configuring a Pure vPC Peer Switch Topology

You can configure a pure vPC peer switch topology by using the peer-switch command and then setting the best possible (lowest) spanning tree bridge priority value.

Before you begin

Ensure that you have enabled the vPC feature.

Note When using a non-VPC dedicated trunk link between the VPC peers, the non-VPC VLANs should have a different global priority on the peers to prevent STP from blocking the VLANs.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **peer-switch**
4. **spanning-tree vlan** *vlan-range* **priority** *value*
5. **exit**
6. **show spanning-tree summary**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**vpc domain** *domain-id*
	Example:
	`switch(config)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Enters the vPC domain number that you want to configure, and enters vpc-domain configuration mode.

Step 3	**peer-switch** **Example:** `switch(config-vpc-domain)# peer-switch`	Enables the vPC switch pair to appear as a single STP root in the Layer 2 topology. Use the **no** form of the command to disable the peer switch vPC topology.
Step 4	**spanning-tree vlan** *vlan-range* **priority** *value* **Example:** `switch(config)# spanning-tree vlan 1 priority 8192`	Configures the bridge priority of the VLAN. Valid values are multiples of 4096. The default value is 32768.

Step 5	**exit**
	Example:
	`switch(config-vpc-domain)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

Step 6	**show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	(Optional) Displays a summary of the spanning tree port states including the vPC peer switch.
Step 7	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure a pure vPC peer switch topology:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vpc domain 5
switch(config-vpc-domain)# peer-switch

2010 Apr 28 14:44:44 switch %STP-2-VPC_PEERSWITCH_CONFIG_ENABLED: vPC peer-switch
configuration is enabled. Please make sure to configure spanning tree "bridge" priority as
per recommended guidelines to make vPC peer-switch operational.

switch(config-vpc-domain)# spanning-tree vlan 1 priority 8192
switch(config-vpc-domain)# exit
switch(config)#
```

Configuring a Hybrid vPC Peer Switch Topology

You can configure a hybrid vPC and non-vPC peer switch topology by using the spanning-tree pseudo-information command to change the designated bridge ID so that it meets the STP VLAN-based load-balancing criteria and then change the root bridge ID priority to a value that is better than the best bridge priority. You then enable the peer switch.

Before you begin

Ensure that you have enabled the vPC feature.

When using a non-VPC dedicated trunk link between the VPC peers, the non-VPC VLANs should have a different pseudo root priority on the peers to prevent STP from blocking the VLANs.

SUMMARY STEPS

1. **configure terminal**
2. **spanning-tree pseudo-information**
3. **vlan** *vlan-id* **designated priority** *priority*
4. **vlan** *vlan-id* **root priority** *priority*
5. **vpc domain** *domain-id*
6. **peer-switch**
7. **exit**
8. **show spanning-tree summary**
9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	**spanning-tree pseudo-information**
	Example:
	`switch(config)# spanning-tree pseudo-information`
	`switch(config-pseudo)#`
		Configures the spanning tree pseudo information.

Step 3	**vlan** *vlan-id* **designated priority** *priority* **Example:** `switch(config-pseudo)# vlan 1 designated priority 8192`	Configures the designated bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440.
Step 4	**vlan** *vlan-id* **root priority** *priority* **Example:** `switch(config-pseudo)# vlan 1 root priority 4096`	Configures the root bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440.

Step 5	**vpc domain** *domain-id*
	Example:
	`switch(config)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Enters the vPC domain number that you want to configure, and enters vpc-domain configuration mode.

Step 6	**peer-switch** **Example:** `switch(config-vpc-domain)# peer-switch`	Enables the vPC switch pair to appear as a single STP root in the Layer 2 topology. Use the **no** form of the command to disable the peer switch vPC topology.

Step 7	**exit**
	Example:
	`switch(config-vpc-domain)# exit`
	`switch#`
		Exits vpc-domain configuration mode.

Step 8	**show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	(Optional) Displays a summary of the spanning tree port states including the vPC peer switch.
Step 9	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure a hybrid vPC peer switch topology:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# spanning-tree pseudo-information
switch(config-pseudo)# vlan 1 designated priority 8192
switch(config-pseudo)# vlan 1 root priority 4096
```

```
switch(config-pseudo)# vpc domain 5
switch(config-vpc-domain)# peer-switch
switch(config-vpc-domain)# exit
switch(config)#
```

Configuring Hitless vPC Role Change

Complete these steps to enable hitless vPC role change.

