Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide 7.x

First Published: 2020-05-20

Last Modified: 2024-10-07

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

CONTENTS

Preface

The preface contains the following sections:

Audience

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

Document Conventions

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

Command descriptions use the following conventions:

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

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

Examples use the following conventions:

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

This document uses the following conventions:

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

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

Related Documentation for Cisco Nexus 7000 Series NX-OS Software

The entire Cisco Nexus 7000 Series NX-OS documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/series.html#~tab-documents

Release Notes

The release notes are available at the following URL:

http://www.cisco.com/en/US/products/ps9402/prod_release_notes_list.html

Configuration Guides

These guides are available at the following URL:

http://www.cisco.com/en/US/products/ps9402/products_installation_and_configuration_guides_list.html

The documents in this category include:

- *Cisco Nexus 7000 Series NX-OS Configuration Examples*
- *Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS IP SLAs Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS LISP Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS MPLS Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS OTV Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS SAN Switching Guide*
- *Cisco Nexus 7000 Series NX-OS Security Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Verified Scalability Guide*
- *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Virtual Device Context Quick Start*
- *Cisco Nexus 7000 Series NX-OS OTV Quick Start Guide*
- *Cisco NX-OS FCoE Configuration Guide for Cisco Nexus 7000 and Cisco MDS 9500*
- *Cisco Nexus 2000 Series Fabric Extender Software Configuration Guide*

Command References

These guides are available at the following URL:

http://www.cisco.com/en/US/products/ps9402/prod_command_reference_list.html

The documents in this category include:

- *Cisco Nexus 7000 Series NX-OS Command Reference Master Index*
- *Cisco Nexus 7000 Series NX-OS FabricPath Command Reference*
- *Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*
- *Cisco Nexus 7000 Series NX-OS High Availability Command Reference*
- *Cisco Nexus 7000 Series NX-OS Interfaces Command Reference*
- *Cisco Nexus 7000 Series NX-OS Layer 2 Switching Command Reference*
- *Cisco Nexus 7000 Series NX-OS LISP Command Reference*
- *Cisco Nexus 7000 Series NX-OS MPLS Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Multicast Routing Command Reference*
- *Cisco Nexus 7000 Series NX-OS OTV Command Reference*
- *Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference*
- *Cisco Nexus 7000 Series NX-OS SAN Switching Command Reference*
- *Cisco Nexus 7000 Series NX-OS Security Command Reference*
- *Cisco Nexus 7000 Series NX-OS System Management Command Reference*
- *Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*
- *Cisco Nexus 7000 Series NX-OS Virtual Device Context Command Reference*
- *Cisco NX-OS FCoE Command Reference for Cisco Nexus 7000 and Cisco MDS 9500*

Other Software Documents

You can locate these documents starting at the following landing page:

https://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/series.html#~tab-documents

- *Cisco Nexus 7000 Series NX-OS MIB Quick Reference*
- *Cisco Nexus 7000 Series NX-OS Software Upgrade and Downgrade Guide*
- *Cisco Nexus 7000 Series NX-OS Troubleshooting Guide*
- *Cisco NX-OS Licensing Guide*
- *Cisco NX-OS System Messages Reference*
- *Cisco NX-OS Interface User Guide*

Documentation Feedback

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

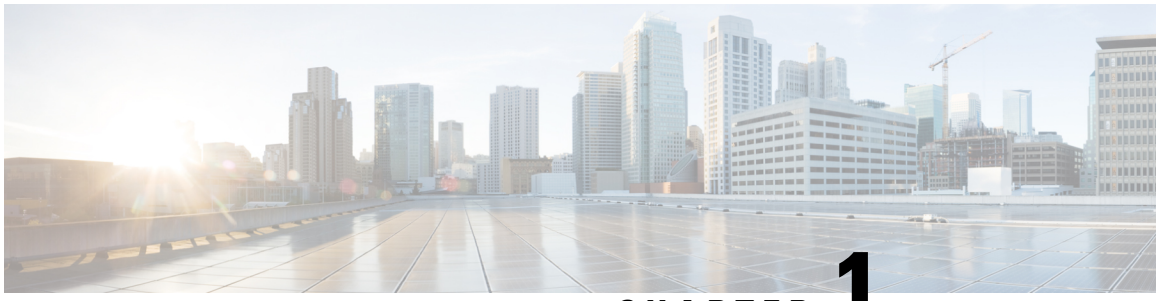
We appreciate your feedback.

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 DevNet.
- 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.



CHAPTER 1

New and Changed Information

New and Changed Information

Your software release might not support all the features in this document. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release.

Table 1: New and Changed Interfaces Features

Feature Name	Description	Changed in Release
Upgrading Line Card Module for vPC	Added support for upgrading line card module for vPC.	7.3(0)DX(1)
Per-link BFD	Added Per-link Bidirectional Forwarding feature support that enables users to configure individual BFD sessions on every Link Aggregation Group member interfaces (as defined in RFC 7130).	7.3(0)D1(1)
Hitless STP for vPC Role Change	Added support for hitless STP for vPC role change.	7.3(0)D1(1)
Asynchronous Link Debounce	Added support for setting separate values for debounce up and debounce down links.	7.3(0)D1(1)
BFD Support for HSPRv6	Added BFD support for HSPRv6.	7.3(0)D1(1)
Port Channel (Random Load Balancing)	Added support for random load balancing on port channels.	7.3(0)D1(1)

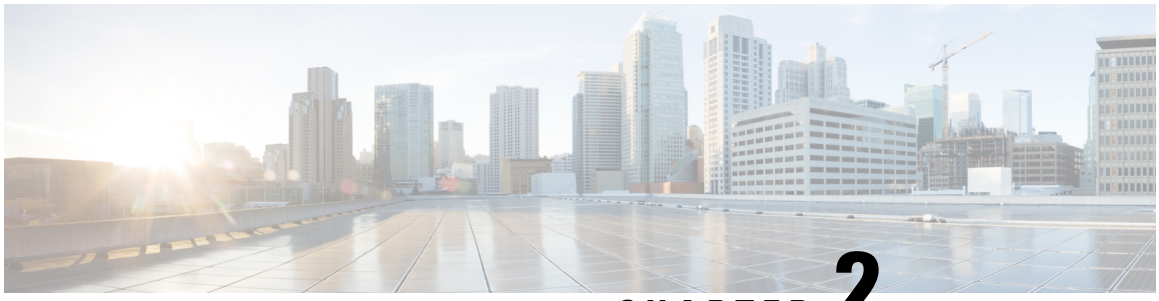
Feature Name	Description	Changed in Release
Ethernet OAM	Ethernet OAM features allow Service Providers to monitor the quality of the connections on a MAN or WAN. Service providers can monitor specific events, take actions on events, and if necessary, put specific interfaces into loopback mode for troubleshooting. Ethernet OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link.	7.3(0)D1(1)
vPC Shutdown	Added the **shutdown** command that shuts down the peer to isolate it for debugging, reloading, or physically removing it from the vPC complex, and enables the peer vPC switch to take over as the primary peer.	7.2(0)D1(1)
Physical Port vPC on F3	Added support for physical port vPCs for F3	7.2(0)D1(1)
1500 host vPC for FEX (Physical Port vPC on FEX)	Added support for this feature.	7.2(0)D1(1)
vPC Configuration Synchronization	Added support for the vPC Configuration Synchronization feature.	7.2(0)D1(1)
Layer 3 over vPC for F2, F2E and F3 Modules	Added support for this feature.	7.2(0)D1(1)
Support for BFD over Layer 2 Over a Fabricpath Core	Added support for BFD over Layer 2 over a fabricpath core.	7.2(0)D1(1)
Support for BFD over SVI Over Fabricbath Core	Added support for BFD over SVI over Fabricbath core.	7.2(0)D1(1)
GRE Tunnels	Added support for F3 Series modules.	6.2(10)
Native VLAN Tagging on Trunk Ports	Added support for the **switchport trunk native vlan tag** command and added the **exclude control** keywords to the **vlan dot1q tag native** command.	6.2(10)
LAN Shutdown	Added the **shutdown lan** command to support this feature.	6.2(6)

Feature Name	Description	Changed in Release
FCoE Over Physical Port vPC	Added support for this feature.	6.2(6)
Physical Port VPCs	Added support for physical port vPCs on the physical interface of vPC peer devices.	6.2(6)
BFD for IPv6 Static	Added support for configuring BFD for IPv6 static routes on an interface.	6.2(2a)
FEX	Cisco Fabric Extenders support Layer 3 protocol adjacencies on host interfaces (HIFs) and DSCP to queue mapping. Before Cisco NX-OS Release 6.2(2), you can configure a Fabric Extender (FEX) port as a Layer 3 interface for host connectivity, but not for routing.	6.2(2)
Error Disabled	Added the ability to view error disabled recovery and detection runtime information.	6.2(2)
Show Interface Status Error Policy	Allows you to view information about interfaces and VLANS that receive an error during policy programming.	6.2(2)
Clear SNMP Counters From an Interface	Added the ability to clear SNMP counters from the interface.	6.2(2)
SVI Autostate Disable	Allows you to disable SVI autostate behavior by allowing an SVI to stay up even if no interface is up in the corresponding VLAN.	6.2(2)
BFD on IPv6	Added support for BFD on IPv6.	6.2(2)
BFD on OSPFv3	Added support for BFD on OPSPv3.	6.2(2)
BFD on IS-ISv6	Added support for BFD on IS-ISv6.	6.2(2)
Asymmetric	Allows you to change the hash mechanism in F2 or F2e modules to asymmetric (symmetric by default), which prevents traffic-drop occurring during bi-directional forwarding and improves load balancing.	6.2(2)

Feature Name	Description	Changed in Release
Mode Auto Command	Allows you to enable certain commands simultaneously.	6.2(2)
Multicast Load Balance	Allows two peers to be partially designated forwarders when both vPC paths are up.	6.1(3)
Result Bundle Hash Load Balancing	Added support for the RBH modulo mode to improve load balancing across port channels.	6.1(3)
Minimum Links for FEX Fabric Port Channel	Added the ability to configure a minimum number of links for the FEX fabric port channel.	6.1(3)
Slow Drain Device Detection and Congestion Avoidance	Added support for the slow drain device detection feature.	6.1(1)
BFD Support on F2 Series and M2 Series Modules	Added support for F2 Series and M2 Series modules.	6.1(1)
There are no changes since Release 5.2(1)	-	-
Fabric Extender (FEX)	Fabric Extender ports have Layer 3 support for host connectivity, and vPCs can be configured through Fabric Extenders (Host vPC).	5.2(1)
BFD SHA1 Authentication	Supports SHA-1 authentication of BFD packets.	5.2(1)
Default Interfaces	Allows you to clear the existing configuration of multiple interface types.	5.2(1)
SVI Autostate Exclude	Allows you to exclude a port from the VLAN interface link-up calculation when there are multiple ports in the VLAN.	5.2(1)
vPC	Configures auto recovery support, provides system display of MST to VLAN consistency failures, FabricPath configuration support, and a vPC connection to Cisco 2000 Series Fabric Extenders.	5.2(1)
Rate Limits	Configures rate limits for packets that reach the supervisor.	5.1(1)

Feature Name	Description	Changed in Release
Inband Management in the Nexus Chassis	Configures inband management in the Cisco Nexus 7000 switches when there are only F1 series module in the chassis.	5.1(1)
F1 Series Modules and M1 Series Modules for the Port Channel	Supports bundling of 16 active ports simultaneously into a port channel on the F series module. On the M Series module, you can bundle up to 8 active and 8 standby.	5.1(1)
LACP Port-Channel MinLinks and MaxBundle	Configures LACP port-channel minlinks and LACP port-channel maxbundle.	5.1(1)
BFD	Makes network profiling and planning easier and reconvergence time consistent and predictable.	5.0(2)
Q-in-Q Tunneling	Enables the segregation of traffic for different customers while still giving you a full range of VLANs for your use.	5.0(2)
vPC and STP Convergence	Supports bringing up the vPC on a switch when its peer fails to function. Enables the vPC switch pair to appear as a single STP root in the Layer 2 topology.	5.0(2)

New and Changed Information



CHAPTER 2

Overview

This chapter provides an overview of the interface types supported by the Cisco NX-OS software.

Information 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.

Note The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

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

Table 2: Interface Parameters

Feature Name	Parameters	Further Information
Basic parameters	description, duplex, error disable, flow control, MTU, beacon	

Feature Name	Parameters	Further Information
Layer 2	Layer 2 access and trunk port settings	
	Layer 2 MAC, VLANs, private VLANs, Rapid PVST+, Multiple Spanning Tree, Spanning Tree Extensions	Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide
	Port security	Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x
Layer 3	medium, IPv4 and IPv6 addresses	
	bandwidth, delay, IP routing, VRFs	Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 6.x Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide
Port Channels	channel group, LACP	
vPCs	Virtual port channels	
Tunnels	GRE Tunneling	
Security	Dot1X, NAC, EOU, port security	Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x
FCoE	From Cisco NX-OS Release 5.2(1), you can run Fibre Channel over Ethernet (FCoE) on the Cisco Nexus 7000 Series Switches	

Ethernet Interfaces

Ethernet interfaces include access ports, trunk ports, private VLAN hosts and promiscuous ports, and routed ports.

Access Ports

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

Trunk Ports

A trunk port carries traffic for two or more VLANs. This type of port is a Layer 2 interface only.

Private VLAN Hosts and Promiscuous Ports

Private VLANs (PVLANs) provide traffic separation and security at the Layer 2 level. A PVLAN is one or more pairs of a primary VLAN and a secondary VLAN, all with the same primary VLAN. The two types of secondary VLANs are called isolated and community VLANs.

In an isolated VLAN, PVLAN hosts communicate only with hosts in the primary VLAN. In a community VLAN, PVLAN hosts communicate only among themselves and with hosts in the primary VLAN but not with hosts in isolated VLANs or in other community VLANs. Community VLANs use promiscuous ports to communicate outside the PVLAN. Regardless of the combination of isolated and community secondary VLANs, all interfaces within the primary VLAN comprise one Layer 2 domain and require only one IP subnet.

You can configure a Layer 3 VLAN network interface, or switched virtual interface (SVI), on the PVLAN promiscuous port, which provides routing functionality to the primary PVLAN.

For more information on configuring PVLAN host and PVLAN promiscuous ports and all other PVLAN configurations, see the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide

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 and does not support Layer 2 protocols, such as the Spanning Tree Protocol (STP).

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 7000 Series NX-OS Fundamentals Configuration Guide. You will also find information on configuring the IP address and default IP routing for the management interface in this document.

Port Channel Interfaces

A port channel is a logical interface that is an aggregation of multiple physical interfaces. You can bundle up to eight individual links to physical ports into a port channel to improve bandwidth and redundancy. You can also use port channeling to load balance traffic across these channeled physical interfaces.

vPCs

Virtual port channels (vPCs) allow links that are physically connected to two different Cisco Nexus 7000 series devices to appear as a single port channel by a third device. The third device can be a switch, server, or any other networking device. You can configure a total of 748 vPCs on each device. vPCs provide Layer 2 multipathing.

Subinterfaces

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

VLAN Network Interfaces

A VLAN network interface is a virtual routed interface that connects a VLAN on the device to the Layer 3 router engine on the same device. You can route across VLAN network interfaces to provide Layer 3 inter-VLAN routing.

Loopback Interfaces

A virtual 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.

Tunnel Interfaces

Tunneling allows you to encapsulate arbitrary packets inside a transport protocol. This feature is implemented as a virtual interface to provide a simple interface for configuration. The tunnel interface provides the services necessary to implement any standard point-to-point encapsulation scheme. You can configure a separate tunnel for each link.

Virtualization Interfaces

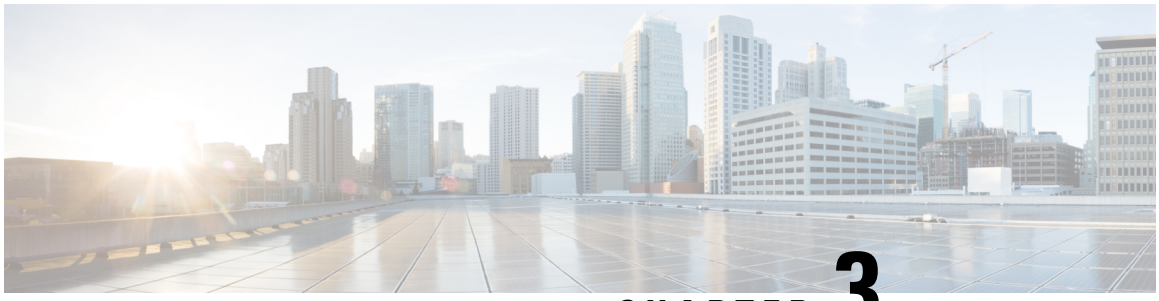
You can create multiple virtual device contexts (VDCs). Each VDC is an independent logical device to which you can allocate interfaces. Once an interface is allocated to a VDC, you can only configure that interface if you are in the correct VDC. For more information on VDCs, see the Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide.

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 after the switchover.

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*.



CHAPTER 3

Configuring Basic Interface Parameters

This chapter describes how to configure bidirectional forwarding detection.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Feature History for Configuring Basic Interface Parameters

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 3: Feature History for Configuring Basic Interface Parameters

Feature Name	Release	Feature Information
Ethernet Link OAM	8.2(3)	The **errdisable recovery cause** command is enhanced with **link-oam-dying-gasp** and **link-oam-discovery-timeout** keywords.
Debounce link up time	7.3(0)D1(1)	Added support for debounce link up time. Updated the**link debounce** {**link-up** | **time**} *milliseconds* command.
Error disabled	6.2(2)	Added the **show errdisable** {**detect** | **recovery**} command.
Display errors during policy programming.	6.2(2)	Added the **show interface status error policy** command which displays the interfaces and VLANs that produce an error during policy programming.
Clear SNMP counters from the interface	6.2(2)	Updated the **clear counters interface** command to include the **snmp** keyword that provides an option to clear SNMP values from the interface.
Interface descriptions	6.2(2)	Updated the **description** command for the increased maximum of 254 case-sensitive, alphanumeric characters.
Enhanced show output for interfaces	6.1(1)	Updated the **show interface eth** command output.
Port profiles	4.2(1)	Allows you to apply several configurations to a range of interfaces at once.
Basic interface settings	4.0(1)	These features were introduced.

Information About Basic Interface Parameters

The following sections provide information about basic interface parameters:

Interface 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 configuring this parameter for other interfaces, see the "Configuring the Interface 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.

MDIX

The medium dependent interface crossover (MDIX) parameter enables or disables the detection of a crossover connection between devices. This parameter applies only to copper interfaces. By default, this parameter is enabled.

For information about configuring the MDIX parameter, see the "Configuring the MDIX Parameter" section.

Debounce Timer

The debounce timer delays notification of a link change, which can decrease traffic loss due to network reconfiguration. You can configure the debounce timer separately for each Ethernet port and specify the delay time in milliseconds. The default value for debounce timer link down is 100 milliseconds and the default value for debounce timer link up is 0 milliseconds.

From Cisco NX-OS Release 7.3(0)D1(1), you can configure separate debounce timer values for debounce timer link down and link up. The debounce timer for link up helps in better convergence after a system reloads and avoids traffic blackholing.

Caution Enabling the debounce timer causes the link-down detections to be delayed, which results in a loss of traffic during the debounce period. This situation might affect the convergence and reconvergence of some Layer 2 and Layer 3 protocols.

For information about configuring the debounce-timer parameters, see the "Configuring the Debounce Timer" 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.

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

From Cisco NX-OS Release 8.2(3) the **link-oam-dying-gasp** and the l**link-oam-discovery-timeout** options under the **errdisable recovery cause** command enables to recover the Ethernet link OAM.

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.

From Cisco NX-OS Release 6.2(2), you can use the **show errdisable recovery** and **show errdisable detect** commands to display the errdisable recovery and detection runtime information.

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, such as a Layer 2 port mode change from access to trunk, which can be ingress, egress or bi-directional, is defined through the command line interface.

Policies are pushed when you configure a policy on an interface, if an interface VLAN membership changes or when the line card boots up, all the configured policies get pushed simultaneously. To ensure that the policies that are pushed are consistenet with hardware policiesThe is used to ensure that policies that are pushed are consistent with the hardware policies and that they display the interfaces and VLANs that have errors during the policy programming, enter the **show interface status error policy** command.

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.

Rate Mode

On a 32-port, 10-Gigabit Ethernet module, each set of four ports can handle 10 Gb/s of bandwidth. You can use the rate-mode parameter to dedicate that bandwidth to the first port in the set of four ports or share the bandwidth across all four ports.

The table below identifies the ports that are grouped together to share each 10 Gb/s of bandwidth and which port in the group can be dedicated to use the entire bandwidth.

Table 4: Dedicated and Shared Ports

Ports Groups that Can Share Bandwidth	Ports that Can be Dedicated to Each 10-Gigabit Ethernet of Bandwidth
1, 3, 5, 7	1
2, 4, 6, 8	2
9, 11, 13, 15	9
10, 12, 14, 16	10

Ports Groups that Can Share Bandwidth	Ports that Can be Dedicated to Each 10-Gigabit Ethernet of Bandwidth
17, 19, 21, 23	17
18, 20, 22, 24	18
25, 27, 29, 31	25
26, 28, 30, 32	26

Note All ports in each port group must be part of the same virtual device context (VDC). For more information on VDCs, see the Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide.

Speed Mode and Duplex Mode

The speed mode and duplex mode are interrelated for each Ethernet and management interface. By default, each of these interfaces autonegotiates its speed and duplex mode with the other interface, but you can change these settings. If you change the settings, be sure to use the same speed and duplex mode setting on both interfaces, or use autonegotiation for at least one of the interfaces.

For information about setting the speed and duplex speed for other interfaces, see the "Configuring the Interface Speed and Duplex Mode" section.

Flow Control

When the receive buffer for an Ethernet port that runs 1 Gb/s or faster fills, flow control enables that port to send an IEEE 802.3x pause frame to the transmitting port to request it to stop transmitting data for a specified amount of time. Transmitting ports, running at any speed, can receive the pause frames to stop their transmission of data.

To allow flow control to work between two ports, you must set the corresponding receive and send flow control parameters for both ports as enabled or desired. When you set the parameter to enabled, the send or receive flow-control function is activated regardless of the setting of the other port. When you set the parameter to desired, the send or receive flow-control function is activated if you set the corresponding flow-control state of the other port to enabled or desired. If you set one of the flow control states to disabled, flow control is disabled for that transmission direction. To see how the different port flow-control states affect the link flow-control state, see the table below.