Before you begin

- • Ensure that the vPC feature is enabled.
- • Ensure that the vPC Peer-Link is up
- • Verify the role priority of devices

SUMMARY STEPS

1. **vpc role preempt**
2. **show vpc role**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**vpc role preempt**
	Example:
	`switch# vpc role preempt`
	`switch(config)#`
		Enable hitless vPC role change.

Step 2	**show vpc role** **Example:** `switch(config)# show vpc role`	(Optional) Verify hitless vPC role change feature.

Example

This example on how to configure hitless vPC role change:

```
switch# show vpc rolevPC Role status
----------------------------------------------------
vPC role                        : secondary
vPC system-mac                  : 00:23:04:ee:be:01
vPC system-priority             : 32667
vPC local system-mac            : 8c:60:4f:03:84:41
vPC local role-priority         : 32668
vPC peer system-mac             : 8c:60:4f:03:84:43
vPC peer role-priority          : 32667

! Configure vPC hitless role change on the device!

switch(config)# vpc role preempt
! The following is an output from the show vpc role command after the
```

```
vPC hitless feature is configured
switch(config)# show vpc role
vPC Role status
----------------------------------------------------
vPC role                      : primary
vPC system-mac                : 00:00:00:00:00:00
vPC system-priority           : 32667
vPC local system-mac          : 8c:60:4f:03:84:41
vPC local role-priority       : 32666
vPC peer system-mac           : 8c:60:4f:03:84:43
vPC peer role-priority        : 32667

switch(config)#
```

Use Case Scenario for vPC Role Change

The hitless vPC role change feature can be used in the following scenarios:

- Role change request—When you want to change the roles of the peer devices in a vPC domain.
- Primary switch reload—When the devices comes up after a reload and roles are defined, you can use the hitless vPC role change feature to restore the roles. For example, after a reload if the primary device takes the role of operational secondary and the secondary device takes the role of primary operational, you can change the vPC peer roles to their original defined roles using the **vpc role preempt** command.

Note Always check the existing device role priority before switching vPC role.

- Dual-active recovery—In a dual-active recovery scenario, the vPC primary switch continues to be (operational) primary, but the vPC secondary switch becomes the targeted primary switch and keeps its vPC member ports up. You can use the vPC hitless feature and restore the device roles. After the Dual-active recovery, if one side is operational primary and the other side operational secondary, then you can use the **vpc role preempt** command to restore the device roles to be primary and secondary

Enabling STP to Use the Cisco MAC Address

This procedure enables STP to use the Cisco MAC address (00:26:0b:xx:xx:xx).

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. **configure terminal**
2. **vpc domain** *domain-id*
3. **[no] mac-address bpdu source version 2**
4. **exit**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**vpc domain** *domain-id* **Example:** `switch(config)# vpc domain 5`	Creates a vPC domain if it does not already exist, and enters vpc-domain configuration mode.
Step 3	**[no] mac-address bpdu source version 2** **Example:** `switch(config-vpc-domain)# mac-address bpdu source version 2`	Enables STP to use the Cisco MAC address (00:26:0b:xx:xx:xx) as the source address of BPDUs generated on vPC ports.
Step 4	**exit** **Example:** `switch(config-vpc-domain)# exit`	Exits vpc-domain configuration mode.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(confg)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the vPC Configuration

To display vPC configuration information, perform one of the following tasks:

Command	Purpose
show feature	Displays whether the vPC is enabled or not.
show vpc brief	Displays brief information about the vPCs.
show vpc consistency-parameters	Displays the status of those parameters that must be consistent across all vPC interfaces.
show running-config vpc	Displays running configuration information for vPCs.
show port-channel capacity	Displays how many port channels are configured and how many are still available on the device.
show vpc statistics	Displays statistics about the vPCs.
show vpc peer-keepalive	Displays information about the peer-keepalive messages.

Command	Purpose
show vpc role	Displays the peer status, the role of the local device, the vPC system MAC address and system priority, and the MAC address and priority for the local vPC device.