Table 5: Port Flow Control Influences on Link Flow Control

Port Flow Control States		Link Flow Control State
Port Receiving Data (Sends Pause Frames)	Port Transmitting Data (Receives Pause Frames)	
Enabled	Enabled	Enabled
Enabled	Desired	Enabled

Port Flow Control States		Link Flow Control State
Port Receiving Data (Sends Pause Frames)	Port Transmitting Data (Receives Pause Frames)	
Enabled	Disabled	Disabled
Desired	Enabled	Enabled
Desired	Desired	Enabled
Desired	Disabled	Disabled
Disabled	Enabled	Disabled
Disabled	Desired	Disabled
Disabled	Disabled	Disabled

Port 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, each port 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 with less overhead. The larger frames, called jumbo frames, can be up to 9216 bytes in size, which is also the default system jumbo MTU size.

On a Layer 3 interface, you can configure an MTU size between 576 and 9216 bytes. You can configure up to 64 MTU settings for each I/O module.

Note The global LAN port MTU size applies to the traffic through a Layer 3 Ethernet LAN port that is configured with a nondefault MTU size.

For a Layer 2 port, you can configure an MTU size that is either the system default (1500 bytes) or the system jumbo MTU size (initially 9216 bytes).

Note If you change the system jumbo MTU size, Layer 2 ports automatically use the system default MTU size (1500 bytes) unless you specify the new system jumbo MTU size for some or all of those ports.

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

Bandwidth

Ethernet ports have a fixed bandwidth of 1,000,000 Kb at the physical level. 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 level. For

example, the Interior Gateway Routing Protocol (IGRP) uses the minimum path bandwidth to determine a routing metric, but the bandwidth at the physical level remains at 1,000,000 Kb.

For information about configuring the bandwidth parameter for other interfaces, see the "Configuring Bandwidth for Ethernet Interfaces" 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 throughput-delay parameter for other interfaces, see the "Configuring 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 other interfaces, see the "Shutting Down and Activating an 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, including spanning tree topology loops.

UDLD is a Layer 2 protocol that works with the Layer 1 protocols to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected LAN ports. When you enable both autonegotiation and UDLD, Layer 1 and Layer2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

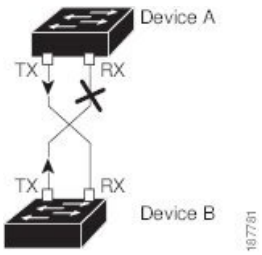
A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does 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 at Layer 2 determines whether those fibers are connected correctly and whether traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by autonegotiation, because autonegotiation operates at Layer 1.

The Cisco Nexus 7000 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 below 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

Table 6: 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 UDLD Mode" section.

UDLD Aggressive and Nonaggressive Modes

UDLD aggressive mode is disabled by default. You can configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. If UDLD aggressive mode is enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops

receiving UDLD frame, UDLD tries to reestablish the connection with the neighbor. After eight failed retries, the port is disabled.

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

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

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

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

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.

Carrier Delay

Note You can configure the carrier delay timer only on VLAN network interfaces. The timer cannot be configured on physical Ethernet interfaces, port channels, and loopback interfaces.

If a link goes down and comes back up before the carrier delay timer expires, the down state is effectively filtered, and the rest of the software on the device is not aware that a link-down event occurred. A large carrier delay timer results in fewer link-up/link-down events being detected. When you set the carrier delay time to 0, the device detects each link-up/link-down event that occurs.

In most environments, a lower carrier delay time is better than a higher one. The exact value that you choose depends on the nature of the link outages and how long you expect these linkages to last in your network. If your data links are subject to short outages (especially if those outages last less time than it takes for your IP routing to converge), you should set a long carrier delay value to prevent these short outages from causing unnecessary problems in your routing tables. However, if your outages tend to be longer, you might want to set a shorter carrier delay time so that the outages are detected sooner, and the IP route convergence begins and ends sooner.

The default carrier-delay time is 100 milliseconds.

Port Channel Parameters

A port channel is an aggregation of physical interfaces that comprise a logical interface. You can bundle up to eight 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 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.

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

Port Profiles

From Cisco NX-OS Release 4.2(1) for the Cisco Nexus 7000 Series devices, you can create a port profile that contains many interface commands and apply that port profile to a range of interfaces. Each port profile can be applied only to a specific type of interface; the choices are as follows:

- Ethernet
- VLAN network interface
- Loopback
- Port channel
- Tunnel

When you choose Ethernet or port channel as the interface type, the port profile is in the default mode which is Layer 3. Enter the **switchport** command to change the port profile to Layer 2 mode.

You inherit the port profile when you attach the port profile to an interface or range of interfaces. When you attach, or inherit, a port profile to an interface or range of interfaces, the system applies all the commands in that port profile to the interfaces. Additionally, you can have one port profile inherit the settings from another port profile. Inheriting another port profile allows the initial port profile to assume all of the commands of the second, inherited, port profile that do not conflict with the initial port profile.Four levels of inheritance are supported. The same port profile can be inherited by any number of port profiles.

The system applies the commands inherited by the interface or range of interfaces according to the following guidelines:

- Commands that you enter under the interface mode take precedence over the port profile's commands if there is a conflict. However, the port profile retains that command in the port profile.
- The port profile's commands take precedence over the default commands on the interface, unless the port-profile command is explicitly overridden by the default command.
- When a range of interfaces inherits a second port profile, the commands of the initial port profile override the commands of the second port profile if there is a conflict.
- After you inherit a port profile onto an interface or range of interfaces, you can override individual configuration values by entering the new value at the interface configuration level. If you remove the

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

- There are no default configurations associated with a port profile.

A subset of commands are available under the port-profile configuration mode, depending on which interface type you specify.

Note You cannot use port profiles with Session Manager. See the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide* http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus7000/sw/system-management/config/cisco_nexus7000_system-management_config_guide_8x.html for information about Session Manager.

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

If you inherit one or more port profiles onto an original port profile, only the last inherited port profile must be enabled; the system assumes that the underlying port profiles are enabled.

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

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

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

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

Just as in the device, you can enter a configuration for an object in port profiles without that object being applied to interfaces yet. For example, you can configure a virtual routing and forward (VRF) instance without it being applied to the system. If you then delete that VRF and related configurations from the port profile, the system is unaffected.

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

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

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

Time Domain Reflectometry Cable Diagnostics

From Cisco NX-OS Release 5.0(2) for the Cisco Nexus 7000 Series devices and the introduction of the latest generation of line cards, you can perform cable diagnostics without the use of expensive third-party equipment. With the cable diagnostic capabilities embedded directly in the line cards, you no longer need to unplug cables and connect cable testers to diagnose a link fault. Each port on the line card can independently detect cabling issues and report them to the switch software using Time Domain Reflectometry (TDR).

You can use TDR to analyze a conductor by transmitting a pulsed waveform signal into it and then examine the polarity, amplitude, and round-trip time of the reflected waveform.

By estimating the speed of propagation of the signal in the cable and by measuring the time it takes for its reflection to travel back to the source, it is possible to measure the distance to the reflecting point. Also, by comparing the polarity and amplitude of the original pulse with its reflection, it is possible to distinguish between different types of faults, such as open or shorted pairs.

Being able to remotely diagnose a cable failure, you can now identify the root cause of a problem more quickly and more effectively, providing your users with a prompt response to connectivity issues.

Default Settings for Basic Interfaces Parameters

Table 7: Default Basic Interface Parameter Settings

Parameter	Default
Description	Blank
Beacon	Disabled
Debounce timer link down	Enabled. 100 milliseconds
Debounce timer link up	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
UDLD per-port enable state for copper media	Disabled on all Ethernet 10/100 and 1000BASE-TX LAN ports
UDLD message interval	Disabled
UDLD aggressive mode	Disabled

Parameter	Default
Carrier delay	100 milliseconds
Error disable	Disabled
Error disable recovery	Disabled
Error disable recovery interval	300 seconds
Link debounce	Enabled
Port profile	Disabled

Guidelines and Limitations for Basic Interfaces Parameters

Basic interface parameters have the following configuration guidelines and limitations:

- 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.
- When configuring flow control for a local port, consider the following:
  - To receive pause frames when you do not know how the remote port send parameter is configured, set the local port receive parameter to desired.
  - To receive pause frames when you know that the remote port send parameter is enabled or desired, set the local port receive parameter to enabled.
  - To ignore received pause frames, set the local port receive parameter to disabled.
  - To send pause frames when you do not know how the remote port receive parameter is configured, set the local port send parameter to desired.
  - To send pause frames when you know that the remote port receive parameter is enabled or desired, set the local port send parameter to enabled.
  - To prevent the sending of pause frames, set the local port send parameter to disabled.
- 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 Speed Mode and Duplex Mode, on page 15 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, 10, 100, or 1000 Mb/s), you must configure the connecting port to match. Do not configure the connecting port to negotiate the speed.

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.

- Debounce timer link up is supported only on F3-Series Line cards.

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

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

Table 8: 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.

Note When you are in the interface configuration mode, the commands that you enter configure the interface that you specified for this mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface*	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.” **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.”

Configuring the Interface Description

You can provide textual interface descriptions for the Ethernet and management interfaces. Descriptions can be a maximum of 254 case-sensitive, alphanumeric characters.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *interface*	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.”
Step 3	switch(config-if)# **description** *text*	Specifies the description for the interface. The description is a maximum of 254 characters.

	Command or Action	Purpose
Step 4	switch(config-if)# **exit**	Exits the interface mode.
Step 5	(Optional) switch(config)# **show interface** *interface*	Displays the interface status, which includes the description parameter.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	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)#
```

From Cisco NX-OS Release 6.1, the output of the **show interface eth** command is enhanced as shown in the following example:

```
switch# show interface eth 2/1
Ethernet2/1 is down (SFP not inserted)
admin state is down, Dedicated Interface
Hardware: 1000 Ethernet, address: 0026.9814.0ec1 (bia f866.f23e.0de8)
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, medium is broadcast
auto-duplex, auto-speed
Beacon is turned off
Auto-Negotiation is turned on
Input flow-control is off, output flow-control is off
Auto-mdix is turned off
Switchport monitor is off
EtherType is 0x8100
EEE (efficient-ethernet) : n/a
Last link flapped never
Last clearing of "show interface" counters never
0 interface resets
30 seconds input rate 0 bits/sec, 0 packets/sec
30 seconds output rate 0 bits/sec, 0 packets/sec
```

Configuring the Beacon Mode

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

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# {**beacon** | **no beacon**}	Enables the beacon mode or disables the beacon mode. The default mode is disabled.
Step 4	(Optional) switch(config)# **show interface ethernet** *slot/port*	Displays the interface status, which includes the beacon mode state.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	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
```

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
```

Changing the Bandwidth Rate Mode

You can specify whether each 10 Gb of bandwidth on a 32-port 10-Gigabit Ethernet module is dedicated to one port or shared by four ports in the same port group.

Dedicating Bandwidth to One Port

When you dedicate the bandwidth to one port, you must first administratively shut down the four ports in the group, change the rate mode to dedicated, and then bring the dedicated port administratively up.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*, **ethernet** *slot/port*, **ethernet** *slot/port*, **ethernet** *slot/port*	Specifies an Ethernet interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **shutdown**	Administratively shuts down the ports.
Step 4	switch(config)# **interface ethernet** *slot/port*	Specifies the first Ethernet interface in a group of interfaces.

	Command or Action	Purpose
Step 5	switch(config-if)# **rate-mode dedicated**	Dedicates the full bandwidth of 10 Gb to one port. When you dedicate the bandwidth, all subsequent commands for the port are for dedicated mode.
Step 6	switch(config-if)# **no shutdown**	Brings the port administratively up.
Step 7	(Optional) switch(config-if)# **show interface ethernet** *slot/port* **capabilities**	Displays the interface information including the current rate mode.
Step 8	switch(config-if)# **exit**	Exits the interface mode.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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)# rate-mode dedicated
switch(config-if)# no shutdown
```

Sharing the Bandwidth Among a Port Group

You can share 10 Gb of bandwidth among a group of ports (four ports) on a 32-port, 10-Gigabit Ethernet module. To share the bandwidth, you must bring the dedicated port administratively down, specify the ports that are to share the bandwidth, change the rate mode to shared, and then bring the ports administratively up.

Before you begin

All ports in the same group must belong to the same VDC.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies the first Ethernet interface in a group of interfaces.
Step 3	switch(config-if)# **shutdown**	Administratively shuts down the ports.

	Command or Action	Purpose
Step 4	switch(config)# **interface ethernet** *slot/port*, **ethernet** *slot/port*, **ethernet** *slot/port*, **ethernet** *slot/port*	Specifies four Ethernet interfaces to configure (they must be part of the same port group), and enters interface configuration mode.
Step 5	switch(config-if)# **rate-mode shared**	Sets the shared rate mode for the specified ports.
Step 6	switch(config-if)# **no shutdown**	Brings the ports administratively up.
Step 7	(Optional) switch(config-if)# **show interface ethernet** *slot/port*	Displays the interface information including the current rate mode.
Step 8	switch(config-if)# **exit**	Exits the interface mode.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the shared 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
switch(config-if)# shutdown
switch(config)# interface ethernet 4/17, ethernet 4/19, ethernet 4/21, ethernet 4/23
switch(config-if)# rate-mode shared
switch(config-if)# no shutdown
```

Configuring the Error-Disabled State

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

Enabling 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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **errdisable detect cause** {**acl-exception** | **all** | **link-flap** | **loopback**}	Specifies a condition under which to place the interface in an error-disabled state. The default is enabled.

	Command or Action	Purpose
Step 3	switch(config)# **shutdown**	Brings the interface down administratively. To manually recover the interface from the error-disabled state, enter this command first.
Step 4	switch(config)# **no shutdown**	Brings the interface up administratively and enables the interface to recover manually from the error-disabled state.
Step 5	(Optional) switch(config)# **show interface status err-disabled**	Displays information about error-disabled interfaces.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	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
```

Enabling 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).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **errdisable recovery cause** {**all** | **bpduguard** | **failed-port-state** |**link-flap** | **loopack** | **miscabling** | **psecure-violation** | **security-violation** | **storm-control** | **udld** | **vpc peerlink** |**link-oam-discovery-timeout** |**link-oam-dying-gasp**}	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.
Step 3	(Optional) switch(config)# **show interface status err-disabled**	Displays information about error-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	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
```

Configuring the Error-Disabled Recovery Interval

You can configure the error-disabled recovery timer value.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **errdisable recovery interval** *interval*	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	(Optional) switch(config)# **show interface status err-disabled**	Displays information about error-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	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
```

Configuring the MDIX Parameter

If you need to detect the type of connection (crossover or straight) with another copper Ethernet port, enable the medium dependent independent crossover (MDIX) parameter for the local port. By default, this parameter is enabled.

Before you begin

You must enable MDIX for the remote port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.

	Command or Action	Purpose
Step 3	switch(config-if)# {**mdix auto** | **no mdix**}	Specifies whether to enable or disable MDIX detection for the port.
Step 4	(Optional) switch(config-if)# **show interface ethernet** *slot/port* **capabilities**	Displays the interface status, which includes the MDIX status.
Step 5	switch(config-if)# **exit**	Exits the interface mode.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable MDIX for Ethernet port 3/1:

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

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

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

Configuring the Debounce Timer

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

You can show the debounce times for all of the Ethernet ports by using the **show interface debounce** command.

Procedure

Step 1 switch# **configure terminal**

Enters the global configuration mode.

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

Specifies an interface to configure, and enters interface configuration mode.

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

Enables the debounce link-up or down timer for the amount of time (0 to 5000 ms) specified. The default value for **link debounce link-up** command is 0. We recommended setting the debounce link-up timer value to 100 ms.

Note The **link debounce** command without the **link-up** keyword refers to link down debounce time.

Note The **link debounce link-up** command will override all previous values configured by the user.

Disables the debounce timer if you specify 0 milliseconds.

Step 4 switch(config-if)# **exit**

Exits the interface mode.

Step 5 (Optional) switch(config)# **show interface debounce**

Shows the link debounce time for all of the Ethernet interfaces.

Step 6 (Optional) switch(config)# **copy running-config startup-config**

Copies the running configuration to the startup configuration.

Example

This example shows how to enable the link down debounce timer to 1000 ms for the Ethernet port 3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# link debounce time 1000
```

This example shows how to enable the debounce link-up timer and set the debounce time to 1000 ms for the Ethernet port 3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# link debounce link-up time 1000
```

This example shows how to disable the debounce timer for the Ethernet port 3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# link debounce time 0
```

Configuring the Interface Speed and Duplex Mode

The interface speed and duplex mode are interrelated, so you should configure both of their parameters at the same time.

To see which speeds and duplex modes you can configure together for Ethernet and management interfaces, see Speed Mode and Duplex Mode, on page 15.

Note The interface speed that you specify can affect the duplex mode used for an interface, so you should set the speed before setting the duplex mode. If you set the speed for autonegotiation, the duplex mode is automatically set to be autonegotiated. If you specify 10- or 100-Mb/s speed, the port is automatically configured to use half-duplex mode, but you can specify full-duplex mode instead. If you specify a speed of 1000 Mb/s (1 Gb/s) or faster, full duplex is automatically used.

Before you begin

Make sure that the remote port has a speed setting that supports your changes for the local port. If you want to set the local port to use a specific speed, you must set the remote port for the same speed or set the local port to autonegotiate the speed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *interface*	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."
Step 3	switch(config-if)# **speed** {{**10** | **100** | **1000** | {**auto** [**10 100** [**1000**]]}} | {**10000** | **auto**}}	For Ethernet ports on the 48-port 10/100/1000 modules, sets the speed at 10 Mb/s, 100 Mb/s, or 1000 Mb/s, or sets the port to autonegotiate its speed with the other 10/100/1000 port on the same link. For Ethernet ports on the 32-port 10-Gigabit Ethernet modules, sets the speed at 10,000 Mb/s (10 Gb/s) or sets the port to autonegotiate its speed with the other 10-Gigabit Ethernet port on the link. For management interfaces, sets the speed as 1000 Mb/s or sets the port to autonegotiate its speed.
Step 4	switch(config-if)# **duplex** {**full** | **half** | **auto**}	Specifies the duplex mode as full, half, or autonegotiate.
Step 5	switch(config-if)# **exit**	Exits the interface mode.
Step 6	(Optional) switch(config)# **show interface** *interface*	Displays the interface status, which includes the speed and duplex mode parameters.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to set the speed of Ethernet port 1 on the 48-port, 10/100/1000 module in slot 3 to 1000 Mb/s and full-duplex mode:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# speed 1000
switch(config-if)# duplex full
switch(config-if)#
```

Configuring Flow Control

For Ethernet ports that run at 1 Gb/s or faster, you can enable or disable the port's ability to send and receive flow-control pause frames. For Ethernet ports that run slower than 1 Gb/s, you can enable or disable only the port's ability to receive pause frames.

When enabling flow control for the local port, you either fully enable the local port to send or receive frames regardless of the flow-control setting of the remote port, or you set the local port to use the desired setting used by the remote port. If you enable both the local and remote port for flow control, or set the desired flow control of the other port, or set a combination of those two states, flow control is enabled for those ports.

Note For ports that run at 10 Gb/s, you cannot use the desired state for the send or receive parameter. The **desired** keyword in the **flowcontrol** command is not supported for higher port speeds such as 40 Gb/s and 100 Gb/s.

Before you begin

Make sure that the remote port has the corresponding setting for the flow control that you need. If you want the local port to send flow-control pause frames, make sure that the remote port has a receive parameter set to on or desired. If you want the local port to receive flow-control frames, make sure that the remote port has a send parameter set to on or desired. If you do not want to use flow control, you can set the remote port's send and receive parameters to off.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an Ethernet interface to configure by its slot number and port number, and enters the interface configuration mode.
Step 3	switch(config-if)# **flowcontrol** {**send** | **receive**} {**desired** | **on** | **off**}	Specifies the flow-control setting for ports. You can set the send setting for only the ports running at 1000 Mb/s or faster. You can set the receive setting for ports running at any speed.
Step 4	switch(config-if)# **exit**	Exits the interface mode.

	Command or Action	Purpose
Step 5	(Optional) switch(config)# **show interface ethernet** *slot/port*	Displays the interface status, which includes the flow control parameters.
Step 6	(Optional) switch(config)# **show interface flowcontrol**	Displays the flow control status for all Ethernet ports.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to set Ethernet port 3/1 to send flow control pause frames:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# flowcontrol send on
```

Configuring MTU Size

You can configure the maximum transmission unit (MTU) size for Layer 2 and Layer 3 Ethernet interfaces. For Layer 3 interfaces, you can configure the MTU to be between 576 and 9216 bytes (even values are required). For Layer 2 interfaces, you can configure the MTU to be either the system default MTU (1500 bytes) or the system jumbo MTU size (which has the default size of 9216 bytes).

Note You can change the system jumbo MTU size, but if you change that value, the Layer 2 interfaces that use that value automatically changes to the new system jumbo MTU value.

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.

After changing the port mode to Layer 2, you can return to configuring Layer 3 interfaces by changing the port mode again, by using the **no switchport** command.

Configuring Interface MTU Size

For Layer 3 interfaces, you can configure an MTU size that is between 576 and 9216 bytes.

For Layer 2 interfaces, you can configure all Layer 2 interfaces to use either the default MTU size (1500 bytes) or the system jumbo MTU size (default size of 9216 bytes).

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an Ethernet interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# {**switchport** | **no switchport**}	Specifies to use Layer 2 or Layer 3.
Step 4	switch(config-if)# **mtu** *size*	For a Layer 2 interface, specifies either the default MTU size (1500) or the system jumbo MTU size (9216 unless you have changed the system jumbo MTU size). For a Layer 3 interface, specifies any even number between 576 and 9216.
Step 5	switch(config-if)# **exit**	Exits the interface mode.
Step 6	(Optional) switch(config)# **show interface ethernet** *slot/port*	Displays the interface status, which includes the MTU size.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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
```

Configuring System Jumbo MTU Size

To configure jumbo MTU size for a Layer 2 interface and Layer 2 interfaces and subinterfaces, perform the following task. If you do not configure the system jumbo MTU size, it defaults to 9216 bytes.