Monitoring vPCs

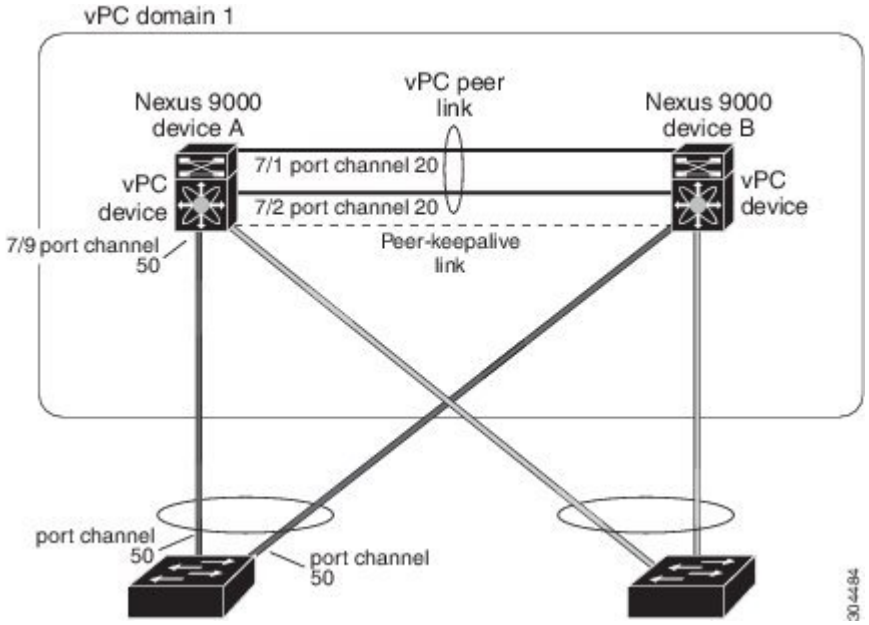
Use the **show vpc statistics**command to display vPC statistics.

Note This command displays the vPC statistics only for the vPC peer device that you are working on.

Configuration Examples for vPCs

The following example shows how to configure vPC on device A as shown in the figure:

Figure 27: vPC Configuration Example



1. Enable vPC and LACP.

```
switch# configure terminal
switch(config)# feature vPC
switch(config)# feature lacp
```

2. (Optional) Configure one of the interfaces that you want to be a vPC Peer-Link in the dedicated port mode.

```
switch(config)# interface ethernet 7/1,
ethernet 7/3, ethernet 7/5. ethernet 7/7
switch(config-if)# shutdown
switch(config-if)# exit
switch(config)# interface ethernet 7/1

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

3. (Optional) Configure the second, redundant interface that you want to be a vPC Peer-Link in the dedicated port mode.

```
switch(config)# interface ethernet 7/2, ethernet 7/4,
ethernet 7/6. ethernet 7/8
switch(config-if)# shutdown
switch(config-if)# exit
switch(config)# interface ethernet 7/2

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

4. Configure the two interfaces (for redundancy) that you want to be in the vPC Peer-Link to be an active Layer 2 LACP port channel.

```
switch(config)# interface ethernet 7/1-2
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk allowed vlan 1-50
switch(config-if)# switchport trunk native vlan 20
switch(config-if)# channel-group 20 mode active
switch(config-if)# exit
```

5. Create and enable the VLANs.

```
switch(config)# vlan 1-50
switch(config-vlan)# no shutdown
switch(config-vlan)# exit
```

6. Create a separate VRF for the vPC peer-keepalive link and add a Layer 3 interface to that VRF.

```
switch(config)# vrf context pkal
switch(config-vrf)# exit
switch(config)# interface ethernet 8/1
switch(config-if)# vrf member pkal
switch(config-if)# ip address 172.23.145.218/24
switch(config-if)# no shutdown
switch(config-if)# exit
```

7. Create the vPC domain and add the vPC peer-keepalive link.

```
switch(config)# vpc domain 1
switch(config-vpc-domain)# peer-keepalive
 destination 172.23.145.217 source 172.23.145.218 vrf pkal
switch(config-vpc-domain)# exit
```

8. Configure the vPC vPC Peer-Link.

```
switch(config)# interface port-channel 20
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk allowed vlan 1-50
switch(config-if)# vpc peer-link
```

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

9. Configure the interface for the port channel to the downstream device of the vPC.

```
switch(config)# interface ethernet 7/9
switch(config-if)# switchport mode trunk
switch(config-if)# allowed vlan 1-50
switch(config-if)# native vlan 20
switch(config-if)# channel-group 50 mode active
switch(config-if)# exit
switch(config)# interface port-channel 50
switch(config-if)# vpc 50
switch(config-if)# exit
switch(config)#
```

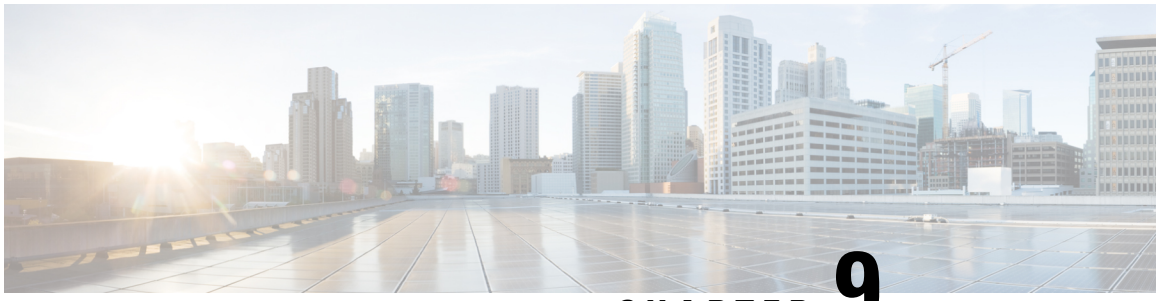
10. Save the configuration.

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

Note If you configure the port channel first, ensure that it is a Layer 2 port channel.

Related Documents

Related Topic	Related Topic
System management	System management
High availability	High availability
Release Notes	Release Notes



CHAPTER 9

Configuring IP Tunnels

This chapter describes how to configure IP tunnels using Generic Route Encapsulation (GRE) on Cisco NX-OS devices.

Information About IP Tunnels

IP tunnels can encapsulate a same-layer or higher layer protocol and transport the result over IP through a tunnel created between two devices.

IP Tunnel Overview

IP tunnels consists of the following three main components:

- Passenger protocol—The protocol that needs to be encapsulated. IPv4 is an example of a passenger protocol.
- Carrier protocol—The protocol that is used to encapsulate the passenger protocol. Cisco NX-OS supports GRE as a carrier protocol.
- Transport protocol—The protocol that is used to carry the encapsulated protocol. IPv4 is an example of a transport protocol. An IP tunnel takes a passenger protocol, such as IPv4, and encapsulates that protocol within a carrier protocol, such as GRE. The device then transmits this carrier protocol over a transport protocol, such as IPv4.

You configure a tunnel interface with matching characteristics on each end of the tunnel.

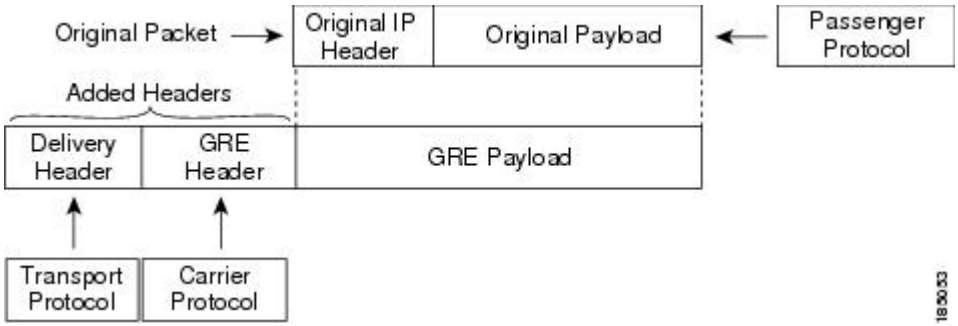
You must enable the tunnel feature before you can configure it. The system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. See the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide for information about rollbacks and checkpoints.

GRE Tunnels

You can use generic routing encapsulation (GRE) as the carrier protocol for a variety of passenger protocols.

The following figure shows the IP tunnel components for a GRE tunnel. The original passenger protocol packet becomes the GRE payload and the device adds a GRE header to the packet. The device then adds the transport protocol header to the packet and transmits it.

Figure 28: GRE PDU



Point-to-Point IP-in-IP Tunnel Encapsulation and Decapsulation

Note The selection of GRE or IP-in-IP tunnel destination based on the PBR policy is not supported.

Multi-Point IP-in-IP Tunnel Decapsulation

Path MTU Discovery

Path maximum transmission unit (MTU) discovery (PMTUD) prevents fragmentation in the path between two endpoints by dynamically determining the lowest MTU along the path from the packet's source to its destination. PMTUD reduces the send MTU value for the connection if the interface receives information that the packet would require fragmentation.

When you enable PMTUD, the interface sets the Don't Fragment (DF) bit on all packets that traverse the tunnel. If a packet that enters the tunnel encounters a link with a smaller MTU than the MTU value for the packet, the remote link drops the packet and sends an ICMP message back to the sender of the packet. This message indicates that fragmentation was required (but not permitted) and provides the MTU of the link that dropped the packet.