When you configure jumbo MTU on a port-channel subinterface you must first enable MTU 9216 on the base interface and then configure it again on the subinterface. If you enable the jumbo MTU on the subinterface before you enable it on the base interface then the following error will be displayed on the console:

```
switch(config)# int po 502.4
switch(config-subif)# mtu 9216
ERROR: Incompatible MTU values
```

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **system jumbomtu** *size*	Specifies the system jumbo MTU size. Use an even number between 1500 and 9216.
Step 3	(Optional) switch(config)# **show running-config all**	Displays the current operating configuration, which includes the system jumbo MTU size.
Step 4	switch(config)# **interface type** *slot/port*	Specifies an interface to configure and enters interface configuration mode. When enabling a port-channel sub interface for Jumbo MTU, first enable the base interface with 'mtu 9216' and then configure each sub interface that supports the MTU size with the 'mtu 9216'. If performed in the incorrect order, Jumbo MTU support will not be enabled.
Step 5	switch(config-if)# **mtu** *size*	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. **Note** To enable jumbo MTU for Layer 3 port-channel subinterfaces, you must first enable the base (parent) interface using the **mtu 9216** command and then configure the **mtu 9216** command for each subinterface on which this MTU size is to be supported. If you configure the commands in the reverse order (subinterface first and then the base interface), the following message is displayed: `Incompatible MTU values`.
Step 6	switch(config-if)# **exit**	Exits the interface mode.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	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 8000
```

Configuring Bandwidth for Ethernet Interfaces

You can configure the bandwidth for Ethernet interfaces. The physical level uses an unchangeable bandwidth of 1 GB, but you can configure a value of 1 to 10,000,000 Kb for Level 3 protocols.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an Ethernet interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **bandwidth** *kbps*	Specifies the bandwidth as an informational-only value between 1 and 10,000,000.
Step 4	switch(config-if)# **exit**	Exits the interface mode.
Step 5	(Optional) switch(config)# **show interface ethernet** *slot/port*	Displays the interface status, which includes the bandwidth value.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	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
```

Configuring 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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an Ethernet interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **delay** *value*	Specifies the delay time in tens of microseconds. You can set an informational value range between 1 and 16777215 tens of microseconds.
Step 4	switch(config-if)# **exit**	Exits the interface mode.
Step 5	(Optional) switch(config)# **show interface ethernet** *slot/port*	Displays the interface status, which includes the throughput-delay time.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	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
```

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

Shutting Down and Activating an 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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *interface*	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.”
Step 3	switch(config-if)# **shutdown**	Disables the interface.
Step 4	(Optional) switch(config-if)# **show interface** *interface*	Displays the interface status, which includes the administrative status.
Step 5	switch(config-if)# **no shutdown**	Reenables the interface.
Step 6	(Optional) switch(config-if)# **show interface** *interface*	Displays the interface status, which includes the administrative status.
Step 7	switch(config-if)# **exit**	Exits the interface mode.
Step 8	(Optional) switch(config)# **copy running-config startup-config**	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
```

Configuring UDLD Mode

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

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

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

Description	Fiber port	Copper or Nonfiber port
Disable UDLD command	**udld disable**	**no udld enable**

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

By default, UDLD is disabled for the 48-port, 10/100/1000 Ethernet module ports but the normal UDLD mode is enabled for the 32-port, 10-Gigabit Ethernet module ports.

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **feature udld**	Enables UDLD for the device. The **no feature udld** disables UDLD for the device.
Step 3	(Optional) switch(config)# **udld message-time** *seconds*	Specifies the interval between sending UDLD messages. The range is from 7 to 90 seconds, and the default is 15 seconds. **Note** The interface level timer changes only if bidirectional UDLD status is detected, otherwise the timer remains at 7 seconds and cannot be changed.
Step 4	(Optional) switch(config)# **udld aggressive**	Specifies UDLD mode to be aggressive. **Note** For copper interfaces, you enter the interface command mode for those interfaces you want to configure for UDLD aggressive mode and issue this command in interface command model.
Step 5	(Optional) switch(config)# **interface ethernet** *slot/port*	Specifies an Ethernet interface to configure, and enters interface configuration mode.
Step 6	(Optional) switch(config-if)# **udld** {**enable** | **disable**}	Enables UDLD on the specified copper port or disables UDLD on the specified fiber port. To enable UDLD on copper ports enter the **udld enable** command. To enable UDLD on fiber ports, enter the **no udld disable** command. See the table above for more details.

	Command or Action	Purpose
Step 7	switch(config-if)# **exit**	Exits the interface mode.
Step 8	(Optional) switch(config)# **show udld** [**ethernet** *slot/port* | **global** | **neighbors**]	Displays the UDLD status.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	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
```

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
```

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

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

This example shows how to enable the aggressive UDLD mode for the copper interface Ethernet 3/1:

```
switch# configure terminal
switch(config)# feature udld
switch(config)# udld aggressive
switch(config)# interface ethernet 3/1
switch(config-if-range)# udld enable
```

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
```

Configuring Carrier Delay Timer

The carrier delay timer sets a time during which all link-down/link-up events are not detected by any of the other software on the device. When you configure a longer carrier delay time, fewer link-down/link-up events are recorded. When you configure the carrier delay time to 0, the device detects each link-down/link-up event.

Note You can configure the carrier delay timer only on VLAN network interfaces; you cannot configure this timer in any other interface modes.

Before you begin

Ensure that you are in VLAN interface mode. You cannot configure the carrier delay timer in any other interface mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface vlan** *vlan-id*	Enters the VLAN interface mode.
Step 3	switch(config-if)# **carrier-delay** {**sec** | **msec** **number**}	Sets the carrier delay timer. You can set the time between 0 to 60 seconds or 0 to 1000 milliseconds. The default is 100 milliseconds.
Step 4	switch(config-if)# **exit**	Exits the interface mode.
Step 5	(Optional) switch(config)# **show interface** *vlan-id*	Displays the interface status.
Step 6	(Optional) switch(config)# **copy** **running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to set the carrier delay timer to 20 seconds for VLAN 5:

```
switch# configure terminal
switch(config)# interface vlan 5
switch(config-if)# carrier-delay 20
```

Configuring Port Profiles

You can apply several configuration parameters to a range of interfaces simultaneously. All the interfaces in the range must be the same type. You can also inherit the configurations from one port profile into another port profile. The system supports four levels of inheritance.

Creating a Port Profile

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **interface-vlan** | **loopback** | **port channel** | **tunnel**}] *name*	Creates and names a port profile for the specified type of interface and enters the port-profile configuration mode.
Step 3	switch(config-ppm)# **exit**	Exits the port-profile configuration mode.
Step 4	(Optional) switch(config)# **show port-profile**	Displays the port-profile configuration.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

```
switch# configure terminal
switch(config)# port-profile type tunnel test
switch(config-ppm)#
```

Entering Port-Profile Configuration Mode and Modifying a Port Profile

You can enter the port-profile configuration mode and modify a port profile. To modify the port profile, you must be in the port-profile configuration mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **interface-vlan** | **loopback** | **port channel** | **tunnel**}] *name*	Enters the port-profile configuration mode for the specified port profile and allows you to add or remove configurations to the profile.
Step 3	switch(config-ppm)# **exit**	Exits the port-profile configuration mode.
Step 4	(Optional) switch(config)# **show port-profile**	Displays the port-profile configuration.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enter the port-profile configuration mode for the specified port profile and bring all the interfaces administratively up:

```
switch# configure terminal
switch(config)# port-profile type tunnel test
switch(config-ppm)# no shutdown
switch(config-ppm)#
```

Assigning a Port Profile to a Range of Interfaces

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** [**ethernet** *slot/port* | **interface-vlan** *vlan-id* | **loopback** *number* | **port-channel** *number* | **tunnel** *number*]	Selects the range of interfaces.
Step 3	switch(config-if)# **inherit port-profile** *name*	Assigns the specified port profile to the selected interfaces.
Step 4	switch(config-ppm)# **exit**	Exits the port-profile configuration mode.
Step 5	(Optional) switch(config)# **show port-profile**	Displays the port-profile configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to assign the port profile named adam to Ethernet interfaces 7/3 to 7/5, 10/2, and 11/20 to 11/25:

```
switch# configure terminal
switch(config)# interface ethernet7/3-5, ethernet10/2, ethernet11/20-25
switch(config-if)# inherit port-profile adam
switch(config-if)#
```

Enabling a Specific Port Profile

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

If you inherit one or more port profiles onto an original port profile, only the last inherited port profile must be enabled; the system assumes that the underlying port profiles are enabled.

You must be in the port-profile configuration mode to enable or disable port profiles.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **interface-vlan** | **loopback** | **port channel** | **tunnel**}] *name*	Creates and names a port profile for the specified type of interface and enters the port-profile configuration mode.
Step 3	switch(config-ppm)# **state enabled**	Enables that port profile.
Step 4	switch(config-ppm)# **exit**	Exits the port-profile configuration mode.
Step 5	(Optional) switch(config)# **show port-profile**	Displays the port-profile configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

```
switch# configure terminal
switch(config)# port-profile type tunnel test
switch(config-ppm)# state enabled
switch(config-ppm)#
```

Inheriting a Port Profile

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)#**port-profile** *name*	Enters the port-profile configuration mode for the specified port profile.
Step 3	switch(config-ppm)# **inherit port-profile** *name*	Inherits another port profile onto the existing one. The original port profile assumes all the configurations of the inherited port profile.
Step 4	switch(config-ppm)# **exit**	Exits the port-profile configuration mode.

	Command or Action	Purpose
Step 5	(Optional) switch(config)# **show port-profile**	Displays the port-profile configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

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

Removing a Port Profile from a Range of Interfaces

You can remove a port profile from some or all of the interfaces to which you have applied the profile. You do this configuration in the interfaces configuration mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** [**ethernet** *slot/port* | **interface-vlan** *vlan-id* | **loopback** *number* | **port-channel** *number* | **tunnel** *number*]	Selects the range of interfaces.
Step 3	switch(config-if)# **inherit port-profile** *name*	Assigns the specified port profile to the selected interfaces.
Step 4	switch(config-if)# **exit**	Exits the port-profile configuration mode.
Step 5	(Optional) switch(config)# **show port-profile**	Displays the port-profile configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to assign the port profile named adam to Ethernet interfaces 7/3 to 7/5, 10/2, and 11/20 to 11/25:

```
switch# configure terminal
switch(config)# interface ethernet 7/3-5, 10/2, 11/20-25
switch(config-if)# inherit port-profile adam
switch(config-if)#
```

Removing an Inherited Port Profile

You can remove an inherited port profile. You do this configuration in the port-profile mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **port-profile** *name*	Enters the port-profile configuration mode for the specified port profile.
Step 3	switch(config-ppm)# **inherit port-profile** *name*	Removes an inherited port profile from this port profile.
Step 4	switch(config-ppm)# **exit**	Exits the port-profile configuration mode.
Step 5	(Optional) switch(config)# **show port-profile**	Displays the port-profile configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

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

Performing TDR Cable Diagnostics

You can perform cable diagnostics without the use of expensive third-party equipment. Each port on the line card can independently detect cabling issues and report them to the switch software using TDR diagnostics.

Before you begin

The TDR test guidelines are as follows:

- TDR can test cables up to a maximum length of 115 meters.
- Do not start the test at the same time on both ends of the cable. Starting the test at both ends of the cable at the same time can lead to false test results.
- Do not change the port configuration during any cable diagnostics test. This action may result in incorrect test results.
- All ports in the associated port group must be shut down before running the TDR test.

Procedure

	Command or Action	Purpose
Step 1	switch# **test cable-diagnostics tdr interface** *number*	Starts the TDR test on the specified interface. You must have previously run the shutdown command on the interface.
Step 2	(Optional) switch# **show interface** *number* **cable-diagnostics-tdr**	Shows the TDR test results for the specified interface.

Example

This example shows how to perform a TDR test on a specific interface. In this example, ethernet 3/1 has a missing cable, and ethernet 3/12 is a good cable and connection.

```
switch(config)# interface ethernet 3/1-12
switch(config-if-range)# shutdown
switch# test cable-diagnostics tdr interface ethernet 3/1
switch# test cable-diagnostics tdr interface ethernet 3/12
switch# show interface ethernet 3/1 cable-diagnostics-tdr


------------------------------------------------------------------------------
Interface     Speed  Pair Cable Length   Distance to fault Channel   Pair Status
------------- ----- ---- -------------- ------------------- ------- -----------
Eth3/1        auto    ---     N/A        1 +/- 2 m           Pair A   Open
              auto    ---     N/A        1 +/- 2 m           Pair B   Open
              auto    ---     N/A        1 +/- 2 m           Pair C   Open
              auto    ---     N/A        1 +/- 2 m           Pair D   Open

n7000# show interface ethernet 3/12 cable-diagnostics-tdr
------------------------------------------------------------------------------
Interface     Speed  Pair Cable Length   Distance to fault Channel   Pair Status
------------- ----- ---- -------------- ------------------- ------- -----------
Eth3/12         1000  ---  N/A                N/A            Pair A  Terminated
                      ---  N/A                N/A            Pair B  Terminated
                      ---  N/A                N/A            Pair C  Terminated
                      ---  N/A                N/A            Pair D  Terminated
```

Configuring Rate Limits for Packets that Reach the Supervisor

From Cisco NX-OS Release 5.1, you can configure rate limits globally on the device for packets that reach the supervisor module. For more information, see the Cisco Nexus 7000 Series NX-OS Security Configuration Guide.

You can configure rate limits for packets that reach the supervisor module on a particular interface.

Note If the rate of incoming or outgoing packets exceeds the configured rate limit, the device logs a system message, but does not drop any packets.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# [**no**] **rate-limit cpu direction** {**input** | **output** | **both**} **pps** *packets* **action log**	Configures the rate limits for packets that reach the supervisor module on a particular interface. If the rate of incoming or outgoing packets exceeds the configured rate limit, the device logs a system message but does not drop any packets. The range is from 1 to 100000. The default rate is 10000.
Step 3	switch(config-ppm)# **exit**	Exits the port-profile configuration mode.
Step 4	(Optional) **show system internal pktmgr interface ethernet** *slot/port*	Displays the inbound and outbound rate limit configuration for packets that reach the supervisor module on a specific interface.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the rate limits for packets that reach the supervisor module on a specific interface:

```
switch# rate-limit cpu direction both pps 1000 action log
switch# show system internal pktmgr interface ethernet 4/9
Ethernet4/9, ordinal: 44
SUP-traffic statistics: (sent/received)
Packets: 528 / 0
Bytes: 121968 / 0
Instant packet rate: 0 pps / 0 pps
Packet rate limiter (Out/In): 1000 pps / 1000 pps
Average packet rates(1min/5min/15min/EWMA):
Packet statistics:
Tx: Unicast 0, Multicast 528
Broadcast 0
Rx: Unicast 0, Multicast 0
Broadcast 0
```

Verifying 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.

Note The system displays only those ports that are allocated to the VDC that you are working in.

Use the information in the below table to verify the basic interface parameters.

Table 9: Verifying Basic Interface Parameters

Command	Purpose
show cdp	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 switchport	Displays the status of Layer 2 ports.
show interface status err-disabled	Displays information about error-disabled interfaces.
show interface status error policy [**detail**]	Displays errors about interfaces and VLANs that are inconsistent with hardware policies. The **detail** command displays the details of the interfaces that produce an error.
show vdc	Displays the status of the existing VDCs.
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.
show port-profile	Displays information about the port profiles.
show system internal pktmgr internal ethernet *slot/port*	Displays the inbound and outbound rate limit configuration for packets that reach the supervisor module on a specific interface.
show errdisable {**recovery** | **detect**}	Displays the errdisable recovery and detection runtime information.

For detailed information about the fields in the output from these commands, see the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference.

Monitoring Interface Counters

Displaying Interface Statistics

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

Note F2 Series I/O modules do not support per-VLAN statistics. Therefore, the show interface command will not display per-VLAN Rx/Tx counters or statistics for switch virtual interfaces (SVIs).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **load-interval counters** {{**1** | **2** | **3**} *seconds*}	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 network interface • 2—300 seconds • 3—not configured
Step 3	(Optional) switch(config)# **show interface** *interface*	Displays the interface status, which includes the counters.
Step 4	switch(config)# **exit**	Exits the port-profile configuration mode.
Step 5	(Optional) switch# **copy running-config startup-config**	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.

Procedure

	Command or Action	Purpose
Step 1	switch# **clear counters interface** {**all** [snmp] | **ethernet** *slot/port* [**snmp**] | **loopback** *number* | **mgmt** *number* | **port channel** *channel-number* | **tunnel** *tunnel-number* | **vlan** *vlan-number*}	Clears the interface counters.
Step 2	(Optional) switch# **show interface** *interface*	Displays the interface status.

	Command or Action	Purpose
Step 3	(Optional) switch# **show interface** [**ethernet** *slot/port* | **port-channel** *channel-number*] **counters**	Displays the interface counters.

Example

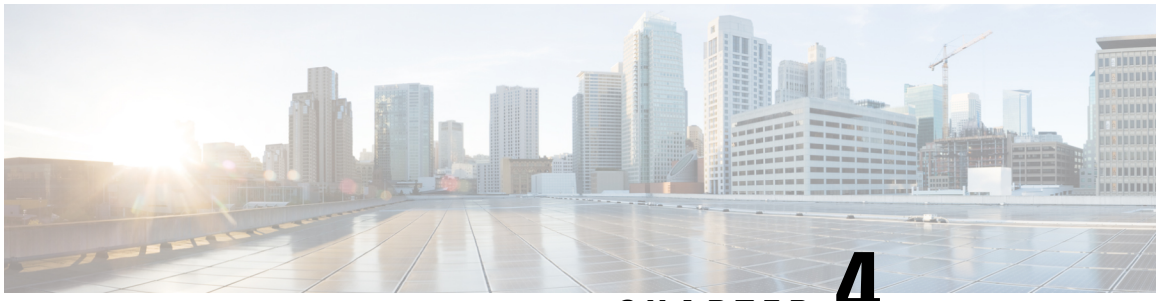
This example shows how to clear the Simple Network Management protocol (SNMP) counters on Ethernet port 5/5:

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

Related Documents

Table 10: Related Documents

Related Topic
Cisco Nexus 7000 Series NX-OS Interfaces Command Reference
Cisco Nexus 7000 Series NX-OS System Management Configuration Guide
Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide
Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 7000 Series Switches, Release 6.x
Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide
Cisco NX-OS Licensing Guide
VLANs, MAC address tables, private VLANs, and the Spanning Tree Protocol. Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide
Cisco Nexus 7000 Series NX-OS FabricPath Command Reference
Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide
Cisco Nexus 7000 Series NX-OS Release Notes



CHAPTER 4

Configuring Layer 2 Interfaces

This chapter describes how to configure Layer 2 interfaces on Cisco NX-OS devices.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Feature History for Configuring Layer 2 Interfaces

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 11: Feature History for Configuring Layer 2 Interfaces

Feature Name	Release	Feature Information
Native VLAN tagging on Trunk Ports	6.2(10)	Added support for the **switchport trunk native vlan tag** command and added the **exclude control** keywords to the **vlan dot1q tag native** command.

Feature Name	Release	Feature Information
Display policy errors on interfaces and vlans	6.2(2)	Added the **show interface status error policy** command to display errors on interfaces and VLANs that are inconsistent with hardware policies.
Clear SNMP counters from the interface	6.2(2)	Updated the **clear counters interface** command to include the **snmp** keyword that provides an option to clear SNMP values from the interface.
SVI autostate disable	6.2(2)	Added the **no autostate** command that allows an SVI to be kept up even if no interface is up in the corresponding VLAN.
Slow drain device detection and congestion avoidance	6.1(1)	Added configuration for slow drain device detection and avoiding congestion.
Default interfaces	5.2(1)	Added the **default interface** command to clear configuration of multiple interfaces.
SVI autostate exclude	5.2(1)	Added the **switchport autostate exclude** command to prevent a port's state from affecting the up or down state of the SVI.
Three configurable sampling intervals for interface statistics	4.2(1)	Added the **load-interval** command.

Information About Layer 2 Interfaces

Note From Cisco NX-OS Release 5.2, the Cisco Nexus 7000 Series devices support FabricPath Layer 2 interfaces. See the Cisco Nexus 7000 Series NX-OS FabricPath Command Reference for complete information about the FabricPath feature and interfaces.

From Cisco NX-OS Release 5.1, a Layer 2 port can function as either one of the following:

- A trunk port
- An access port
- A private VLAN port (see the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide for more information about private VLANs)
- A FabricPath port (see the Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide for information about FabricPath)

From Cisco NX-OS Release 5.2(1), a Layer 2 port can also function as a shared interface. You cannot configure an access interface as a shared interface. See the Cisco NX-OS FCoE Configuration Guide for Cisco Nexus 7000 and Cisco MDS 9000 for information about shared interfaces.

A Layer 2 port can function as either a trunk port, an access port, or a private VLAN port.

Note

- From Cisco NX-OS Release 6.1, the slow drain device detection and congestion avoidance mechanism is supported on F series I/O modules that carry the Fabric Channel over Ethernet (FCoE) traffic. See the "Configuring Slow Drain Device Detection and Congestion Avoidance" section for more information about configuring slow drain device detection and congestion avoidance on the Cisco Nexus 7000 Series platform.

  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.

- A Layer 2 port can function as either a trunk port, an access port, or a private VLAN port. See the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide for more information about private VLANs.

Access and Trunk Interfaces

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

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 ports on the device are Layer 3 ports.

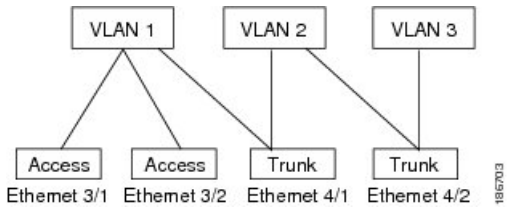
You can make all ports Layer 2 ports using the setup script or by entering the **system default switchport** command. See the Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide for information about using the setup script. To configure the port as a Layer 2 port using the CLI, use the **switchport** command.

All ports in one trunk must be in the same virtual device context (VDC). See the Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide for information about VDCs.

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

The figure below 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



See the Cisco Nexus 7000 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 7000 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.

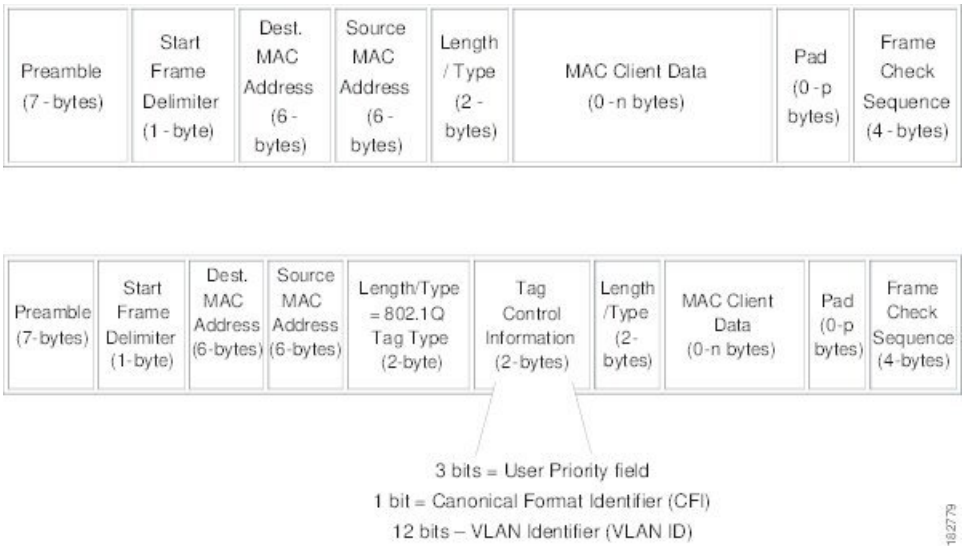
IEEE 802.1Q Encapsulation

Note For information about VLANs, see the Cisco Nexus 7000 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 (see the figure below). 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

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

See the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide for complete information on private 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.

From Cisco NX-OS Release 6.2(10), you can specify whether control and data packets are tagged or untagged using the **switchport trunk native vlan tag** command at the port level. For example, by using the **switchport trunk native vlan tag exclude control** command, you can specify that data packets are tagged and control packets are untagged.

Note When a port-level configuration is applied, the global configuration for native VLAN tagging will no longer take effect on that port. Port-level configurations take priority over global configurations.

See the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference for more information on the **switchport trunk native vlan tag** command.

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.

See the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide for more information about STP.

From Cisco Release 5.2, you can change the block of VLANs reserved for internal use. See the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide for more information about changing the reserved VLANs.

Default Interfaces

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. 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 The default interface feature is not supported for management interfaces because the device could go to an unreachable state.

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 Exclude

Typically, when a VLAN interface has multiple ports in the VLAN, the SVI goes to the down state when all the ports in the VLAN go down. You can use the SVI autostate exclude feature to exclude specific ports and port channels while defining the status of the SVI (up or down) even if it belongs to the same VLAN. For example, even if the excluded port or port channel is in the up state and other ports are in the down state in the VLAN, the SVI state is changed to down.

You can configure the SVI autostate Exclude feature on an Ethernet interface or a port channel. You can use the autostate Exclude option to enable or disable the port from bringing up or down the SVI calculation and applying it to all VLANs that are enabled on the selected port. You can also use the SVI autostate Exclude VLAN feature to exclude a VLAN from the autostate excluded interface.

Note You can use the SVI autostate exclude feature only for switched physical Ethernet ports and port channels.

SVI Autostate Disable

You can also use the SVI for inband managment of a device. 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

The software supports high availability for Layer 2 ports.

Note See the Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide for complete information about high availability features.

Virtualization Support

The device supports virtual device contexts (VDCs).

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

Note See the Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide for complete information about VDCs and assigning resources.

Prerequisites for Layer 2 Interfaces

Layer 2 interfaces have the following prerequisites:

- You are logged onto the device.
- 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.

Default Settings for Layer 2 Interfaces

Table 12: Default Access and Trunk Port Mode Parameters

Parameter	Default
Switchport mode	Access
Allowed VLANs	1 to 3967, 4048 to 4094
Access VLAN ID	VLAN1
Native VLAN ID	VLAN1
Native VLAN ID tagging	Disabled
Administrative state	Shut
SVI autostate	Enabled

Guidelines and Limitations for Layer 2 Interfaces

VLAN trunking has the following configuration guidelines and limitations:

- A port can be either a Layer 2 or a Layer 3 interface; it cannot be both simultaneously.
- QSFP-100G-DR-S and QSFP-100G-FR-S transceivers does not support breakout.
- You can view a link-up time difference of few seconds for QSFP-100G-DR-S transceiver.
- 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.
- Changing the native VLAN on an access port or trunk port will flap the interface. This behavior is expected.

Configuring Access and Trunk Interfaces

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.

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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** {{**type** *slot/port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport mode** {**access** | **trunk**}	Sets the interface as a nontrunking nontagged, single-VLAN 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	switch(config-if)# **switchport access vlan** *vlan-id*	Specifies the VLAN for which this access port will carry traffic. If you do not enter this command, the access port carries traffic on VLAN1 only; use this command to change the VLAN for which the access port carries traffic.
Step 5	switch(config-if)# **exit**	Exits the interface mode.
Step 6	switch(config)# **exit**	Exits global configuration mode.
Step 7	(Optional) switch# **show interface**	Displays the interface status and information.
Step 8	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 9	(Optional) switch# **no shutdown**	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 10	(Optional) switch(config)# **copy running-config startup-config**	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
```

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 the Cisco Nexus 7000 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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport host**	Sets the interface to 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	switch(config-if)# **exit**	Exits the interface mode.
Step 5	switch(config)# **exit**	Exits the configuration mode.

	Command or Action	Purpose
Step 6	(Optional) switch# **show interface**	Displays the interface status and information.
Step 7	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 8	(Optional) switch# **no shutdown**	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	(Optional) switch(config)# **copy running-config startup-config**	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
```

Configuring a Trunk Port

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.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **interface** {{**type** *slot/port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport mode** {**access** | **trunk**}	Sets the interface as a 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	switch(config-if)# **exit**	Exits the interface mode.
Step 5	switch(config)# **exit**	Exits the configuration mode.
Step 6	(Optional) switch# **show interface**	Displays the interface status and information.
Step 7	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 8	(Optional) switch# **no shutdown**	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	(Optional) switch(config)# **copy running-config startup-config**	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
```

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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** {{**type** *slot/port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport trunk native vlan** *vlan-id*	Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 4094, except those VLANs reserved for internal use. The default value is VLAN1.
Step 4	switch(config-if)# **exit**	Exits the interface mode.
Step 5	switch(config)# **exit**	Exits global configuration mode.
Step 6	switch# **show vlan**	Displays the status and information of VLANs.
Step 7	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 8	(Optional) switch# **no shutdown**	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	(Optional) switch(config)# **copy running-config startup-config**	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
```

Configuring the Allowed VLANs for Trunking Ports

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

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.

From Cisco Release 5.2, you can change the block of VLANs reserved for internal use. See the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide for more information about changing the reserved VLANs.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** {{**type** *slot/port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport trunk allowed vlan** {*vlan-list* | **add** *vlan-list* | **all** | **except** *vlan-list* | **none** | **remove** *vlan-list*}	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. From Cisco Release 5.2(1), the default reserved VLANs are 3968 to 4094, and you can change the block of reserved VLANs. See the Cisco Nexus 7000 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	switch(config-if)# **exit**	Exits the interface mode.
Step 5	switch(config)# **exit**	Exits the configuration mode.
Step 6	switch# **show vlan**	Displays the status and information of VLANs.

	Command or Action	Purpose
Step 7	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 8	(Optional) switch# **no shutdown**	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	(Optional) switch(config)# **copy running-config startup-config**	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
```

Configuring a Default Interface

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **default interface** *int-if* [**checkpoint** *name*]	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	switch(config)# **exit**	Exits global configuration mode.
Step 4	switch# **show interface**	Displays the interface status and information.
Step 5	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies.

	Command or Action	Purpose
		Use the **detail** command to display the details of the interfaces that produce an error.
Step 6	(Optional) switch# **no shutdown**	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
```

Configuring SVI Autostate Exclude

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** {{**type** *slot/port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport**	Sets the interface as a Layer 2 interface.
Step 4	switch(config-if)# [**no**] **switchport autostate exclude**	Excludes this port from the VLAN interface link-up calculation when there are multiple ports in the VLAN. To revert to the default settings, use the **no** form of this command.
Step 5	switch(config-if)# [**no**] **switchport autostate exclude vlan** *vlan id*	Excludes a vlan or a set of vlans from the autostate-excluded interface. This will help to minimize any disruption to the system. To revert to the default settings, use the **no** form of this command.
Step 6	switch(config-if)# **exit**	Exits the interface mode.
Step 7	switch(config)# **exit**	Exits global configuration mode.

	Command or Action	Purpose
Step 8	(Optional) switch# **show running-config interface** {{**type** *slot/port*} | {**port-channel** *number*}	Displays configuration information about the specified interface.
Step 9	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 10	(Optional) switch# **no shutdown**	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 11	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to exclude a port from the VLAN interface link-up calculation on the Cisco NX-OS device:

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

This example shows how to exclude a VLAN from the auto-excluded interface:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport
switch(config-if)# switchport autostate exclude
switch(config-if)# switchport autostate exclude vlan 10
```

Configuring SVI Autostate Disable for the System

You can configure the SVI autostate disable feature to keep an SVI up even if no interface is up in the corresponding VLAN. Use this procedure to configure this feature for the entire system.

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **system default interface-vlan no autostate**	Disables the default autostate behavior for the device.
Step 3	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 4	(Optional) switch# **no shutdown**	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 5	(Optional) switch# **show running-config** [**all**]	Displays the running configuration. 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)# system default interface-vlan no 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.

Before you begin

Before you configure this feature at SVI-level, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **feature interface-vlan**	Enables VLAN interface mode.
Step 3	switch(config)# **interface vlan** *vlan-id*	Creates a VLAN interface and enters interface configuration mode. The range is from 1 and 4094.
Step 4	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	switch(config-if)# **exit**	Exits interface configuration mode.
Step 6	(Optional) switch(config)#**show running config-interface vlan** *vlan-id*	Displays the running configuration for the specified VLAN interface.
Step 7	(Optional) switch(config)# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 8	(Optional) switch(config)# **no shutdown**	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	(Optional) switch(config)# **show startup-config interface vlan** *vlan id*	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.

Note If you enable 802.1Q tagging on the device, you need to enable **vlan dotq tag native exclude control** globally or enable **switchport trunk native vlan tag exclude control** at interface level. This will ensure the port-channel with LACP to work correctly.

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command. You can repeat VLAN names and IDs in different VDCs, so you must confirm that you are working in the correct VDC.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	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. The default is disabled.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show vlan**	Displays the status and information for VLANs.
Step 5	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.

	Command or Action	Purpose
Step 6	(Optional) switch# **no shutdown**	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	(Optional) switch# **copy running-config startup-config**	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.

See the Cisco NX-OS FCoE Configuration Guide for Cisco Nexus 7000 and Cisco MDS 9000 for information on setting the system default port mode to Fibre Channel in storage VDCs.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **system default switchport** [**shutdown**]	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**.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show interface brief**	Displays the status and information for interfaces.

	Command or Action	Purpose
Step 5	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 6	(Optional) switch# **no shutdown**	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	(Optional) switch(config)# **copy running-config startup-config**	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
```

Configuration Examples for Access Ports 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)#
```

Configuring Slow Drain Device Detection and Congestion Avoidance

The data traffic between the end devices in Fibre Channel over Ethernet (FCoE) uses link level and per-hop based flow control. When the slow devices are attached to the fabric, the end devices do not accept the frames at a configured rate. The presence of the slow devices leads to traffic congestion on the links. The traffic congestion affects the unrelated flows in the fabric that use the same inter-switch links (ISLs) for its traffic, even though the destination devices do not experience the slow drain.

From Cisco NX-OS Release 6.1, slow drain device detection and congestion avoidance is supported on the F-series I/O modules that carry the FCoE traffic. The enhancements are mainly on the edge ports that are connected to the slow drain devices to minimize the congestion condition in the edge ports.

Once the slow drain devices are detected on the network, you can configure a smaller frame timeout value for the edge ports and force a timeout drop for all the packets that are using the configured thresholds. The smaller frame timeout value helps to alleviate the slow drain condition that affects the fabric by dropping the packets on the edge ports sooner than the time they actually get timed out. The default timeout value is 500 milliseconds. This function empties the buffer space in ISL, which can be used by other unrelated flows that do not experience the slow drain condition.

If you try to override the Embedded Event Manager (EEM) system policy __ori_mac_edge_pause for the F1 I/O module and __clm_sw_edge_port_pause for the F2 /I/O module, the default-action, default syslog, will also appear. We recommend that you specify the action err-disable to isolate the faulty port where this condition occurs.

This example shows how to override the EEM system policy for an F1 I/O module:

```
event manager applet my_eem_policy override __ori_mac_edge_pause
description "my_f1_Pause_eem_policy"
event policy-default count 1 time 2
action 1.0 cli switchto vdc storage
action 2.0 cli eth-port-manager internal-errdisable $interface $cause $SYSERR
```

Configuring a Congestion Frame Timeout Value

When an FCoE frame takes longer than the congestion-drop timeout period to be transmitted by the egress port, the frame is dropped. This dropping of the frames is useful in controlling the effect of slow egress ports that are paused almost continuously (long enough to cause congestion), but not long enough to trigger the pause timeout drop. Frames dropped due to the congestion drop threshold are counted as egress discards against the egress port. Egress discards release buffers in the upstream ingress ports of the switch, allowing the unrelated flows to move continuously through them.

The default congestion frame timeout value is 500 milliseconds. We recommend that you retain the default configuration for the ISLs and configure a value that does not exceed the default value for the edge ports. If the frame is in the switch for a longer time than the configured congestion frame timeout, it gets dropped, which empties the buffer space in the ISL and alleviates the congestion.

To configure the congestion drop timeout value for FCoE, perform the following steps:

Procedure

Step 1 Enter configuration mode:

switch# **configure terminal**

Step 2 Depending on the Cisco Nexus 7000 NX-OS release version you are using, use one of the following commands to configure the system-wide FCoE congestion drop timeout, in milliseconds, for either core or edge ports

- Cisco Nexus 7000 NX-OS Release 8.1(1) and earlier releases:

  switch(config)# **system default interface congestion timeout** *milliseconds* **mode {core | edge}**

  Configures a new congestion frame timeout value in milliseconds and the port mode for the device. The FCoE congestion drop timeout range is from 100 to 1000 ms. To prevent premature packet drops, the minimum value recommended for FCoE congestion drop timeout is 200 milliseconds.

- Cisco Nexus 7000 NX-OS Release 8.2(1) and later releases:

  switch(config)# **system timeout fcoe congestion-drop** {*milliseconds* | **default**} **mode {core | edge}**

  Configures a new congestion frame timeout value in milliseconds and the port mode for the device. The FCoE congestion drop timeout range is from 200 to 500 ms.

(Optional) Depending on the Cisco Nexus 7000 NX-OS release version you are using, use one of the following commands to revert to the default FCoE congestion drop timeout value of 500 milliseconds:

- Cisco Nexus 7000 NX-OS Release 8.1(1) and earlier releases:

  switch(config)# **no system default interface congestion timeout** *milliseconds* **mode {core | edge}**

- Cisco Nexus 7000 NX-OS Release 8.2(1) and later releases:

  switch(config)# **no system timeout fcoe congestion-drop** {*milliseconds* | **default**} **mode {core | edge}**

Step 3 (Optional) switch# **show logging onboard flow-control request-timeout**

Displays the request timeout for a source-destination pair per module with the timestamp information.

Example

Note
- The congestion frame timeout configuration is local to a vdc and will be effective only on the ports (edge/core) owned by the vdc.
- Use the default configuration for the core ports and configure a congestion frame timeout value for the fabric edge ports that does not exceed 500 milliseconds. The recommended range for the congestion frame timeout value is from 100 to 200 milliseconds.

The following example shows how to display the request timeout for a source-destination pair per module with the timestamp information for the supervisor CLI:

```
SUP CLI:
switch# show logging onboard flow-control request-timeout
----------------------------
Module: 2
----------------------------
-------------------------------------------------------------------------------
| Dest |     Source |   Events   |   Timestamp    |  Timestamp      |
| Intf |       Intf |    Count   |     Earliest   |   Latest        |
```

```
--------------------------------------------------------------------------------
|fc4/3 |eth2/1,eth2/2 | 1736|11/14/2002-00:40:07|11/14/2002-00:57:22|
--------------------------------------------------------------------------------
|fc4/3 |eth2/1,eth2/2 | 3477|11/13/2002-23:23:27|11/14/2002-00:00:48|
--------------------------------------------------------------------------------
|fc4/3 |eth2/1,eth2/2 | 4298|11/13/2002-22:31:40|11/13/2002-23:18:00|
--------------------------------------------------------------------------------
|fc4/3 |eth2/1,eth2/2 | 9690|11/13/2002-04:54:50|11/13/2002-07:31:58|
```

The following example shows how to display the request timeout for a source-destination pair per module with the time-stamp information for the module CLI:

```
Module CLI:
module--x# show logging onboard flow-control request-timeout
--------------------------------------------------------------------------------
| Dest |     Source |   Events   |   Timestamp    |  Timestamp      |
| Intf |       Intf |    Count   |     Earliest   |   Latest        |
--------------------------------------------------------------------------------
|fc4/3 |eth2/1,eth2/2 | 1736|11/14/2002-00:40:07|11/14/2002-00:57:22|
--------------------------------------------------------------------------------
|fc4/3 |eth2/1,eth2/2 | 3477|11/13/2002-23:23:27|11/14/2002-00:00:48|
--------------------------------------------------------------------------------
|fc4/3 |eth2/1,eth2/2 | 4298|11/13/2002-22:31:40|11/13/2002-23:18:00|
--------------------------------------------------------------------------------
|fc4/3 |eth2/1,eth2/2 | 9690|11/13/2002-04:54:50|11/13/2002-07:31:58|
```

Note The following example outputs are applicable for Cisco Nexus 7000 NX-OS 8.2(1) release and later:

The following example shows how to configure congestion-drop timeout to the default value of 500 milliseconds for a core device:

```
switch# configure terminal
switch(config)# system timeout fcoe congestion-drop default mode core
```

The following example shows how to configure congestion-drop timeout to the default value of 500 milliseconds for a edge device:

```
switch# configure terminal
switch(config)# system timeout fcoe congestion-drop default mode edge
```

Configuring a smaller timeout on the edge ports, for example 200 milliseconds, helps to reduce the congestion on the edge ports. When congestion is observed, the packets on these ports time out sooner.

Note

- The congestion frame timeout configuration is local to a Virtual Device Context (VDC) and will be effective only on the ports (edge/core) owned by the VDC.
- Use the default configuration for the core ports and configure a congestion-frame timeout value for the fabric-edge ports that does not exceed 500 milliseconds. The recommended range for the congestion-frame timeout value is from 200 to 500 milliseconds.

Configuring a Pause Frame Timeout Value

From Cisco NX-OS 6.1 release, you can enable or disable a pause frame timeout value on a port. The system periodically checks the ports for a pause condition and enables a pause frame timeout on a port if it is in a continuous pause condition for a configured period of time. This situation results in all frames that come to that port getting dropped in the egress. This function empties the buffer space in the ISL link and helps to reduce the fabric slowdown and the congestion on the other unrelated flows using the same link.

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

The pause frame timeout is enabled by default and the value is set to 500 milliseconds. We recommend that you retain the default configuration for the ISLs and configure a value that does not exceed the default value for the edge ports.

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

Procedure

	Command or Action	Purpose
Step 1	Enter configuration mode:	switch# **configure terminal**
Step 2	Depending on the Cisco Nexus 7000 NX-OS release version you are using, use one of the following commands to configure the system-wide FCoE pause drop timeout value, in milliseconds, for either edge or core ports:	• Cisco Nexus 7000 NX-OS Release 8.1(1) and earlier releases: switch(config)# **system default interface pause timeout** *milliseconds* **mode {core | edge}** • (Optional) Depending on the Cisco Nexus 7000 NX-OS release version you are using, use one of the following commands to enable the FCoE pause drop timeout to the default value of 500 milliseconds for edge or core ports: • Cisco Nexus 7000 NX-OS Release 8.1(1) and earlier releases: switch(config)# **system default interface pause mode {core | edge}** • (Optional) Depending on the Cisco Nexus 7000 NX-OS release version you are using, use one of the following commands to disable the FCoE pause drop timeout for edge or core ports: • Cisco Nexus 7000 NX-OS Release 8.1(1) and earlier releases:

	Command or Action	Purpose
		switch(config)# **no system default interface pause mode {core | edge}**
Step 3	(Optional) switch# **show logging onboard flow-control pause-event** [**module** *x*]	Displays the total number of the pause events per module per interface.
Step 4	(Optional) switch# **show logging onboard flow-control pause-count** [**module** *x*] [*last mm minutes*] [*last hh hours*] [*last dd days*]	Displays the pause counters per module per interface with the time-stamp information.
Step 5	(Optional) switch# **show logging onboard flow-control timeout-drops** [**module** *x*] [*last mm minutes*] [*last hh hours*] [*last dd days*]	Displays the timeout drops per module per interface with the time-stamp information.

Example

The example shows how to display the total number of the pause events per module per interface for the supervisor CLI:

```
SUP CLI:
switch# show logging onboard flow-control pause-event module 2
----------------------------
Module: 2
----------------------------
--------------------------------------------------------------------------------
STATISTICS INFORMATION FOR DEVICE ID 137 DEVICE Orion MAC Driver
--------------------------------------------------------------------------------
                          |           | Time Stamp      |In|Port
Error Stat Counter Name   | Count     |MM/DD/YY HH:MM:SS|st|Range
                          |           |                 |Id|
--------------------------------------------------------------------------------
SW PL0 pause event VL3    |0x4e45b    |06/18/03 05:27:50 |00|1
SW PL0 pause event VL3    |0x4e1a0    |06/18/03 05:25:50 |00|1
SW PL0 pause event VL3    |0x4dee5    |06/18/03 05:23:50 |00|1
SW PL0 pause event VL3    |0x4dc2a    |06/18/03 05:21:50 |00|1
```

The example shows how to display the total number of the pause events per module per interface for the module CLI:

```
Module CLI:
module-2# show logging onboard flow-control pause-event
--------------------------------------------------------------------------------
STATISTICS INFORMATION FOR DEVICE ID 137 DEVICE Orion MAC Driver
--------------------------------------------------------------------------------
                          |           | Time Stamp      |In|Port
Error Stat Counter Name   | Count     |MM/DD/YY HH:MM:SS|st|Range
                          |           |                 |Id|
--------------------------------------------------------------------------------
SW PL0 pause event VL3    |0x4e45b    |06/18/03 05:27:50|00|1
SW PL0 pause event VL3    |0x4e1a0    |06/18/03 05:25:50|00|1
SW PL0 pause event VL3    |0x4dee5    |06/18/03 05:23:50|00|1
SW PL0 pause event VL3    |0x4dc2a    |06/18/03 05:21:50|00|1
SW PL0 pause event VL3    |0x4d96f    |06/18/03 05:19:50|00|1
```

The following example shows how to display the pause counters per module per interface with time-stamp information for the supervisor CLI:

```
SUP CLI:
switch# show logging onboard flow-control pause-count
--------------------------------------------------------------------------------
STATISTICS INFORMATION FOR DEVICE ID 137 DEVICE Orion MAC Driver
--------------------------------------------------------------------------------
                                  |            | Time Stamp      |In|Port
Error Stat Counter Name           | Count      |MM/DD/YY HH:MM:SS|st|Range
                                  |            |                 |Id|
--------------------------------------------------------------------------------
GD Received pause transitions of XO|0x984       |06/17/03 14:23:59|00|1
FF-XON UP3                        |            |                 |  |
GD Received pause transitions of XO|0x41f       |06/17/03 14:21:59|00|1
FF-XON UP3                        |            |                 |  |
```

The example shows how to display the pause counters per module per interface with time-stamp information for the module CLI:

```
Module CLI:
module-2# show logging onboard flow-control pause-count
--------------------------------------------------------------------------------
STATISTICS INFORMATION FOR DEVICE ID 137 DEVICE Orion MAC Driver
--------------------------------------------------------------------------------
                                  |            | Time Stamp      |In|Port
Error Stat Counter Name           | Count      |MM/DD/YY HH:MM:SS|st|Range
                                  |            |                 |Id|
--------------------------------------------------------------------------------
GD Received pause transitions of XO|0x984 |06/17/03 14:23:59|00|1
FF-XON UP3                        |      |                 |  |
GD Received pause transitions of XO|0x41f |06/17/03 14:21:59|00|1
FF-XON UP3                        |      |                 |  |
```

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

```
SUP CLI:
switch# show logging onboard flow-control timeout-drops
switch# show logging onboard flow-control timeout-drops
----------------------------
Module: 2
----------------------------
--------------------------------------------------------------------------------
STATISTICS INFORMATION FOR DEVICE ID 137 DEVICE Orion MAC Driver
--------------------------------------------------------------------------------
                                  |            | Time Stamp      |In|Port
Error Stat Counter Name           | Count      |MM/DD/YY HH:MM:SS|st|Range
                                  |            |                 |Id|
--------------------------------------------------------------------------------
ORI_EB_CNT_P0_SF_TIMESTAMP_DROP   |0x100e      |11/14/02 00:45:43|00|1
ORI_EB_CNT_P0_SF_TIMESTAMP_DROP   |0xfd2       |11/14/02 00:43:42|00|1
Module CLI:
```

The following example shows how to display the timeout drops per module per interface with time-stamp information for the module CLI:

```
module-2# show logging onboard flow-control timeout-drops
--------------------------------------------------------------------------------
STATISTICS INFORMATION FOR DEVICE ID 137 DEVICE Orion MAC Driver
--------------------------------------------------------------------------------
```

```
                                    |            | Time Stamp      |In|Port
Error Stat Counter Name             | Count      |MM/DD/YY HH:MM:SS|st|Range
                                    |            |                 |Id|
-------------------------------------------------------------------------------
ORI_EB_CNT_P0_SF_TIMESTAMP_DROP     |0x100e      |11/14/02 00:45:43|00|1
ORI_EB_CNT_P0_SF_TIMESTAMP_DROP     |0xfd2       |11/14/02 00:43:42|00|1
```

Note The following examples are applicable for Cisco Nexus 7000 NX-OS 8.2(1) release and later:

The following example shows how to configure pause-drop timeout to the default value of 500 milliseconds for a core device:

```
switch# configure terminal
switch(config)# system timeout fcoe pause-drop default mode core
```

The following example shows how to configure pause-drop timeout to the default value of 500 milliseconds for a edge device:

```
switch# configure terminal
switch(config)# system timeout fcoe pause-drop default mode edge
```

Use the [**no**] **system timeout fcoe pause-drop** {*milliseconds* |**default** }[**mode**] **edge** command to disable the pause frame timeout value on the edge ports.