Note PMTUD on a tunnel interface requires that the tunnel endpoint can receive ICMP messages generated by devices in the path of the tunnel. Check that ICMP messages can be received before using PMTUD over firewall connections.

High Availability

IP tunnels support stateful restarts. A stateful restart occurs on a supervisor switchover. After the switchover, Cisco NX-OS applies the runtime configuration after the switchover.

Prerequisites for IP Tunnels

IP tunnels have the following prerequisites:

- You must be familiar with TCP/IP fundamentals to configure IP tunnels.
- You are logged on to the switch.
- You must enable the tunneling feature in a device before you can configure and enable any IP tunnels.

Guidelines and Limitations

IP tunnels have the following configuration guidelines and limitations:

- The **show** commands with the **internal** keyword are not supported.
- Cisco NX-OS supports only the following protocols:
  - IPv4 passenger protocol.
  - GRE carrier protocol.
- Cisco NX-OS supports the following maximum number of tunnels:
  - IP tunnels - 8 tunnels.
  - GRE and IP-in-IP regular tunnels - 8 tunnels. (6.1(2)I3(4) and later)
- IP tunnels do not support access control lists (ACLs) or QoS policies.
- Cisco NX-OS supports the GRE header defined in IETF RFC 2784. Cisco NX-OS does not support tunnel keys and other options from IETF RFC 1701.
- Cisco NX-OS does not support GRE tunnel keepalives.
- All unicast routing protocols are supported by IP tunnels.
- The IP tunnel interface cannot be configured to be a span source or destination.
- IP tunnels do not support PIM or other Multicast features and protocols. (6.1(2)I3(4) and later)
- The selection of GRE or IP-in-IP tunnel based on the PBR policy is not supported. (6.1(2)I3(4) and later)
- The **feature tunnel** feature on Cisco Nexus 9000 switches cannot co-exist with the VXLAN feature **feature nv overlay**.
- IP tunnels are supported only in the default **system routing** mode and not in other modes. (6.1(2)I3(4) and later)

- BGP adjacency over tunnel is not supported in a scenario where the tunnel interface and tunnel source are in same VRF (example: VRF-A) and tunnel destination is reachable with route-leak from opposite end (example: via VRF-B)
- Configuring two tunnel interfaces with the same source and destination address is not supported. Loopback interfaces may be configured as the source addresses instead.
- Cisco Nexus 9200, 9300-EX, 9300-FX, 9300-FX2 series switches and Cisco Nexus 9500 platform switches with 9700-EX/FX line cards may not have multiple tunnel interfaces in a single VRF that are sourced from or destined to the same IP address. For example, a device may not have tunnel 0 and tunnel 1 interfaces in the default VRF that are sourced from the same IP address or interface.

Default Settings

The following table lists the default settings for IP tunnel parameters.

Table 14: Default IP Tunnel Parameters

Parameters	Default
Path MTU discovery age timer	10 minutes
Path MTU discovery minimum MTU	64
Tunnel feature	Disabled

Configuring IP Tunnels

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Enabling Tunneling

You must enable the tunneling feature before you can configure any IP tunnels.

SUMMARY STEPS

1. **configure terminal**
2. **feature tunnel**
3. **exit**
4. **show feature**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**feature tunnel**
	Example:
	`switch(config)# feature tunnel`
	`switch(config-if)#`
		Allows the creation of a new tunnel interface.
		To disable the tunnel interface feature, use the **no** form of this command.

Step 3	**exit**
	Example:
	`switch(config-if)# exit`
	`switch#`
		Exits the interface mode and returns to the configuration mode.

Step 4	**show feature** **Example:** `switch(config-if)# show feature`	(Optional) Displays information about the features enabled on the device.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves this configuration change.

Creating a Tunnel Interface

You can create a tunnel interface and then configure this logical interface for your IP tunnel.

Note Cisco NX-OS supports a maximum of 8 IP tunnels.

Note Use the **no interface tunnel** command to remove the tunnel interface and all associated configuration.

Command	Purpose
no interface tunnel *number* **Example:** `switch(config)# no interface tunnel 1`	Deletes the tunnel interface and the associated configuration.
description *string* **Example:** `switch(config-if)# description GRE tunnel`	Configures a description for the tunnel.
mtu *value* **Example:** `switch(config-if)# mtu 1400`	Sets the MTU of IP packets sent on an interface.
tunnel ttl *value* **Example:** `switch(config-if)# tunnel ttl 100`	Sets the tunnel time-to-live value. The range is from 1 to 255.

Before you begin

You can configure the tunnel source and the tunnel destination in different VRFs. Ensure that you have enabled the tunneling feature.

SUMMARY STEPS

1. **configure terminal**
2. **interface tunnel** *number*
3. **tunnel mode** {**gre ip** | **ipip** {**ip** | **decapsulate-any**}}
4. **tunnel source** {*ip-address* |*interface-name*}
5. **tunnel destination** {*ip-address* |*host-name*}
6. **tunnel use-vrf** *vrf-name*
7. **show interfaces tunnel** *number*
8. **copy running-config startup-config**

DETAILED STEPS

	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 tunnel** *number*
	Example:
	`switch(config)# interface tunnel 1`
	`switch(config-if)#`
		Creates a new tunnel interface.

Step 3	**tunnel mode** {**gre ip** | **ipip** {**ip** | **decapsulate-any**}}	Sets this tunnel mode to GRE, ipip, or ipip decapsulate-only. The **gre** and **ip** keywords specify that GRE encapsulation over IP will be used. The **ipip** keyword specifies that IP-in-IP encapsulation will be used. The optional **decapsulate-any** keyword terminates IP-in-IP tunnels at one tunnel interface. This keyword creates a tunnel that will not carry any outbound traffic. However, remote tunnel endpoints can use a tunnel configured as their destination.
Step 4	**tunnel source** {*ip-address* |*interface-name*} **Example:** `switch(config-if)# tunnel source ethernet 1/2`	Configures the source address for this IP tunnel. The source can be specified by IP address or logical interface name.
Step 5	**tunnel destination** {*ip-address* |*host-name*} **Example:** `switch(config-if)# tunnel destination 192.0.2.1`	Configures the destination address for this IP tunnel. The destination can be specified by IP address or logical host name.
Step 6	**tunnel use-vrf** *vrf-name* **Example:** `switch(config-if)# tunnel use-vrf blue`	(Optional) Uses the configured VRF to look up the tunnel IP destination address.
Step 7	**show interfaces tunnel** *number* **Example:** `switch# show interfaces tunnel 1`	(Optional) Displays the tunnel interface statistics.
Step 8	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves this configuration change.

Example

This example shows how to create a tunnel interface

```
switch# configure terminal
switch(config)# interface tunnel 1
switch(config-if)# tunnel source ethenet 1/2
switch(config-if)# tunnel destination 192.0.2.1
switch(config-if)# copy running-config startup-config
```

Configuring a Tunnel Interface

You can set a tunnel interface to GRE tunnel mode, ipip mode, or ipip decapsulate-only mode. GRE mode is the default tunnel mode. .

Before you begin

Ensure that you have enabled the tunneling feature.

SUMMARY STEPS

1. **configure terminal**
2. **interface tunnel** *number*
3. **tunnel mode** {**gre ip** | **ipip** | {**ip** | **decapsulate-any**}}
4. **show interfaces tunnel** *number*
5. **mtu** *value*
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface tunnel** *number*
	Example:
	`switch(config)# interface tunnel 1`
	`switch(config-if)#`
		Creates a new tunnel interface.

Step 3	**tunnel mode** {**gre ip** | **ipip** | {**ip** | **decapsulate-any**}}	Sets this tunnel mode to GRE, ipip, or ipip decapsulate-only. The **gre** and **ip** keywords specify that GRE encapsulation over IP will be used. The ipip keyword specifies that IP-in-IP encapsulation will be used. The optional **decapsulate-any** keyword terminates IP-in-IP tunnels at one tunnel interface. This keyword creates a tunnel that will not carry any outbound traffic. However, remote tunnel endpoints can use a tunnel configured as their destination.
Step 4	**show interfaces tunnel** *number* **Example:** `switch(config-if)# show interfaces tunnel 1`	(Optional) Displays the tunnel interface statistics.
Step 5	**mtu** *value*	Sets the maximum transmission unit (MTU) of IP packets sent on an interface. The range is from 64 to 9192 units.

	Command or Action	Purpose
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves this configuration change.