Verifying the Interface Configuration

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

Table 13: Verifying the Interface Configuration

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.
show interface capabilities	Displays information about the capabilities of the interfaces.
show interface status error policy [**detail**]	Displays errors about interfaces and VLANs that are inconsistent with hardware policies. The **detail** command displays the details of the interfaces that produce an error.

Command	Purpose
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.

For detailed information about these commands, see the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Command Reference.

Monitoring Layer 2 Interfaces

Use the following commands to display Layer 2 interfaces:

Table 14: Monitoring Layer Interfaces

Command	Purpose
clear counters interface [*interface*]	Clears the counters.
load-interval {**interval** *seconds* {**1** | **2** | **3**}}	From Cisco NX-OS Release 4.2(1) for the Cisco Nexus 7000 Series devices, 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 as well as output packets and bytes.
show interface counters errors [**module** *module*]	Displays information on the number of error packets.

See the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference for information on these commands.

Related Documents

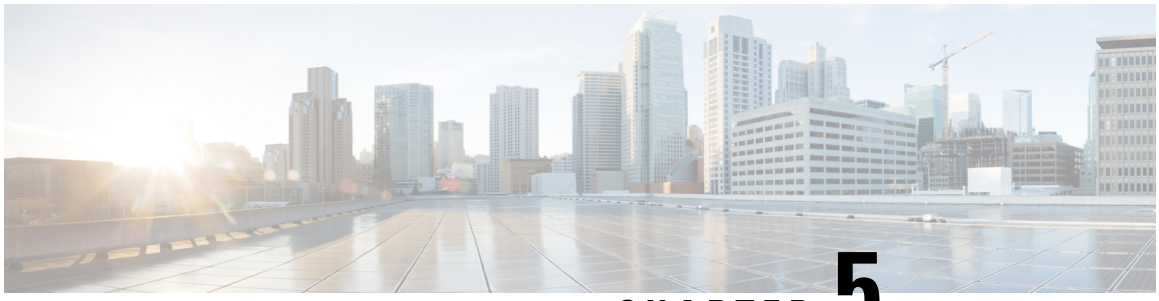
Table 15: Related Documents

Related Topic
Cisco Nexus 7000 Series NX-OS Interfaces Command Reference

Related Topic
Cisco Nexus 7000 Series NX-OS System Management Configuration Guide
Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide
Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 7000 Series Switches, Release 6.x
Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide
Cisco NX-OS Licensing Guide
VLANs, MAC address tables, private VLANs, and the Spanning Tree Protocol. Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide
Cisco Nexus 7000 Series NX-OS FabricPath Command Reference
Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide
Cisco Nexus 7000 Series NX-OS Release Notes

MIBs

MIBs	MIBs Link
• BRIDGE-MIB • IF-MIB • CISCO-IF-EXTENSION-MIB • ETHERLIKE-MIB	To locate and download MIBs, go to: https://cfnng.cisco.com/mibs



CHAPTER 5

Configuring Layer 3 Interfaces

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Feature History for Layer 3 Interfaces

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 16: Feature History for Layer 3 Interfaces

Feature Name	Release	Feature Information
Display errors during policy programming.	6.2(2)	Added the show interface status error policy command which displays the interfaces and VLANs that produce an error during policy programming.

Feature Name	Release	Feature Information
Clear SNMP counters from the interface	6.2(2)	Updated the **clear counters interface** command to include a keyword **snmp** that provides an option to clear SNMP values from the interface.
FEX	6.2(2)	Cisco Fabric Extenders support Layer 3 protocol adjacencies on host interfaces (HIFs) and DSCP to queue mapping. **Note** Before Cisco NX-OS Release 6.2(2), you can configure a Fabric Extender (FEX) port as a Layer 3 interface for host connectivity, but not for routing.
Enhanced show output for sub-interfaces	6.1(1)	Updated the **show interface eth** command output.
Three configurable sampling intervals for interface statistics	4.2(1)	Added the load-interval command.
Layer 3 interfaces	4.0(1)	This feature was introduced.

Information 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.

You cannot configure a shared interface as a Layer 3 interface. See the Cisco NX-OS FCoE Configuration Guide for Cisco Nexus 7000 and Cisco MDS 9000 for information about shared interfaces.

For Cisco NX-OS Release 6.2(2) and later releases, the Cisco Fabric Extenders support Layer 3 protocol adjacencies on host interfaces (HIFs) and DSCP to queue mapping. Before Cisco NX-OS Release 6.2(2), you can configure a Fabric Extender (FEX) port as a Layer 3 interface for host connectivity, but not for routing. See the Configuring the Cisco Nexus 2000 Series Fabric Extender for more information about fabric extenders. See the Cisco Nexus 2000 Series Fabric Extender Software Configuration Guide for Cisco Nexus 7000 Series Switches, Release 7.x for more information about fabric extenders.

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 or through the **system default switchport** command.

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

From Cisco Release 4.2(1), you can assign a static MAC address to a Layer 3 interface. By default, the MAC address for the Layer 3 interfaces is the MAC address of the virtual device context (VDC) it is assigned to. For information on configuring MAC addresses, see the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide.

Routed interfaces and subinterfaces 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 or a port channel.

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.

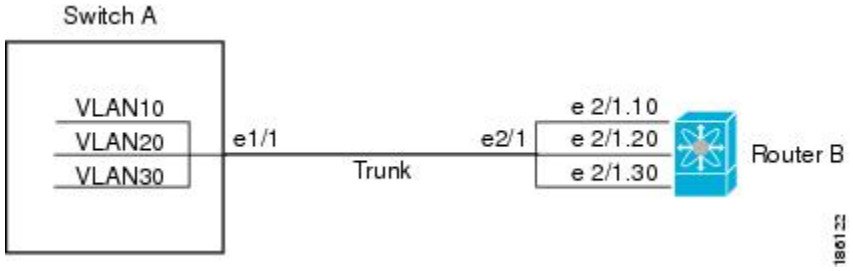
You create a subinterface with a name that consists of the parent interface name (for example, Ethernet 2/1) followed by a period and then by a number that is unique for that subinterface. For example, you could create a subinterface for Ethernet interface 2/1 named Ethernet 2/1.1 where .1 indicates 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 figure below 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 7000 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. Beginning in Cisco NX-OS Release 4.2, the system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. See the Cisco Nexus 7000 Series NX-OS System Management Configuration Guide for information on rollbacks and checkpoints.

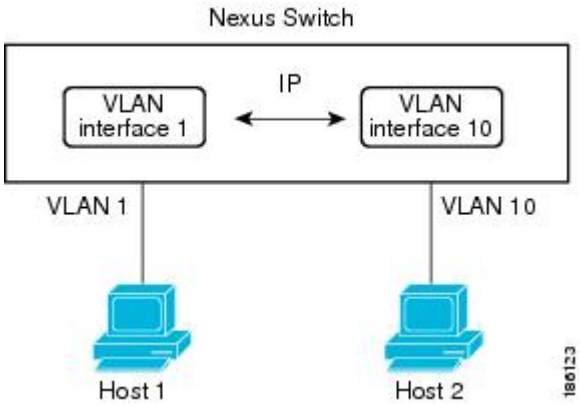
You must configure the VLAN network interface in the same VDC as the VLAN.

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 7000 Series NX-OS Unicast Routing Configuration Guide.

The figure below 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



Note You can configure VLAN interface for an inband management in the Cisco Nexus 7000 Series devices with the F1 Series modules in the chassis.

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 per VDC, 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.

Tunnel Interfaces

Cisco NX-OS supports tunnel interfaces as 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 routers.

High Availability for Layer 3 Interfaces

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 7000 Series NX-OS High Availability and Redundancy Guide for complete information on high availability.

Virtualization Support for Layer 3 Interfaces

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 unless you specifically configure another VDC and VRF. A Layer 3 logical interface (VLAN interface, loopback) configured in one VDC is isolated from a Layer 3 logical interface with the same number configured in another VDC. For example, loopback 0 in VDC 1 is independent of loopback 0 in VDC 2.

You can configure up to 1024 loopback interfaces per VDC.

You can associate the interface with a VRF. For VLAN interfaces, you must configure the VLAN interface in the same VDC as the VLAN.

See the Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide for information about VDCs and see the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for information about configuring an interface in a 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 have entered the desired VDC (see the Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, if you are configuring VDCs.

- You are familiar with IP addressing and basic configuration. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, for more information on IP addressing.

Guidelines and Limitations for Layer 3 Interfaces

Layer 3 interfaces have the following configuration guidelines and limitations:

- QSFP-100G-DR-S and QSFP-100G-FR-S transceivers does not support breakout.
- You can view a link-up time difference of few seconds for QSFP-100G-DR-S transceiver.
- 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.
- 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.
- Configuring a subinterface on a physical interface that is configured to be a member of a port-channel is not supported. One must configure the subinterface under the port-channel interface itself.
- The Cisco Nexus 2000 Fabric Extender cannot participate in a routing protocol adjacency with a device attached to its port. Only a static direct route is supported. This restriction applies to both of the supported connectivity cases:
  - SVI with Fabric Extender single port or portchannel in Layer 2 mode.
  - Fabric Extender port or portchannel in Layer 3 mode.
- Layer 3 router interfaces and subinterfaces cannot be configured on an F1 I/O module.
- When using an L3 interface on F-series modules (F2/F2e/F3) in Cisco Nexus 7000 series it is mandatory to configure QoS mapping on DSCP instead of CoS.

  Do not configure the QoS mapping on Cos because when the matching happens on CoS the L3 control traffic is placed into the default class and could be dropped due to normal congestion.

  The QoS mapping is configured in the Admin VDC using **hardware qos dscp-to-queue ingress module-type all** command or **hardware qos dscp-to-queue ingress module-type f-series** command.
- F2-series I/O modules do not support per-VLAN statistics. Therefore, the show interface command does not display per-VLAN Rx/Tx counters or statistics for switch virtual interfaces (SVIs).

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 for Layer 3 Interfaces

Table 17: Default Layer 3 Interface Parameters

Parameter	Default
Administrative state	Shut

Configuring Layer 3 Interfaces

Configuring a Routed Interface

You can configure any Ethernet port as a routed interface.

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **no switchport**	Configures the interface as a Layer 3 interface and deletes any configuration specific to Layer 2 on this interface.
Step 4	switch(config-if)# {**ip** | **ipv6**} **address** *ip-address/length*	Configures an IP address for this interface. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for more information about IP and IPv6 addresses.
Step 5	(Optional) switch(config-if)# **show interfaces**	Displays the Layer 3 interface statistics.
Step 6	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 7	(Optional) switch# **no shutdown**	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

	Command or Action	Purpose
		up. If policies do not correspond, the errors are placed in an error-disabled policy state.
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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

Command	Purpose
medium {**broadcast** | **p2p**}	Configures the interface medium as either point to point or broadcast.

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 confi**g command.

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

Command	Purpose
switchport	Configures the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.

Example

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

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

Before you begin

- Configure the parent interface as a routed interface.
- See the "Configuring a Routed Interface" section.
- Create the port-channel interface if you want to create a subinterface on that port channel.

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# {**ip** | **ipv6**} **address** *ip-address/length*	Configures an IP address for this interface. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for more information about IP and IPv6 addresses.
Step 4	switch(config-if)# **encapsulation dot1q** *vlan-id*	Configures IEEE 802.1Q VLAN encapsulation on the subinterface. The range is from 2 to 4093.
Step 5	(Optional) switch(config-if)# **show interfaces**	Displays the Layer 3 interface statistics.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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
```

This example shows the output of the **show interface eth** command that is enhanced for the subinterfaces from Cisco NX-OS Release 6.1:

```
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
Encapsulation 802.1Q Virtual LAN, Vlan ID 11, medium is broadcast
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 the Bandwidth on an Interface

You can configure the bandwidth for a routed interface, port channel, or subinterface. Higher layer protocols use a bandwidth parameter to calculate path costs. You can configure the bandwidth on a subinterface with one of the following methods:

- Explicit—Sets the bandwidth value for the subinterface directly.
- Inherit—Sets the bandwidth that all subinterfaces inherit from the parent interface as either a specific value or as the bandwidth of the parent interface.

If you do not set the subinterface bandwidth or configure it to inherit the bandwidth from the parent interface, Cisco NX-OS determines the subinterface bandwidth as follows:

- If the parent interface is up, the bandwidth of the subinterface is the same as the operational speed of the parent interface. For ports, the subinterface bandwidth is the configured or negotiated link speed. For port channels, the subinterface bandwidth is the aggregate of the link speeds of individual members of the port channel.
- If the parent interface is down, the bandwidth of the subinterface depends on the type of parent interface:
  - Port-channel subinterfaces have 100-Mb/s bandwidth for subinterfaces.
  - 1-Gb/s Ethernet ports have 1-Gb/s bandwidth for subinterfaces.
  - 10-Gb/s Ethernet ports have 10-Gb/s bandwidth for subinterfaces.

To configure the bandwidth of an interface, use the following command in interface mode:

Command	Purpose
bandwidth	Configures the bandwidth parameter for a routed interface, port channel, or subinterface.

To configure subinterfaces to inherit the bandwidth from the parent interface, use the following command in interface mode:

Command	Purpose
bandwidth inherit [*value*]	Configures all subinterfaces of this interface to inherit the bandwidth value configured. If you do not configure the value, the subinterfaces inherit the bandwidth of the parent interface. The range is from 1 to 10000000, in kilobytes.

Configuring a VLAN interface

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

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)#**feature interface-vlan**	Enables VLAN interface mode.
Step 3	switch(config)# **interface vlan** *number*	Creates a VLAN interface. The number range is from 1 to 4094.
Step 4	switch(config-if)# {**ip** | **ipv6**} **address** *ip-address/length*	Configures an IP address for this interface. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for more information about IP and IPv6 addresses.
Step 5	(Optional) switch(config-if)# **show interface vlan** *number*	Displays the Layer 3 interface statistics.
Step 6	(Optional) switch(config-if)# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 7	(Optional) switch(config-if)# **no shutdown**	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	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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 Inband Management in the Nexus Chassis

You can create a VLAN interface for inband management in the Cisco Nexus 7000 Series devices when there are only F1 Series modules in the chassis.

Note We recommend that you use a dedicated VLAN for inband management on the F1 Series modules. Do not run data traffic on the VLAN that you are using for inband management.

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)#**feature interface-vlan**	Enables VLAN interface mode.
Step 3	switch(config)# **interface vlan** *number*	Creates a VLAN interface. The number range is from 1 to 4094.
Step 4	switch(config-if)# **no shutdown**	Brings an interface administratively up (enable/disable an interface).
Step 5	switch(config-if)# **management**	Allows in-band management access to a VLAN interface IP address.
Step 6	switch(config-if)# **ip address** *ip-address/length*	Configures an IP address for this interface. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for more information about IP addresses.
Step 7	(Optional) switch(config-if)# **show interface vlan** *number*	Displays the Layer 3 interface statistics.
Step 8	(Optional) switch(config-if)# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce errors during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 9	(Optional) switch(config-if)# **no shutdown**	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 10	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create an inband management in the Cisco Nexus 7000 chassis:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan 5
switch(config-if)# no shutdown
switch(config-if)# management
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.
- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface loopback** *instance*	Creates a loopback interface. The range is from 0 to 1023.
Step 3	switch(config-if)# {**ip** | **ipv6**} **address** *ip-address/length*	Configures an IP address for this interface. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for more information about IP and IPv6 addresses.
Step 4	switch(config-if)# **show interfaces loopback** *instance*	Displays the loopback interface statistics.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create a loopback interface:

```
switch# configure terminal
switch(config)# interface loopback 0
```

```
switch(config-if)# ip address 192.0.2.100/8
switch(config-if)# copy running-config startup-config
```