Example

This example shows how to create the tunnel interface to GRE:

```
switch# configure terminal
switch(config)# interface tunnel 1
switch(config-if)# tunnel mode gre ip
switch(config-if)# copy running-config startup-config
```

This example shows how to create an ipip tunnel:

```
switch# configure terminal
switch(config)# interface tunnel 1
switch(config-if)# tunnel mode ipip
switch(config-if)# mtu 1400
switch(config-if)# copy running-config startup-config
switch(config-if)# no shut
```

Configuring a GRE Tunnel

You can set a tunnel interface to GRE tunnel mode.

Note Cisco NX-OS supports only the GRE protocol for IPV4 over IPV4.

Before you begin

Ensure that you have enabled the tunneling feature.

SUMMARY STEPS

1. **configure terminal**
2. **interface tunnel** *number*
3. **tunnel mode gre ip**
4. **show interfaces tunnel** *number*
5. **copy running-config startup-config**

DETAILED STEPS

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

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

Step 2	**interface tunnel** *number*
	Example:
	`switch(config)# interface tunnel 1`
	`switch(config-if)#`
		Creates a new tunnel interface.

Step 3	**tunnel mode gre ip** **Example:** `switch(config-if)# tunnel mode gre ip`	Sets this tunnel mode to GRE.
Step 4	**show interfaces tunnel** *number* **Example:** `switch(config-if)# show interfaces tunnel 1`	(Optional) Displays the tunnel interface statistics.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves this configuration change.

Enabling Path MTU Discovery

Use the **tunnel path-mtu discovery** command to enable path MTU discovery on a tunnel.

SUMMARY STEPS

1. **tunnel path-mtu-discovery age-timer** *min*
2. **tunnel path-mtu-discovery min-mtu** *bytes*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**tunnel path-mtu-discovery age-timer** *min* **Example:** `switch(config-if)# tunnel path-mtu-discovery age-timer 25`	Enables Path MTU Discovery (PMTUD) on a tunnel interface. • min—Number of minutes. The range is from 10 to 30. The default is 10.
Step 2	**tunnel path-mtu-discovery min-mtu** *bytes* **Example:** `switch(config-if)# tunnel path-mtu-discovery min-mtu 1500`	Enables Path MTU Discovery (PMTUD) on a tunnel interface. • bytes—Minimum MTU recognized. The range is from 64 to 9192. The default is 64.

Assigning VRF Membership to a Tunnel Interface

You can add a tunnel interface to a VRF.

Before you begin

Ensure that you have enabled the tunneling feature.

Assign the IP address for a tunnel interface after you have configured the interface for a VRF.

SUMMARY STEPS

1. **configure terminal**
2. **interface tunnel** *number*
3. **vrf member** *vrf-name*
4. **ip address** *ip-prefix/length*
5. **show vrf** [*vrf-name*] **interface** *interface-type number*
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface tunnel** *number*
	Example:
	`switch(config)# interface tunnel 0`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member RemoteOfficeVRF`	Adds this interface to a VRF.
Step 4	**ip address** *ip-prefix/length* **Example:** `switch(config-if)# ip address 192.0.2.1/16`	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 5	**show vrf** [*vrf-name*] **interface** *interface-type number* **Example:** `switch(config-vrf)# show vrf Enterprise interface tunnel 0`	(Optional) Displays VRF information.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Saves this configuration change.

Example

This example shows how to add a tunnel interface to the VRF:

```
switch# configure terminal
switch(config)# interface tunnel 0
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 209.0.2.1/16
switch(config-if)# copy running-config startup-config
```

Verifying the IP Tunnel Configuration

To verify the IP tunnel configuration information, perform one of the following tasks:

Command	Purpose
show interface tunnel *number*	Displays the configuration for the tunnel interface (MTU, protocol, transport, and VRF). Displays input and output packets, bytes, and packet rates.
show interface tunnel *number* **brief**	Displays the operational status, IP address, encapsulation type, and MTU of the tunnel interface.
show interface tunnel *number* **counters**	Displays interface counters of input/output packets. **Note** The byte count displayed with the interface counters include the internal header size.
show interface tunnel *number* **description**	Displays the configured description of the tunnel interface.
show interface tunnel *number* **status**	Displays the operational status of the tunnel interface.
show interface tunnel *number* **status err-disabled**	Displays the error disabled status of the tunnel interface.

Configuration Examples for IP Tunneling

The following example shows a simple GRE tunnel. Ethernet 1/2 is the tunnel source for router A and the tunnel destination for router B. Ethernet interface 2/1 is the tunnel source for router B and the tunnel destination for router A.

Router A:

```
feature tunnel
interface tunnel 0
ip address 209.165.20.2/8
tunnel source ethernet 1/2
tunnel destination 192.0.2.2
tunnel mode gre ip
```

```
tunnel path-mtu-discovery 25 1500

interface ethernet 1/2
ip address 192.0.2.55/8
```

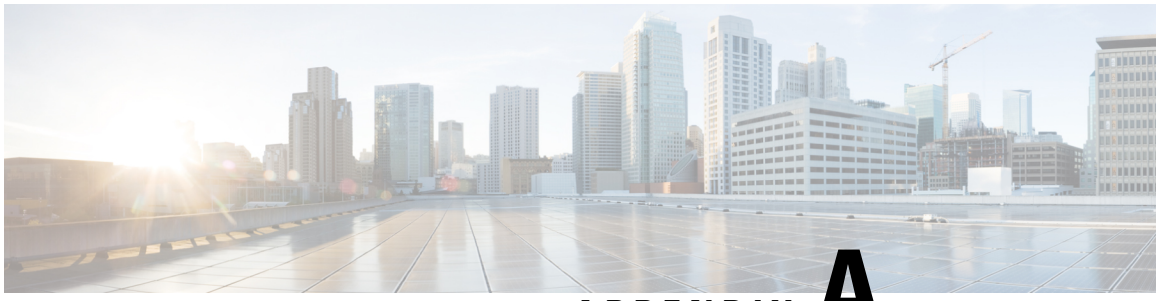
Router B:

```
feature tunnel
interface tunnel 0
ip address 209.165.20.1/8
tunnel source ethernet 2/1
tunnel destination 192.0.2.55
tunnel mode gre ip

interface ethernet 2/1
ip address 192.0.2.2/8
```

Related Documents

Related Topic	Document Title
IP Tunnel commands	*Cisco Nexus 9000 Series NX-OS Interfaces Command Reference*



APPENDIX A

IETF RFCs supported by Cisco NX-OS Interfaces

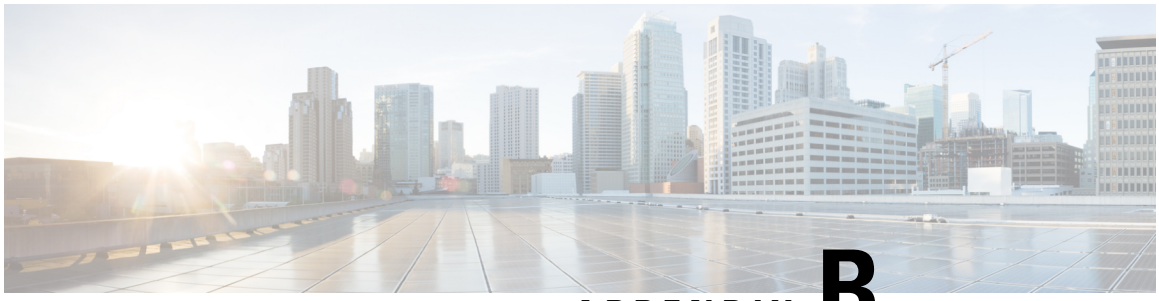
This appendix lists the IETF RFCs for interfaces supported by Cisco NX-OS.

• IPv6 RFCs, on page 243

IPv6 RFCs

RFCs	Title
RFC 2373	*IP Version 6 Addressing Architecture*
RFC 2374	*An Aggregatable Global Unicast Address Format*
RFC 2460	*Internet Protocol, Version 6 (IPv6) Specification*
RFC 2462	*IPv6 Stateless Address Autoconfiguration*
RFC 2464	*Transmission of IPv6 Packets over Ethernet Networks*
RFC 2467	*Transmission of IPv6 Packets over FDDI Networks*
RFC 2472	*IP Version 6 over PPP*
RFC 2492	*IPv6 over ATM Networks*
RFC 2590	*Transmission of IPv6 Packets over Frame Relay Networks Specification*
RFC 3021	*Using 31-Bit Prefixes on IPv4 Point-to-Point Links*
RFC 3152	*Delegation of IP6.ARPA*
RFC 3162	*RADIUS and IPv6*
RFC 3513	*Internet Protocol Version 6 (IPv6) Addressing Architecture*
RFC 3596	*DNS Extensions to Support IP version 6*
RFC 4193	*Unique Local IPv6 Unicast Addresses*

IETF RFCs supported by Cisco NX-OS Interfaces



APPENDIX B

Configuration Limits for Cisco NX-OS Interfaces

The configuration limits are documented in the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

Configuration Limits for Cisco NX-OS Interfaces