Assigning an Interface to a VRF

You can add a Layer 3 interface to a VRF.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Assign the IP address for a tunnel interface after you have configured the interface for a VRF.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *interface-type number*	Enters interface configuration mode.
Step 3	switch(config-if)# **vrf member** *vrf-name*	Adds this interface to a VRF.
Step 4	switch(config-if)# **ip address** *ip-address/length*	Configures an IP address for this interface. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for more information about IP addresses.
Step 5	(Optional) switch(config-if)# **show vrf** [*vrf-name*] **interface** *interface-type number*	Displays VRF information.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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
```

Verifying the Layer 3 Interfaces Configuration

Table 18: Verifying the Layer 3 Interfaces Configuration

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).
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* **private-vlan mapping**	Displays the VLAN interface private VLAN information.
show interface vlan *number* **status**	Displays the VLAN interface administrative status and protocol status.

Command	Purpose
show interface status error policy [**detail**]	Displays errors on interfaces and VLANs that are inconsistent with hardware policies. The **detail** command displays the details of the interfaces and VLANs that receive an error.

Monitoring Layer 3 Interfaces

Use the following commands to display Layer 2 interfaces:

Table 19: Monitoring Layer 3 Interfaces

Command	Purpose
load- interval {**interval** *seconds* {**1** | **2** | **3**}}	From Cisco NX-OS Release 4.2(1) for the Cisco Nexus 7000 Series devices, sets 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 *slot/port* **counters 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 *slot/port* **counters errors**	Displays the Layer 3 interface input and output errors.
show interface ethernet *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* **counters detailed** [**all**]	Displays the loopback interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).

Command	Purpose
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.

See the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference for information on these commands.

Related Documents

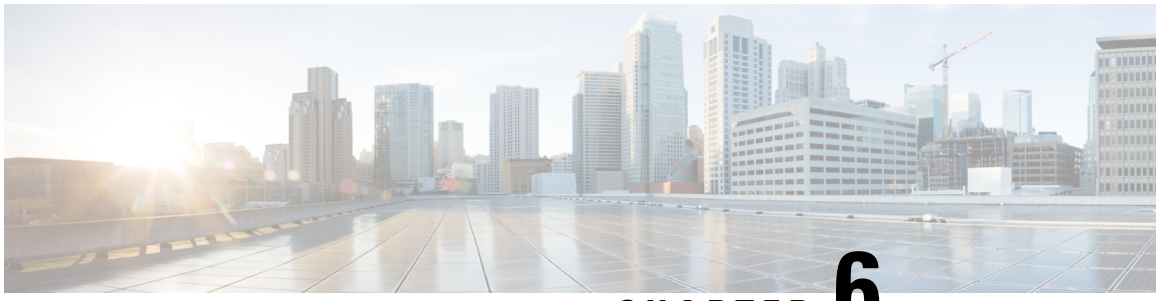
Table 20: Related Documents

Related Topic
Cisco Nexus 7000 Series NX-OS Interfaces Command Reference
Cisco Nexus 7000 Series NX-OS System Management Configuration Guide
Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide
Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 7000 Series Switches, Release 6.x
Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide
Cisco NX-OS Licensing Guide
VLANs, MAC address tables, private VLANs, and the Spanning Tree Protocol. Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide
Cisco Nexus 7000 Series NX-OS FabricPath Command Reference
Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide
Cisco Nexus 7000 Series NX-OS Release Notes

MIBs

Table 21: MIBs

MIBs	MIBs Link
• IF-MIB • CISCO-IF-EXTENSION-MIB • ETHERLIKE-MIB	To locate and download MIBs: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml



CHAPTER 6

Configuring Bidirectional Forwarding Detection

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

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Feature History for BFD

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 22: Feature History for Configuring Basic Interface Parameters

Feature Name	Release	Feature Information
BFD FSA offload on M3	7.3(0)DX(1)	Added support for BFD FSA offload on the M3 line cards.
BFD Support for HSRPv6	7.3(0)D1(1)	Added support for BFD on HSRPv6.

Feature Name	Release	Feature Information
BFD Enhancement to Address Per-link Efficiency	7.3(0)D1(1)	Added support for configuring individual BFD session on every LAG member interface as an enhancement to BFD address per-link efficiency.
BFD on unnumbered Interfaces	7.2(1)D1(1)	Added support for configuring BFD on unnumbered interface.
BFD FSA offload on F3	7.2(1)D1(1)	Added support for BFD FSA offload on the F3 line card.
Support for BFD over Layer 2 over a fabricpath core	7.2(0)D1(1)	Added support for BFD over Layer 2 over a fabricpath core.
Support for BFD over SVI over Fabricbath core	7.2(0)D1(1)	Added support for BFD over SVI over Fabricbath core.
BFD on IPv6 Static Routes	6.2(2a)	Added support for configuring BFD on all IPv6 static routes on an interface.
BFD Interoperability	6.2(2)	Added support for configuring BFD interoperability with Cisco NX-OS and Cisco IOS software.
BFD on IPv6	6.2(2)	Added support for BFD on IPv6.
BFD on OSPFv3	6.2(2)	Added support for BFD on OPSPv3.
BFD on IS-ISv6	6.2(2)	Added support for BFD on IS-ISv6.
BFD on M2 and F2 modules	6.1(1)	Added a note on M2 and F2 module support
BFD Authentication	5.2(1)	Keyed SHA-1 authentication is supported on BFD packets.
BFD for VRRP	5.2(1)	Added support for BFD in VRRP.
BFD	5.0(2)	This feature was introduced.

Information 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.

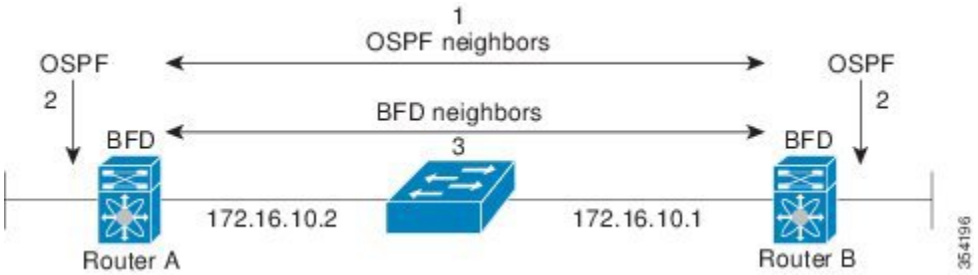
Asynchronous Mode

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 figure below shows how a BFD session is established. The figure shows a simple network with two routers running 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



Detection of Failures

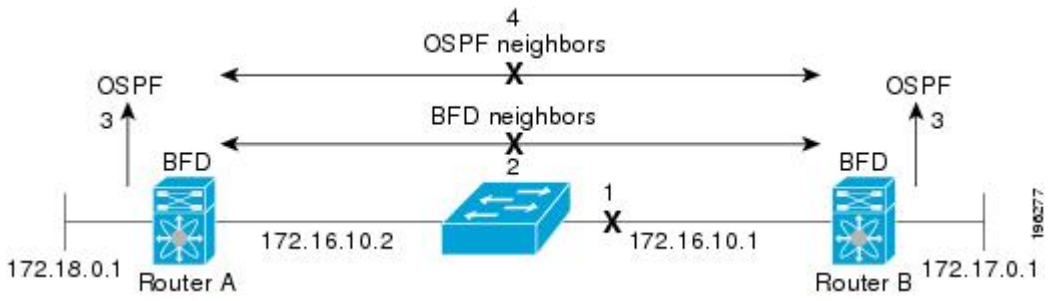
Once a BFD session has been established and timer negotiations are complete, BFD neighbors send BFD control packets that act in the same manner as an IGP hello protocol to detect liveliness, except at a more accelerated rate. BFD detects a failure, but the protocol must take action to bypass a failed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

Note Unicast Reverse Path Forwarding check (uRPF) is disabled by default. If you need to enable it on an interface functioning with BFD, the BFD echo function must be disabled.

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.

From Cisco NX-OS Release 5.2, you can configure SHA-1 authentication of BFD packets.

High Availability

BFD supports stateless restarts and in-service software upgrades (ISSUs). ISSU allows you to upgrade software without impacting forwarding. 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 unless you specifically configure another VDC and VRF. For more information, see the Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide.

BFD Interoperability

This feature enables BFD interoperability between Cisco IOS software, Cisco NX-OS software, and Cisco IOS-XR software.

BFD FSA Offload on F3 Line Card and M3 Line Card

The BFD Fabric Services Accelerator (FSA) Offload on F3 Line Card feature allows the offload of asynchronous and echo BFD transmission (Tx) and reception (Rx) to the network processing unit on the F3 line card. The BFD FSA Offload on F3 Line Card feature improves scale and reduces the overall network convergence time by sending rapid failure detection packets or messages to the routing protocols for recalculating the routing table. You should explicitly enable the BFD FSA Offload on F3 Line Card feature for each VDC, using the **bfd hw-offload-module module-name** command. To disable the feature, use the **no bfd hw-offload-module module-name** command. The feature can be enabled only if there are no active BFD sessions hosted on the line card in that particular VDC.

The BFD FSA Offload feature is introduced on the M3 line card in Cisco Nexus 7000 Series Release 7.3(0)DX(1).

The offload of BFD sessions to the FSA is disabled by default on the F3 line card, and is enabled by default on the M3 line card. BFD sessions can run at 15 ms when the session is offloaded to FSA.

BFD on Unnumbered Interfaces

The Cisco unified fabric needs to support 32 spines with 1024 leaves with bipartite connectivity. In a 32-spine Vinci Fabric, a given leaf will have 32 Layer3 links, one to each spine. Similarly, each spinet will have 1024 Layer3 links, one to each leaf. Typically, each Layer3 link at spine and leaf needs as many IP addresses, which is complex to assign and manage. To reduce the complexity, these Layer3 links derive IP address from a specified loopback interface, and such Layer3 links are referred as unnumbered links. These Layer3 unnumbered links are associated with their respective Router's MAC address. BFD is used for fast failure detection on these links. This necessitates support for BFD over unnumbered interfaces.

You can use either OSPF or ISIS protocols to provide Layer3 connectivity between spines and leaves.

The following BFD sub features are applicable on unnumbered interfaces:

- **Address Family Support**

  BFD clients can bootstrap BFD with either IPv4 or IPv6 address.

- **Echo Support**

  By default, echo function is supported on both IPv4 and IPv6 BFD sessions. However, if BFD IPv6 sessions are bootstrapped with link-local addresses, echo will not be supported.

- **BFD session over unnumbered port-channel**

  Both Logical Mode and Per-link mode sessions are supported. By default, with no configuration on the port-channel, BFD sessions are in the logical mode.

The following configurations are not supported on unnumbered interfaces:

- Switched Virtual Interfaces (SVIs) are not expected to be unnumbered.
- Multipath links between the same set of spines and leaves are not supported.
- Sub interfaces are not expected to be unnumbered and hence sub interface optimization is not supported.

BFD Enhancement to Address Per-link Efficiency

The Bidirectional Forwarding (BFD) enhancement to address per-link efficiency feature enables users to configure individual BFD sessions on every Link Aggregation Group (LAG) member interfaces (as defined in RFC 7130).

With this enhancement BFD sessions will run on each member link of the port-channel. If BFD detects a link failure, the member link is removed from the forwarding table. This mechanism delivers faster failure detection as the BFD sessions are created on individual port-channel interface.

Users can configure RFC 7130 BFD over main port-channel interface, which does bandwidth monitoring over LAG by having one micro-BFD session over each member. If any of the member port goes down, the port is removed from the forwarding table and this prevents black holing of traffic on that member.

Micro BFD sessions (BFD sessions running on member links of the port-channel are called as "micro BFD sessions") are supported for both LACP and non-LACP based-port channels.

Prerequisites for BFD

BFD has the following prerequisites:

- You must enable the BFD feature
- For any client protocols that you want to enable BFD on, you enable BFD in that client protocol.
- Disable Internet Control Message Protocol (ICMP) redirect messages on a BFD-enabled interfaces.
- Disable the IP packet verification check for identical IP source and destination addresses in the default VDC.
- See other detailed prerequisites that are listed with the configuration tasks.
- From Cisco NX-OS Release 6.2(2), BFD for IPv6 is supported.
- To configure the Intermediate System-to-Intermediate System (IS-IS) IPv6 client for BFD, IS-IS must be running on all participating routers. In addition, the baseline parameters for BFD sessions must be configured on the interfaces that run BFD sessions to BFD neighbors.

- To enable BFD interoperability between Cisco IOS Software, Cisco NX-OS software, and Cisco IOS-XR software, use BFD in echo mode. In addition, configure the **no ip redirect** command on all the interfaces that are part of BFD and also on the peer device.

Guidelines and Limitations for BFD

BFD has the following configuration guidelines and limitations:

- BFD supports BFD version 1.
- In Cisco NX-OS Release 6.2(2) and later releases, BFD supports IPv4 and IPv6.
- BFD supports only one session per address family (IPv4 or IPv6), per interface.
- BFD supports keyed SHA-1 authentication from Cisco NX-OS Release 5.2 onwards.
- BFD supports the following Layer 3 interfaces—physical interfaces, port channels, subinterfaces, and VLAN interfaces.
- When configuring BFD for iBGP, ensure to configure BGP neighbor **update-source** command on connected 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.
- When Layer 3 over vPC feature is enabled using the **layer3 peer-router** command, BFD enabled with echo function is not supported on a switched virtual interface (SVIs) using vPC VLANs that are part of a vPC peer-link.
- BFD does not support monitoring of multiple IPv6 next hops in the same subnet on a single interface.
- 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).
- BFD over VLAN interfaces that have member ports only on a N7K-F132XL-15 module are not supported. You should disable BFD over any VLAN with member ports only on a N7K-F132XL-15 module.

Note If you enable BFD at the router level (for example, from OSPF), any BFD sessions over a N7K-F132XL-15 line card will not come up. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for information about OSPF and other routing protocols.

- 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.
- Fabricpath BFD sessions are not supported on port-channel logical interface on any type of a line card.
- 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.

- The BFD echo function is not supported when using IPv6 link-local addresses.
- The following BFD command configurations are not supported during a rollback configuration:
  - **bfd** {**ipv4 | ipv6**} **echo**
  - **bfd** {**ipv4 | ipv6**} **per-link**
  - **bfd hw-offload-module** *module-number*
  - **port-channel bfd track-member-link**
  - **port-channel bfd destination** *destination-ip-address*
- HSRP on IPv4 and IPv6 is supported with BFD.
- If HSRP BFD ALL-INTERFACE is configured, all IPv4 and IPv6 HSRP groups on all interfaces automatically support BFD.
- BFD is not supported for Anycast HSRP.
- Supports only port-channel interfaces that are directly connected between two switches (peer devices) running BFD sessions.
- Supports Layer 3 port channel interfaces in both On mode and LACP mode.
- Supports all Line cards with Layer 3 capabilities.
- IPv6 is not supported.
- Fabric port-channel are not supported.
- vPC is not supported.
- Virtual switch interface over port-channels is not supported.
- Storage VDCs is not supported.
- Echo functionality is not supported for micro-BFD sessions.
- RFC 7130 links cannot be configured along with proprietary links and BFD logical links.
- If RFC 7130 is configured on the main port-channel interface and logical BFD is configured on subinterfaces, the logical BFD session should have lesser aggressive timers than the RFC 7130 BFD sessions.
- Micro BFD sessions are not supported on port-channel sub interfaces.
- FEX interfaces (HIF) ports are not supported.
- If IEFT-BFD is enabled on a port-channel interface, the operational state of port-channel will depend on the minimum micro-BFD session members that are able to establish a session. If the minimum number of links required to have port-channel UP is not met, the port-channel interface is brought down. This in turn brings down the port-channel sub interfaces and the logical BFD sessions.

- If a LACP port-channel has members in hot-standby state and BFD failure link is one of the active link, then hot-standby links might not come up directly. When the active link with BFD failure goes down, the hot-standby member becomes active. This scenario can cause port-channel to go down before the hot-standby can come up.
- BFD per-link is supported for BGP.
- To configure BFD Echo timer to less than 50 milliseconds, you need to configure both the **bfd interval** and the **bfd echo-rx interval** commands.
- Port channel configuration limitations:
  - For Layer 3 port channels used by BFD, you must enable LACP on the port channel. BFD per-link is supported only for EIGRP, ISIS, and OSPF clients.

Note To configure BFD per-link on a port channel, you need to shut down the interface and configure the per-link and then bring up the port-channel again.

  - For Layer 2 port channels used by SVI sessions, you must enable LACP on the port channel.
- SVI limitations:
  - An ASIC reset will cause traffic disruption for other ports. This event could possibly cause SVI sessions on other ports to flap. Some triggers for an ASIC reset are port moves between VDCs, reloading a VDC, or if the carrier interface is a virtual port channel (vPC), BFD is not supported over the SVI interface.
  - 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.

Note 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 re-enable 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.

Default Settings

Table 23: Default BFD Parameters

Parameter	Default
BFD feature	Disabled
Required minimum receive interval	50 milliseconds

Parameter	Default
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
Subinterface optimization	Disabled

Configuring BFD

Configuration Hierarchy

You can configure BFD at the global level and at the interface or subinterface level (for physical interfaces and port channels). The interface or subinterface configuration overrides the global configuration. On supported interfaces, the subinterface-level configuration overrides the interface or port channel configuration unless subinterface optimization is enabled. See the "Optimizing BFD on Subinterfaces" section for more information.

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

For physical ports that are members of a port channel, the member port inherits the master port channel BFD configuration. The member port subinterfaces can override the master port channel BFD configuration, unless subinterface optimization is enabled.

Task Flow for Configuring BFD

Follow these steps to configure BFD:

Step 1: Enabling BFD, on page 116

Step 2: Configuring Global BFD Parameters, on page 117 or Configuring BFD on an Interface, on page 118

Step 3: Configuring BFD for IPv6, on page 123

Enabling BFD

You must enable the BFD feature before you can configure BFD on an interface and protocol within a device (VDC).

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **feature bfd**	Enables the BFD feature.
Step 3	(Optional) switch(config)# **show feature** | **include bfd**	Displays enabled and disabled features.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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

Command	Purpose
no feature bfd	Disables the BFD feature and removes all associated configuration.

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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*	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 15 to 999 milliseconds

	Command or Action	Purpose
		and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3. **Note** Even if the value of the *mintx* argument is configured as 15 ms, if the **bfd hw-offload-module** command is not enabled on the session, the configuration is not applied and the session functions at the default timer value, which is 50 ms.
Step 3	switch(config)# **bfd slow-timer** [*interval*]	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	switch(config-if)# **bfd echo-interface loopback** *interface number*	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.
Step 5	(Optional) switch(config)# **show running-config bfd all**	Displays all the BFD running configurations.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 3	switch(config)# **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*	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 15 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3. **Note** The recommended BFD interval value for logical interfaces (such as Switch Virtual Interface, sub-interface, and so on) is 300 milliseconds and the multiplier default is 3. **Note** Even if the value of the *mintx* argument is configured as 15 ms, if the **bfd hw-offload-module** command is not enabled on the session, the configuration is not applied and the session functions at the default timer value, which is 50 ms.
Step 4	(Optional) switch(config-if)# **bfd authentication keyed-sha1 keyid** *id* **key** *ascii_key*	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	(Optional) switch(config-if)# **show running-config bfd**	Displays the BFD running configuration.
Step 6	(Optional) switch(config-if)#**copy running-config startup-config**	Copies the running configuration to the startup configuration.

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, unless you configure subinterface-level BFD parameters on a member port. In that case, the member port subinterface uses the subinterface BFD configuration if subinterface optimization is not enabled. See the "Optimizing BFD on Subinterfaces" section for more information.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.
- Ensure that you enable LACP on the port channel before you enable BFD.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface port-channel** *number*	Enters port channel configuration mode. Use the **?** keyword to display the supported number range.
Step 3	switch(config-if)# **bfd per-link**	Configures the BFD sessions for each link in the port channel.
Step 4	switch(config-if)# **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*	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 15 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3. **Note** Even if the value of the *mintx* argument is configured as 15 ms, if the **bfd hw-offload-module** command is not enabled on the session, the configuration is not applied and the session functions at the default timer value, which is 50 ms.
Step 5	(Optional) switch(config-if)# **bfd authentication keyed-sha1 keyid** *id* **key** *ascii_key*	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

	Command or Action	Purpose
		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 6	(Optional) switch(config-if)# **show running-config bfd**	Displays the BFD running configuration.
Step 7	(Optional) switch(config-if)#**copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring 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 complaince with RFC 5880. The slow timer becomes the required minimum receive interval if the echo function is enabled.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.
- Configure the BFD session parameters.
- 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 in the default VDC. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for more information about this command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **bfd slow-timer** *echo-interval*	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.

	Command or Action	Purpose
Step 3	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 4	switch(config-if)# **bfd echo**	Enables the echo function. The default is enabled. **Note** To configure BFD Echo timer to less than 50 milliseconds, you need to configure both the **bfd interval** and the **bfd echo-rx interval** commands.
Step 5	(Optional) switch(config-if)# **show running-config bfd**	Displays the BFD running configuration.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Optimizing BFD on Subinterfaces

You can optimize BFD on subinterfaces. BFD creates sessions for all configured subinterfaces. BFD sets the subinterface on which the session comes up first as the master subinterface and that subinterface uses the BFD session parameters of the parent interface. The remaining subinterfaces use the slow timer. If the optimized subinterface session detects an error, BFD marks all subinterfaces on that physical interface as down.

Note BFD hardware offload feature and subinterface optimization must not be used together.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.
- Configure the BFD session parameters.
- Ensure that these subinterfaces connect to another Cisco NX-OS device. This feature is supported on Cisco NX-OS only.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.

	Command or Action	Purpose
Step 3	switch(config-if)# **bfd optimize subinterface**	Optimizes subinterfaces on a BFD-enabled interface. The default is disabled.
Step 4	(Optional) switch(config-if)# **show running-config bfd**	Displays the BFD running configuration.
Step 5	(Optional) switch(config-if)#**copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring BFD for IPv6

Configuring Global BFD Parameters for IPv6

You can specify either the IPv4 or the IPv6 address family when you configure BFD parameters.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **bfd** [**ipv4** | **ipv6**] **interval** [*interval* **min_rx interval multiplier** *interval-multiplier*	Configures the BFD session parameters, in milliseconds, for all BFD session on the device.

Configuring Per Interface BFD Parameters for IPv6

Before you begin

BFD must be enabled on the device.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface type** *number*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 3	switch(config-if)# **bfd** [**ipv4** | **ipv6**] **interval** [*interval* **min_rx interval multiplier** *interval-multiplier*	Configures the BFD session parameters, in milliseconds, for all BFD session on the device.
Step 4	(Optional) switch(config-if)# **bfd** [**ipv4** | **ipv6**] **authentication keyed-sha1 keyid** *id* **key** *ascii_key*	Configures Secure Hash Algorithm 1 (SHA-1) authentication for all BFD sessions on the specified address family.

Configuring BFD on IPv6 Static Routes

You can configure BFD for all IPv6 static routes on an interface.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that BFD is enabled on the devices at each end of the static route.
- Configure the BFD session parameters.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **vrf context** *vrf-name*	Enters VRF configuration mode to configure BFD on an IPv6 static route. • Specify the VRF in which the route is to BFD tracked.
Step 3	switch(config-vrf)# **ipv6 route** *route interface* {*nh-address* | *nh-prefix*}	Creates an IPv6 static route. • Specify the IPv6 address for the route argument. • Use the **?** keyword to display the supported interfaces. • Specify the next hop (nh) address or prefix for this static route.
Step 4	switch(config-vrf)# **ipv6 route static bfd** *network-interface* {*nh-address* | *nh-prefix*}	Enables BFD for all IPv6 static routes on this interface and next hop combination. • Use the **?** keyword to display the supported interfaces. • Specify the next hop (nh) address or prefix for this static route.
Step 5	(Optional) switch(config-vrf)# **show bfd neighbors**	Displays information about BFD neighbors.
Step 6	(Optional) switch(config-vrf)# **show ipv6 route static**	Displays static routes.
Step 7	(Optional) switch(config-vrf)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure BFD on an IPv6 static route between two BFD neighbors:

```
switch(config)# vrf context red
switch(config-vrf)# ipv6 route 1::5/64 ethernet 3/1 2::2
switch(config-vrf)# ipv6 route static bfd ethernet 3/1 2::2 <===Enables BFD on static
routes for the interface/next hop combination.
```

Configuring BFD Echo Mode for IPv6

The BFD echo function is not supported on devices with IPv6 link-local addresses. The echo function is enabled by default. You can disable it for IPv4, IPv6, or all address families.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 3	switch(config-if)# **bfd** [**ipv4** | **ipv6**] **echo**	Enables the echo function for the specified address. The default is enabled. To disable the echo function for the specified address family, use the **no** form of the command.

Configuring a BFD Echo Interface for IPv6

Perform this task to configure the loopback interface as the source address for all echo frames.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface loopback number**	Creates a loopback interface and enters interface configuration mode.
Step 3	switch(config-if)# **ip address** *ip-address mask*	Configures the IP address for the interface.
Step 4	switch(config-if)# **ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits / prefix-length*}	Configures the IPv6 address as the source address for all echo frames.

Configuring BFD Slow Timer for IPv6

Echo mode is enabled by default. You can configure the slow-timer value and disable or enable echo mode for an address family.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 3	switch(config-if)# **bfd** [**ipv4** | **ipv6**] **slow-timer** [*interval*]	Configures the slow timer, in milliseconds, used in the echo function for the specified address family.

Configuring BFD Support for Routing Protocols

Configuring BFD on BGP

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

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.
- Configure the BFD session parameters.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	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	switch(config-router)# **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *as-number*	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	switch(config-router-neighbor)# **bfd**	Enables BFD for this BGP peer.

	Command or Action	Purpose
Step 5	(Optional) switch(config-router-neighbor)# **show running-config bfd**	Displays the BFD running configuration.
Step 6	(Optional) switch(config-router-neighbor)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring BFD on EIGRP

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

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.
- Configure the BFD session parameters.
- Enable the EIGRP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **router eigrp** *instance-tag*	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 command to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.
Step 3	(Optional) switch(config-router-neighbor)# **bfd**	Enables BFD for all EIGRP interfaces.
Step 4	switch(config-router-neighbor)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 5	(Optional) switch(config-if)# **ip eigrp** *instance-tag* **bfd**	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.
Step 6	(Optional) switch(config-if)# **show ip eigrp** [**vrf** *vrf-name*] [**interfaces** *if*]	Displays information about EIGRP. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters.

	Command or Action	Purpose
Step 7	(Optional) switch(config-if)#**copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring BFD on OSPF

You can configure BFD for the Open Shortest Path First version 2 (OSPFv2).

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.
- Configure the BFD session parameters.
- Enable the OSPF feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 3	(Optional) switch(config-router)# **bfd**	Enables BFD for all OSPFv2 interfaces.
Step 4	switch(config-router)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 5	(Optional) switch(config-if)# **ip ospf bfd**	Enables or disables BFD on an OSPFv2 interface. The default is disabled.
Step 6	(Optional) switch(config-if)# **show ip ospf** [**vrf** *vrf-name*] [**interfaces** *if*]	Displays information about OSPF. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters.
Step 7	(Optional) switch(config-if)#**copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring BFD on OSPFv3

BFD supports Open Shortest Path First version 3 (OSPFv3), which is a link-state routing protocol for IPv6 networks.

There are two methods for enabling BFD support for OSPFv3:

- You can enable BFD for all of the interfaces for which OSPFv3 is routing by entering the bfd command in router configuration mode. You can disable BFD support on individual interfaces by entering the **ospfv3 bfd disable** command in interface configuration mode.
- You can enable BFD for a subset of the interfaces for which OSPFv3 is routing by entering the **ospfv3 bfd** command in interface configuration mode.

Note OSPF will only initiate BFD sessions for OSPF neighbors that are in the FULL state.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 3	switch(config-if)# **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*	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 15 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3. **Note** Even if the value of the *mintx* argument is configured as 15 ms, if the **bfd hw-offload-module** command is not enabled on the session, the configuration is not applied and the session functions at the default timer value, which is 50 ms.
Step 4	switch(config-if)# **end**	Exits interface configuration mode and returns to global configuration mode.

Configuring BFD for OSPFv3 for All Interfaces

Before you begin

OSPFv3 must be running on all participating devices. The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **router ospfv3** *process-id*	Configures an OSFPv3 routing process.
Step 3	switch(config-router)# **bfd**	Enables BFD for all interfaces participating in the routing process.
Step 4	switch(config-router)# **exit**	Enter this command twice to return to EXEC mode.
Step 5	switch# **show bfd neighbors** [**details**]	Displays a line-by-line listing of existing BFD adjacencies.
Step 6	switch# **show ospfv3** [*process-id*]	Displays general information about OSPFv3 routing processes.

Configuring BFD for OSPFv3 on One or More Interfaces

Before you begin

OSPFv3 must be running on all participating devices. The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 3	switch(config-if)# **ospfv3 bfd** [**disable**]	Enables BFD on a per-interface basis for one or more interfaces associated with the OSPFv3 routing process.
Step 4	switch(config-if)# **exit**	Enter this command twice to return to EXEC mode.
Step 5	switch# **show bfd neighbors** [**details**]	Displays a line-by-line listing of existing BFD adjacencies.
Step 6	switch# **show ospfv3** [*process-id*]	Displays general information about OSPFv3 routing processes.

Configuring BFD on IS-IS

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

Note Fabricpath BFD sessions are not supported on port-channel logical interface on any type of a line card.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.
- Configure the BFD session parameters.
- Enable the IS-IS feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **router isis** *instance-tag*	Creates a new IS-IS instance with the configured instance tag.
Step 3	(Optional) switch(config-router)# **bfd**	Enables BFD for all IS-IS interfaces.
Step 4	switch(config-router)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 5	(Optional) switch(config-if)# **isis bfd**	Enables or disables BFD on an IS-IS interface. The default is disabled.
Step 6	(Optional) switch(config-if)# **show isis** [**vrf** *vrf-name*] [**interfaces** *if*]	Displays information about IS-IS. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters.
Step 7	(Optional) switch(config-if)#**copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring BFD on IS-ISv6

When BFD support is configured with IS-IS as a registered protocol with BFD, IS-IS receives forwarding path detection failure messages from BFD. BFD support for IS-IS can be configured in either router address-family configuration mode or interface configuration mode. IS-IS IPv6 runs in single-topology mode.

IS-IS BFD supports both IPv4 and IPv6 on the same adjacency for single-topology mode. If BFD is enabled for both IPv4 and IPv6, IS-IS sends two BFD session creation requests to BFD. For single-topology mode, the IS-IS adjacency state can only be up if both BFD sessions are up. If either of the BFD sessions is down, the associated IS-IS adjacency state is also down.

When IS-IS BFD IPv6 is disabled on an interface, IS-IS removes related BFD sessions for IPv6 from the adjacent device. When the IS-IS adjacency entry is deleted, all BFD sessions are also deleted. IS-IS requests BFD to remove each BFD session that it has requested when any of the following events occur:

- The IS-IS instance is deleted or un-configured.
- The IS-IS adjacency entry is deleted.
- IS-IS BFD is disabled on the next hop interface for an address-family.

Configuring IS-IS IPv6 Client Support on an Interface

IS-IS requests a BFD session for the interface and the IPv6 address of the neighboring device when all of the following conditions are met:

- An IS-IS adjacency entry exists.
- The Address Family Identifier (AFI) specific peer interface address is known.
- IS-IS BFD is enabled for that AFI on an interface.
- IS-IS is enabled for that AFI on the local interface.
- If the neighboring device supports RFC 6213, BFD must be enabled for the specified Network Layer Protocol Identifier (NLPID).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 3	switch(config-if)# **isis ipv6 bfd**	Enables IPv6 BFD on a specific interface that is configured for IS-IS.
Step 4	switch(config-if)# **end**	Exits interface configuration mode and returns to global configuration mode.
Step 5	(Optional) switch(config)# **show isis interface** *type number*	Displays interface information about IS-IS.

Configuring IS-IS IPv6 Client Support for BFD on All Interfaces

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **router isis** *process-id*	Enables the IS-IS routing protocol and enters router configuration mode.
Step 3	(Optional) switch(config-router)# **metric-style transition**	Configures a device that is running IS-IS so that it generates and accepts only new style, type, length, value objects (TLVs).
Step 4	switch(config-router)# **address-family ipv6 unicast**	Enters address family configuration mode for configuring IS-IS routing session that use standard IPv6 address prefixes.
Step 5	switch(config-router-af)# **bfd**	Enables BFD for all interfaces that are participating in the routing process.

	Command or Action	Purpose
Step 6	switch(config-router-af)# **end**	Exits address family configuration mode and returns to global configuration mode.
Step 7	(Optional) switch(config)# **show isis** [*process-id*]	Displays interface information about IS-IS.

Configuring FabricPath BFD on a Specific Interface

Before you begin

- Enable the BFD feature.
- Configure the BFD session parameters.
- The ISIS feature is enabled by default when entering the **feature-set fabricpath** command.
-
-

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] bfd fabricpath encap-ce**	Enables the user to choose an encapsulation mode for the L2BFD frames on a per-session basis. On enabling the command, it sends out the frames without Fabricpath encapsulation. The default mode is to send frames with Fabricpath encapsulation.
Step 3	switch(config-if)# **fabricpath isis bfd**	Enables the FabricPath BFD on the interface.

Example

This example shows how to configure FabricPath BFD on a specific interface:

```
switch# configure terminal
switch(config)# [no] bfd fabricpath encap-ce
switch(config-if)# fabricpath isis bfd
```

Configuring FabricPath BFD on All IS-IS Interfaces

Before you begin

- Ensure that you are in the correct VRF.
- Enable the BFD feature.
- Configure the BFD session parameters.

- The ISIS feature is enabled by default when entering the **feature-set fabricpath** command.
-
-

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **fabricpath domain default**	Enters the global FabricPath Layer 2 Intermediate System, to Intermediate System (IS-IS) configuration mode.
Step 3	switch(config-fabricpath-isis)# **bfd**	Enables FabricPath BFD on all IS-IS interfaces.

Example

This example show how to configure FabricPath BFD on all IS-IS interfaces:

```
switch# configure terminal
switch(config)# fabricpath domain default
switch(config-fabricpath-isis)# 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 expiry and takes over as the active HSRP router.

Use the **show hsrp bfd-sessions** command to display the HSRP BFD session information for all the interfaces.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.
- Configure the BFD session parameters.
- Enable the HSRP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	(Optional) switch(config)# **hsrp bfd all-interfaces**	Enables or disables BFD on all HSRP interfaces. The default is disabled.

	Command or Action	Purpose
Step 3	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 4	(Optional) switch(config-if)# **hsrp bfd**	Enables or disables BFD on an HSRP interface. The default is disabled.
Step 5	(Optional) switch(config-if)# **show running-config hsrp**	Displays the HSRP running configuration.
Step 6	(Optional) switch(config-if)#**copy running-config startup-config**	Copies the running configuration to the startup configuration.

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** will show this event as BFD@Act-down or BFD@Sby-down.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.
- Configure the BFD session parameters.
- Enable the VRRP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 3	switch(config-if)# **vrrp** *group-no*	Specifies the VRRP group number.
Step 4	switch(config-if)# **vrrp bfd** *address*	Enables or disables BFD on an VRRP interface. The default is disabled.
Step 5	(Optional) switch(config-if)# **show running-config vrrp**	Displays the VRRP running configuration.
Step 6	(Optional) switch(config-if)#**copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring BFD on PIM

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

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.
- Enable the PIM feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **ip pim bfd**	Enables BFD for PIM.
Step 3	(Optional) switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 4	switch(config-if)# **ip pim bfd-instance** [**disable**]	Enables or disables BFD on a PIM interface. The default is disabled.
Step 5	(Optional) switch(config-if)# **show running-config pim**	Displays the PIM running configuration.
Step 6	(Optional) switch(config-if)#**copy running-config startup-config**	Copies the running configuration to the startup configuration.

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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	(Optional) switch(config)# **vrf context** *vrf-name*	Enters VRF configuration mode.

	Command or Action	Purpose
Step 3	switch(config-vrf)# **ip route** *route interface* {*nh-address* | *nh-prefix*}	Creates a static route Use the **?** keyword to display the supported interfaces.
Step 4	switch(config-vrf)# **ip route static bfd interface** {*nh-address* | *nh-prefix*}	Enables BFD for all static routes on an interface. Use the **?** keyword to display the supported interfaces.
Step 5	(Optional) switch(config-vrf)# **show ip route static** [**vrf** *vrf-name*]	Displays the static routes.
Step 6	(Optional) switch(config-vrf)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring BFD on MPLS TE Fast Reroute

MPLS Traffic Engineering (TE) uses BFD accelerate the detection of node failures and to provide fast forwarding path failure detection times. BFD for MPLS TE fast reroute is configured automatically when you enable the fast reroute on a tunnel. See the “Configuring MPLS TE Fast Reroute Link and Node Protection” chapter in the Cisco Nexus 7000 Series NX-OS MPLS Configuration Guide for more information.

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:

Table 24: Disabling BFD on an Interface

Command	Purpose
ip eigrp *instance-tag* **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	Disables BFD on an OSPFv2 interface.
isis bfd disable	Disables BFD on an IS-IS interface.

Configuring BFD on Unnumbered Interfaces

The following are some basic switch configurations to set up BFD over unnumbered interfaces:

1. Configure the ethernet interface that is unnumbered.
2. Configure the loopback interface from which the IP address is derived for the unnumbered interface.
3. Configure ISIS or OSPF with VRF on the router.

Procedure

Step 1 The following are the steps to configure ISIS on the Ethernet interface that is unnumbered:

a) Enter the global configuration mode:

switch# **configure terminal**

b) Enter the interface config mode:

switch(config)# **interface ethernet** *slot* / *port*

c) Configure the interface medium as point to point:

switch(config-if)# **medium p2p**

d) Enable IP processing on loopback interface:

switch(config-if)# **ip unnumbered** *instance*

e) Configure the ISIS metric to calculate the cost of routing at different levels:

switch(config-if)# **isis metric** {*metric-value* | **maximum**} [**level-1** | **level-2**]

f) Configure the type of adjacency:

switch(config-if)# **isis circuit-type** [**level-1** | **level-1-2** | **level-2-only**]

g) Configure an IS-IS routing process for the IP on the configured interface and attach an area designator to the routing process:

switch(config-if)#**ip router isis** *area-tag*

h) Enable BFD

switch(config-if)#**isis bfd** *instance*

i) Exit the config mode:

switch(config-if)#**end**

Step 2 The following are the steps to configure the loopback interface from which the IP address for the unnumbered interface is derived:

a) Create a loopback interface and enter the interface config mode:

switch(config)# **interface loopback** *instance*

b) Configure an IP address for this loopback interface:

switch(config-if)#**ip address** *address*

c) Configure an IS-IS routing process for the IP on the configured interface and attach an area designator to the routing process:

switch(config-if)#**ip router isis** *area-tag*

Example

This example shows how to configure BFD on unnumbered ethernet interface with ISIS protocol:

```
interface Ethernet1/2
  medium p2p
  ip unnumbered loopback1
  isis metric 10 level-1
  isis circuit-type level-1
  ip router isis 100
  isis bfd
  no shutdown
router isis 100
  net 49.0001.0000.0000.000a.00
  is-type level-1
    address-family ipv6 unicast
```

This example shows how to configure BFD over unnumbered interface with OSPF and VRF:

```
vrf context vrf3
interface Ethernet1/14
  medium p2p
  vrf member vrf3
  ip unnumbered loopback1
  ip router ospf 10 area 0.0.0.0
  no shutdown

interface loopback1
  vrf member vrf3
  ip address 10.1.1.2/32
line vty
router ospf 10
  bfd
  vrf vrf3
  bfd
```

Configuring BFD Interoperability

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *int-if*	Enters interface configuration mode. Use the **?** keyword to display the supported interfaces.
Step 3	switch(config-if)# **ip ospf bfd**	Enables BFD on an OSPFv2 interface. The default is disabled. You can enable BFD of any of the supported protocols.
Step 4	switch(config-if)# **no ip redirect**	Prevents the device from sending redirects.

	Command or Action	Purpose
Step 5	switch(config-if)# **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*	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 15 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3. **Note** Even if the value of the *mintx* argument is configured as 15 ms, if the **bfd hw-offload-module** command is not enabled on the session, the configuration is not applied and the session functions at the default timer value, which is 50 ms.
Step 6	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

BFD is supported on switched virtual interfaces configured on L3 switches. The ports connecting two such switches can be connected in the following modes:

- Trunk– The ports of two such devices can be connected using classic Ethernet and configured in the trunk mode.
- Fabric– The ports of two such devices can be connected using a fabric path core and configured in the fabricpath mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *int-if*	Creates a dynamic Switch Virtual Interface (SVI).
Step 3	switch(config-if)# **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*	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 15 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.

	Command or Action	Purpose
		Note Even if the value of the *mintx* argument is configured as 15 ms, if the **bfd hw-offload-module** command is not enabled on the session, the configuration is not applied and the session functions at the default timer value, which is 50 ms.
Step 4	switch(config-if)# **no ip redirect**	Prevents the device from sending redirects.
Step 5	switch(config-if)# **ip address** *ip-address/length*	Configures an IP address for this interface.
Step 6	switch(config-if)# **ip ospf bfd**	Enables BFD on an OSPFv2 interface. The default is disabled.
Step 7	switch(config-if)# **exit**	Exits interface configuration mode and returns to EXEC mode.
Step 8	switch(config)# **interface** *int-if*	Configures the port connected to another switch configured as described in the steps above.
Step 9	Do one of the following: • switch(config-if)# **switchport mode trunk** • switch(config-if)# **switchport mode fabricpath**	The interface is configured as a classic ethernet trunk port or a fabric path port.
Step 10	switch(config-if)# **end**	Returns to privileged EXEC mode.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)# **interface** *type number.subinterface-id*	Enters port channel configuration mode. Use the **?** keyword to display the supported number range.
Step 3	switch(config-if)# **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*	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 15 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.

	Command or Action	Purpose
		Note Even if the value of the *mintx* argument is configured as 15 ms, if the **bfd hw-offload-module** command is not enabled on the session, the configuration is not applied and the session functions at the default timer value, which is 50 ms.
Step 4	switch(config-if)# **no ip redirect**	Prevents the device from sending redirects.
Step 5	switch(config-if)# **ip ospf bfd**	Enables BFD on an OSPFv2 interface. The default is disabled.
Step 6	switch(config-if)# **exit**	Exits interface configuration mode and returns to EXEC mode.

Verifying BFD Interoperability in a Cisco Nexus 7000 Series Device

Table 25: Verifying BFD Interoperability in a Cisco Nexus 7000 Series Device

Command	Purpose
show bfd neighbors [**details**]	Displays a line-by-line listing of existing BFD adjacencies.

These examples show how to verify BFD interoperability in a Cisco Nexus 7000 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 BFD FSA Offload on F3 and M3 Modules

To display BFD configuration information, use one of the following commands:

Table 26: Verifying BFD FSA Offload on F3 and M3 Modules

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.
show bfd neighbors	Displays information about BFD neighbors.
show bfd neighbors neighbors	Displays information about BFD neighbor details.

This example shows how to verify BFD FSA Offload on F3 and M3 line cards feature. The output contains an asterisk (‘*’) symbol displayed beside the offloaded sessions.

```
switch# show bfd neighbors

OurAddr   NeighAddr  LD/RD       RH/RS   Holdown(mult) State  Int   Vrf
*10.2.2.2   10.2.2.1   1124073477/1  Up      N/A(3)        Up     Eth1/45   default

10.1.1.2   10.1.1.1    1124073478/1   Down    N/A(3)        Down   Eth1/46  default

*10.3.3.2  10.3.3.1    1124073479/1   Up      N/A(3)        Up     Eth1/47  default

10.4.4.2   10.4.4.1    1124073480/1   Down    N/A(3)        Down   Eth1/48  default
```

This example shows how to verify BFD FSA Offload on a F3 and M3 line cards. The output has a field indicating that a particular session is offloaded.

```
switch# show bfd neighbors details

OurAddr   NeighAddr LD/RD                RH/RS  Holdown(mult) State Int     Vrf
*10.2.2.1  10.2.2.2   1107296257/1124073474 Up    4880(3)       Up    Eth1/2  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: 3
Holdown (hits): 6000 ms (0), Hello (hits): 2000 ms (1142)
Rx Count: 1139, Rx Interval (ms) min/max/avg: 0/5132/1693 last: 1119 ms ago
Tx Count: 1142, Tx Interval (ms) min/max/avg: 1689/1689/1689 last: 1120 ms ago
Registered protocols:  hsrp_engine
Uptime: 0 days 0 hrs 32 mins 3 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.: 1124073474    - Your Discr.: 1107296257
             Min tx interval: 50000   - Min rx interval: 2000000
             Min Echo interval: 50000 - Authentication bit: 0
Hosting LC: 1, Down reason: None, Reason not-hosted: None, Offloaded: Yes
```

For detailed information about the fields in the output from these commands, see the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference.

Configuring Micro BFD Sessions

Configuring Port Channel Interface

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable the BFD feature. For more information, see "Enabling BFD, on page 116"

Procedure

Step 1 Configure interface port-channel:

switch(config-if)# **interface port-channel** *port-number*

Enters port channel configuration mode. Use the **?** keyword to display the supported number range.

Step 2 Configure interface as Layer 3 port-channel:

switch(config-if)# **no switchport**

(Optional) Configuring BFD Start Timer

Procedure

Configure BFD start timer for a port-channel:

switch(config-if)# **port-channel bfd start** *60*

Note The default value is infinite (that is no timer is running). The range of BFD Start Timer value for port-channel is from 60 to 3600 seconds. For start timer to work, configure start timer value before completing the port-channel BFD configurations (that is before port-channel bfd track-member-link and port-channel bfd destination are configured for Layer 3 port-channel interface with active members).

Enabling IETF Per-Link BFD

Procedure

Enable IETF BFD on port-channel interface:

switch(config-if)# **port-channel bfd track-member-link**

Configuring BFD Destination Address

Procedure

Configure an IPv4 address to be used for BFD sessions on member links:

switch(config-if)# **port-channel bfd destination** *ip-address*

Related Documents

Table 27: Related Documents

Related Topic
Cisco Nexus 7000 Series NX-OS Interfaces Command Reference
Cisco Nexus 7000 Series NX-OS System Management Configuration Guide
Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide
Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 7000 Series Switches, Release 6.x
Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide
Cisco NX-OS Licensing Guide

Related Topic
VLANs, MAC address tables, private VLANs, and the Spanning Tree Protocol. Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide
Cisco Nexus 7000 Series NX-OS FabricPath Command Reference
Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide
Cisco Nexus 7000 Series NX-OS Release Notes

RFCs

Table 28: RFCs

RFCs	Title
RFC 5880	*Bidirectional Forwarding Detection (BFD)*
RFC 5881	*BFD for IPv4 and IPv6 (Single Hop)*

Verifying the BFD Configuration

To verify BFD configuration, use one of the folowing commands:

Table 29: Verifying the BFD Configuration

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.

To verify micro BFD session configurations, use one of the following commands:

Table 30: Verifying Micro BFD Session Configurations

Command	Purpose
show port-channel summary	Display the port-channel and port-channel member operational state.
show bfd neighbors	Display micro-BFD sessions on port-channel members.
show bfd neighbors detail	Display BFD session for a port channel interface, and the associated micro-BFD sessions on members.
show tech-support bfd	Display technical support information for BFD.

show tech-support lacp all	Display the technical support information for Ethernet Port Manager, Ethernet Port-channel Manager and LACP.
show running-config interface port-channel *port-channel-number*	Display running configuration information of the port-channel interface.

For detailed information about the fields in the output from these commands, see the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference.

Monitoring BFD

To display BFD monitoring, use the following commands :

Table 31: Monitoring 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.
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.

For detailed information about the fields in the output from these commands, see the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference.

Configuration Examples for BFD

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

The fields shown below are self-explanatory.

```
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:

The fields shown below are self-explanatory.

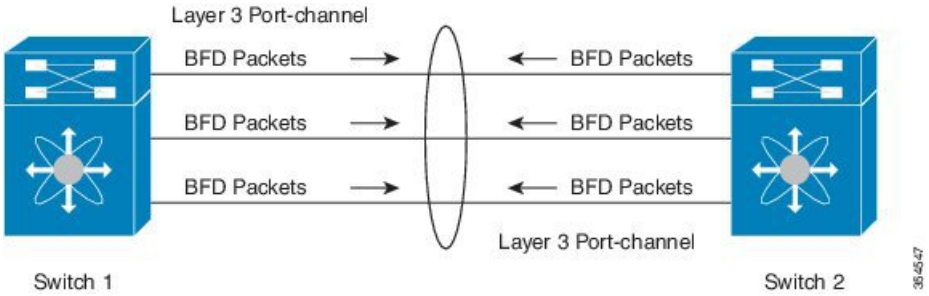
```
feature bfd
feature eigrp
bfd interval 100 min_rx 100 multiplier 4
```

```
router eigrp Test2
 bfd
```

The following example shows how to configure micro BFD sessions :

The following topology is used in the example that follows:

Figure 8: Configuring a Micro BFD Session



This example shows how to configure a micro BFD session on switch 1

The fields shown below are self-explanatory.

```
feature bfd
configure terminal
    interface port-channel 10
       port-channel bfd track-member-link
       port-channel bfd destination 10.1.1.2
       port-channel bfd start 60
       ip address 10.1.1.1/24
```

This example shows how to configure a micro BFD session on switch 2

The fields shown below are self-explanatory.

```
feature bfd
configure terminal
     interface port-channel 10
           port-channel bfd track-member-link
           port-channel bfd destination 10.1.1.1
           port-channel bfd start 60
             ip address 10.1.1.2/24
```

This example displays the show output of the **show port-channel summary** command.

The fields shown below are self-explanatory.

```
switch(config-if-range)# show port-channel summary

Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        b - BFD Session Wait
        S - Switched    R - Routed
        U - Up (port-channel)
        M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
```

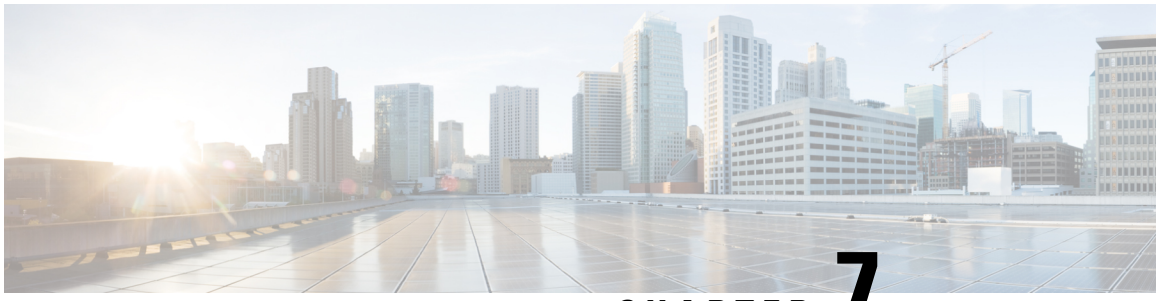
```
--------------------------------------------------------------------
10     Po10(RD)     Eth      NONE     Eth7/26(b)   Eth7/27(b) Eth7/28(b)
```

This example displays the show output of the **show bfd neighbors detail** command.

The fields shown below are self-explanatory.

```
switch(config-if-range)# show bfd neighbors detail


OurAddr   NeighAddr   LD/RD         RH/RS  Holdown(mult)  State    Int   Vrf
10.1.1.1  10.1.1.2    1107296277/0  Down    N/A(3)        Down     Po10  default
Session state is Down and not using echo function
Local Diag: 1, Demand mode: 0, Poll bit: 0, Authentication: None
MinTxInt: 0 us, MinRxInt: 0 us, Multiplier: 0
Received MinRxInt: 0 us, Received Multiplier: 0
Holdown (hits): 0 ms (0), Hello (hits): 0 ms (0)
Rx Count: 0, Rx Interval (ms) min/max/avg: 0/0/0 last: 0 ms ago
Tx Count: 0, Tx Interval (ms) min/max/avg: 0/0/0 last: 0 ms ago
Registered protocols:  eth_port_channel
Downtime: 0 days 0 hrs 0 mins 4 secs
Last packet: Version: 0               - Diagnostic: 0
             State bit: AdminDown     - Demand bit: 0
             Poll bit: 0              - Final bit: 0
             Multiplier: 0            - Length: 24
             My Discr.: 0             - Your Discr.: 0
             Min tx interval: 0       - Min rx interval: 0
             Min Echo interval: 0     - Authentication bit: 0
Hosting LC: 0, Down reason: Control Detection Time Expired, Reason not-hosted: SUCCESS,
Offloaded: No
Parent session, please check port channel config for member info
```



CHAPTER 7

Configuring Port Channels

This chapter describes how to configure port channels.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Feature History for Configuring Port Channels

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Release	Feature Information
LACP Fast Timers	8.2(4)	Improved the validation for the number of interfaces with LACP Fast Timers.

Feature Name	Release	Feature Information
GPRS Tunneling Protocol (GTP) Load Balance Support on M3-Series Modules	7.3(0)DX(1)	Improved port-channel and ECMP load balancing for GTP traffic with M3 modules.
Random Load Balance (Port Channel)	7.3(0)D1(1)	Added support for Random Load Balancing on port channels. Added the **random** keyword to **port-channel load-balance** command to improve load balancing across port channels.
DisplayPpolicy Errors on Interfaces and VLANs	6.2(2)	Added the **show interface status error policy** command.
Prevent Traffic-Drop During Bi-Directional Flow on F2 or F2e Modules	6.2(2)	Added the **asymmetric** keyword to **port-channel load-balance** command to improve load balancing across port channels.
Result Bundle Hash Load Balancing	6.1(3)	Support for the RBH modulo mode to improve load balancing across port channels.
Minimum Links for FEX Fabric Port Channel	6.1(3)	This feature was introduced.
Port Channels Hash Distribution	6.1(1)	Support for port channel hash distribution fixed and adaptive mode.
Load-Balancing Supports F2 Modules	6.0(1)	Added support for F2 modules on load-balancing across port channels.
Port Channels	5.2(1)	Support increased to 528 port channels.
Minimum Links and Maxbundle for LACP	5.1(1)	This feature was introduced.
Port Channels	4.2(1)	Support increased to 256 port channels.
Port Channels	4.0(1)	This feature was introduced.

Information About Port Channels

A port channel is an aggregation of multiple physical interfaces that creates a logical interface. You can bundle up to 8 individual active links into a port channel to provide increased bandwidth and redundancy. 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. Port channeling also load balances traffic on the M series module and across these physical interfaces. The port channel stays operational as long as at least one physical interface within the port channel is operational.

Note

- From Cisco NX-OS Release 5.1, you can bundle up to 16 active links into a port channel on the F-series module.

You cannot configure a shared interface to be part of a port channel. See the Cisco NX-OS FCoE Configuration Guide for Cisco Nexus 7000 and Cisco MDS 9000 for more information about shared interfaces.

You can bundle up to 8 ports into a static port channel without using any aggregation protocol. On the M-series module, you can bundle up to 8 active and 8 standby on the M-series module and up to 16 ports on the F Series module. Starting from Cisco NX-OS Release 8.3(1), you can bundle up to 16 active links on M3 modules.

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.

Note This 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.

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. After you create a Layer 3 port channel, you can add an IP address to the port-channel interface and create subinterfaces on the Layer 3 port channel. You cannot combine Layer 2 and Layer 3 interfaces in the same port channel.

From Cisco NX-OS Release 4.2, you can apply port security to port channels. See the Cisco Nexus 7000 Series NX-OS Security Configuration Guide for information about port security. All ports in the port channel must be in the same virtual device context (VDC); you cannot configure port channels across VDCs.

You can also change the port channel from Layer 3 to Layer 2.

Any configuration changes that you apply to the port channel are applied to each member interface of that port channel. For example, if you configure Spanning Tree Protocol (STP) parameters on the port channel, the Cisco NX-OS software applies those parameters to each interface in the port channel.

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 create subinterfaces on a Layer 3 port channel, even though a subinterface is part of the logical port-channel interface. See the "Subinterfaces" section for more information about port-channel subinterfaces.

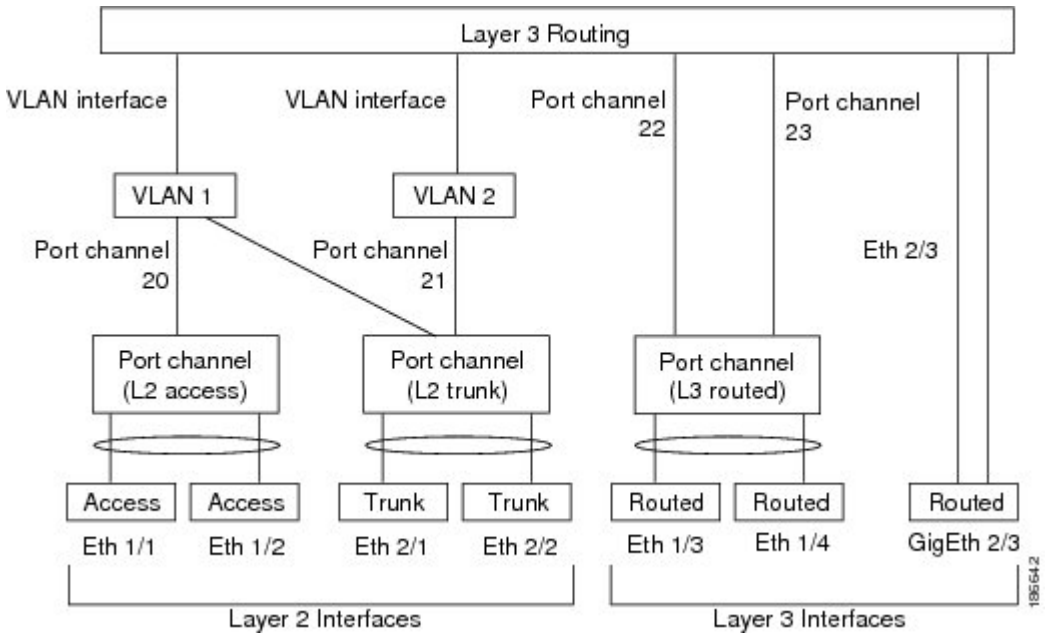
You can use static port channels, with no associated aggregation protocol, for a simplified configuration. For more flexibility, you can use the Link Aggregation Control Protocol (LACP), which is defined in IEEE 802.3ad. When you use LACP, the link passes protocol packets. You cannot configure LACP on shared interfaces.

See the "LACP" section for information about LACP.

Port-Channel Interfaces

The figure below shows port-channel interfaces.

Figure 9: 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 and might have subinterfaces.

From Cisco NX-OS Release 4.2(1), 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 7000 Series NX-OS Layer 2 Switching Configuration Guide for information about configuring static MAC addresses on Layer 3 port channels.

Basic Settings

You can configure the following basic settings for a 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.

- Interface Description—Use this setting to provide a unique name for each interface so that you can quickly identify the interface when you are looking at a listing of multiple interfaces.
- Duplex—By default, each interface autonegotiates its duplex mode with the other interface, but you can change these settings. If you change the settings, be sure to use the same duplex mode setting on both interfaces, or use autonegotiation for at least one of the interfaces.
- Flow control—Use this setting to allow flow control to work between two ports. You must set the corresponding receive and send flow control parameters for both ports as enabled or desired.
- IP addresses—Both IPv4 and IPv6
- Maximum Transmission Unit (MTU)—Use this setting to specify the maximum frame size that an Ethernet port can process.
- Shutdown—Use this setting to bring down or up an interface.
- Speed—By default, each interface autonegotiates its speed mode with the other interface, but you can change these settings. If you change the settings, be sure to use the same speed mode setting on both interfaces, or use autonegotiation for at least one of the interfaces.

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:

- (Link) speed capability
- Access VLAN
- Allowed VLAN list
- Check rate mode
- Duplex capability
- Duplex configuration
- Flow-control capability
- Flow-control configuration
- Layer 3 ports—(Cannot have subinterfaces)
- MTU size
- Media type, either copper or fiber
- Module Type
- Network layer
- Port mode
- SPAN—(Cannot be a SPAN source or a destination port)

- Speed configuration
- Storm control
- Tagged or untagged
- Trunk native VLAN

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 (v4 and v6)
- 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

If you configure subinterfaces for the port-channel interface and remove a member port from the port channel, the configuration of the port-channel subinterface does not propagate to the member ports.

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 or on specified modules. The per-module configuration takes precedence over the load-balancing configuration for the entire device. You can configure one load-balancing mode for the entire device, a different mode for specified modules, and another mode for the other specified modules. 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.

Note The default load-balancing mode for Layer 3 interfaces is the source and destination IP address, 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. For additional information about this command, see the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference.

F1-series modules do not support load balancing of non-IP traffic based on a MAC address. If ports on an F1-series module are used in a port channel and non-IP traffic is sent over the port channel, Layer 2 traffic might get out of order. From Cisco NX-OS Release 6.0(1), load balancing supports F2 modules.

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

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.

Note You cannot configure load balancing using port channels per virtual device context (VDC). You must be in the default VDC to configure this feature; if you attempt to configure this feature from another VDC, the system displays an error.

You can configure load balancing either by the entire system or by specific modules, regardless of the VDC. The port-channel load balancing is a global setting across all VDCs.

If the ingress traffic is Multiprotocol Label Switching (MPLS) traffic, the software looks under the labels for the IP address on the packet.

Multicast traffic inherits the same port-channel load balancing configuration as unicast traffic. This is applicable for both system-wide and module-specific load balancing configurations.

Note Devices that run Cisco IOS can optimize the behavior of the member ports of ASICs if a failure of a single member occurred if you enter the **port-channel hash-distribution** command. The Cisco Nexus 7000 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 either for the entire device or on a per-module basis. See the *Cisco Nexus 7000 Series NX-OS Interfaces Command Reference* for information about this command.

Cisco NX-OS Release 6.1(3) supports a new Result Bundle Hash (RBH) mode to improve load balancing on port-channel members on Cisco Nexus 7000 M Series I/O XL modules and on F Series modules. With the new RBH modulo mode, the RBH result is based on the actual count of port-channel members.

Symmetric Hashing

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 ip
- dst ip rotate
- dst ip
- src ip rotate
- src-dst ip
- src ip-l4port
- dst ip-l4port rotate
- dst ip-l4port
- src ip-l4port rotate
- src-dst ip-l4port-vlan
- dst ip-vlan
- src ip-vlan rotate
- src-dst ip-vlan
- src l4port
- dst l4port rotate
- dst l4port
- src l4port rotate
- src-dst l4port
- src mac
- dst mac rotate
- dst mac
- src mac rotate
- src-dst mac

Random Load Balancing (Port Channel)

Random load balancing on port channels is a software solution that enables better port-link bandwidth utilization for GPRS Tunneling Protocol (GTP) over IP-UDP packets. The existing M1, M2, F1, F2 and F2e line card hardware does not have the capability to perform random load balancing and hence, this software solution helps in load balancing and optimizing the port channels bandwidth. Random load balancing is supported only on F3 series line cards. Random load balancing is applicable on all types of traffic and is effective on egress ports of Layer 3 traffic. The Cisco NX-OS software does random load balancing of all traffic across all interfaces in a port channel by using polynomial scheme.

LACP

LACP allows you to configure up to 16 interfaces into a port channel. A maximum of 8 interfaces can be active, and a maximum of 8 interfaces can be placed in a standby state on the M-series modules.

From Cisco NX-OS Release 5.1, you can bundle up to 16 active links into a port channel on the F-Series module.

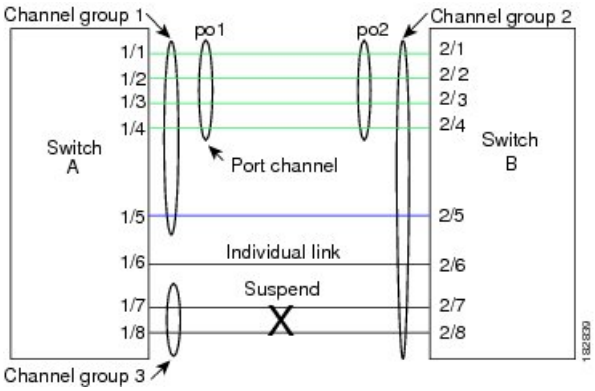
Note You must enable LACP before you can use LACP. By default, LACP is disabled.

See the "Enabling LACP" section for information about enabling LACP.

From Cisco NX-OS Release 4.2, the system automatically takes a checkpoint before disabling the feature, and you can roll back to this checkpoint. See the Cisco Nexus 7000 Series NX-OS System Management Configuration Guide for information about rollbacks and checkpoints.

The figure below shows how individual links can be combined into LACP port channels and channel groups as well as function as individual links.

Figure 10: Individual Links Combined into a Port Channel



With LACP, you can bundle up to 16 interfaces in a channel group. If the channel group has more than 8 interfaces, the remaining interfaces are in hot standby for the port channel associated with this channel group on the M-series modules.

From Cisco NX-OS Release 5.1, you can bundle up to 16 active links into a port channel on the F-series module.

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 active or passive. You can configure either 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.

Table 32: Port-Channel Modes

Channel Mode	Description
passive	LACP mode that places a port into a passive negotiating state in which the port responds to LACP packets that it receives but does not initiate LACP negotiation.
active	LACP mode that places a port into an active negotiating state in which the port initiates negotiations with other ports by sending LACP packets.
on	All static port channels (that are not running LACP) remain in this mode. If you attempt to change the channel mode to active or passive before enabling LACP, the device displays an error message. You enable LACP on each channel by configuring the interface in that channel for the channel mode as either active or passive. 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 default port-channel mode is on.

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.

Ports can form an LACP port channel when they are in different LACP modes if the modes are compatible as seen in these examples:

- A port in active mode can form a port channel successfully with another port that is in active mode.
- A port in active mode can form a port channel with another port in passive mode.

- A port in passive mode cannot form a port channel with another port that is also in passive mode, because neither port will initiate negotiation.
- A port in on mode is not running LACP and cannot form a port channel with another port that is in active or passive mode.

LACP ID 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.

The system ID is different for each VDC.

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.

Differences Between LACP-Enabled Port Channels and Static Port Channels

The table below summarizes the major differences between port channels with LACP enabled and static port channels.

Table 33: Differences Between LACP-Enabled Port Channels and Static Port Channels

Configuration	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	16	8 Starting from Cisco NX-OS Release 5.1, the maximum number of links supported in a channel is 16 on the Fseries modules.

LACP Compatibility Enhancements

Several new commands have been added in Release 4.2(3) to address interoperability issues and to assist with faster LACP protocol convergence.

When a Cisco Nexus 7000 Series device is connected to a non-Nexus peer, its 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. To address these conditions, the **lacp graceful-convergence** command was added.

By default, LACP sets a port to the suspended state if it does not receive an LACP PDU from the peer. In some cases, although this feature helps in preventing loops created due to misconfigurations, it can cause servers to fail to boot up because they require LACP to logically bring up the port. You can place a port in an individual state by using the **no lacp suspend-individual** command.

LACP Port-Channel Minimum Links and MaxBundle

A port channel aggregates similar ports to provide increased bandwidth in a single manageable interface.

With the Cisco NX-OS Release 5.1, 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 Offload to Fabric Extenders

To reduce the load on the control plane of the Cisco Nexus 7000 Series device, Cisco NX-OS provides the ability to offload link-level protocol processing to the Fabric Extender CPU. This feature is supported by LACP by default as soon as there is at least one LACP port channel configured on a Fabric Extender.

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 stateful switchover cannot be guaranteed with LACP fast timers.

Minimum Number of Links on the FEX Fabric Port Channel

In a network configuration of dual-homed hosts (active/standby), you can configure the Cisco Nexus 2000 Series Fabric Extender (FEX) to support a minimum number of links for fabric port channels.

When the number of fabric port-channel links falls below the specified threshold, the host-facing FEX interfaces are brought down, which allows for a NIC switchover on the connection between the host and the FEX. The automatic recovery of the FEX interfaces to the standby FEX is triggered when the number of fabric port-channel links reaches the specified threshold.

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 and you can reuse these port-channel numbers in different VDCs. For example, you can configure port channel 100 in VDC1 and also configure a different port channel 100 in VDC2.

However, the LACP system ID is different for each VDC. For more information about LACP, see the "LACP" section.

Note See the Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guidefor complete information about VDCs and assigning resources.

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. The port channels can originate in one VDC (with all ports in that channel in the same VDC) and partner with a port channel in another VDC (again, all ports in that channel must be in that VDC).

Note The port-channeling load-balancing mode works either for a single module or across the entire device. You must configure load balancing using port channels in the default VDC. You cannot configure load balancing using port channels within specified VDCs. 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 7000 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 in the channel group must be in the same VDC.
- 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 for Port Channels

Port channeling has the following configuration guidelines and limitations:

- 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 "Configuring Basic Interface Parameters," 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.
- You cannot configure the ports from an F1- and an M1-series module in the same port channel because the ports will fail to meet the compatibility requirements.
- You cannot configure the ports from an M1- and an M2-series module in the same port channel.
- You cannot configure the ports from an F2e- and an F3-series module in the same port channel because the ports will fail to meet the compatibility requirements.
- You cannot configure the ports from an F3- and M3-series module in the same port channel because the ports will fail to meet the compatibility requirements.
- You cannot configure the ports from an F4- and M3-series module in the same port channel because the ports will fail to meet the compatibility requirements.
- You cannot configure the ports from an F3- and F4-series module in the same port channel because the ports will fail to meet the compatibility requirements.
- You cannot configure the ports from an M2- and F3/M3/F4-Series Module in the same port channel because the ports will fail to meet the compatibility requirements.
- From Cisco NX-OS Release 5.1, you can bundle up to 16 active links into a port channel on the F1-series module.

- F1-series modules do not support load balancing of non-IP traffic based on a MAC address. If ports on an F1-series module are used in a port channel and non-IP traffic is sent over the port channel, Layer 2 traffic might get out of order.
- Only F series and the XL type of M-series modules support the RBH modulo mode.
- Random load balance on port channel is supported only on F3-series modules. Ensure both sides of the port channel are F3 modules only.

Default Settings

Table 34: Default Port-Channel Parameters

Parameter	Default
Port channel	Admin up
Load-balancing method for Layer 3 interfaces	Source and destination IP address
Load-balancing method for Layer 2 interfaces	Source and destination MAC address
Load balancing per module	Disabled
RBH modulo mode	Disabled
LACP	Disabled
Channel mode	on
LACP system priority	32768
LACP port priority	32768
Minimum links for LACP	1
Maxbundle	16
Minimum links for FEX fabric port channel	1
Random load balancing (port channels)	Disabled

Configuring Port Channels

Creating a Port Channel

You can create a port channel before you create a channel group. The software automatically creates the associated channel group.

Before you begin

- Enable LACP if you want LACP-based port channels.
- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	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	switch(config-if)# **show port-channel summary**	Displays information about the port channel.
Step 4	(Optional) switch(config-if)# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce an error during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 5	(Optional) switch(config-if)# **no shutdown**	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	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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*	Removes the port channel and deletes the associated channel group.

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.

Before you begin

- Enable LACP if you want LACP-based port channels.
- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- All Layer 2 member ports must run in full-duplex mode and at the same speed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	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	switch(config-if)# **switchport**	Configures the interface as a Layer 2 access port.
Step 4	(Optional) switch(config-if)# **switchport mode trunk**	Configures the interface as a Layer 2 trunk port.
Step 5	(Optional) switch(config-if)# **switchport trunk** {**allowed vlan** *vlan-id* | **native** *vlan-id*}	Configures necessary parameters for a Layer 2 trunk port.
Step 6	switch(config-if)# **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]	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.

	Command or Action	Purpose
		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	(Optional) switch(config-if)# **show interface** *type slot/port*	Displays interface information.
Step 8	(Optional) switch(config-if)# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce an error during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 9	(Optional) switch(config-if)# **no shutdown**	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 10	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Use the **no channel-group** command to remove the port from the channel group.

Table 35: Removing a Port From the Channel Group

Command	Purpose
no channel-group	Removes the port from the channel group.

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. You can also add subinterfaces to an existing Layer 3 port channel.

Before you begin

- Enable LACP if you want LACP-based port channels.
- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Remove any IP addresses configured on the Layer 3 interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	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	switch(config-if)# **no switchport**	Configures the interface as a Layer 2 access port.
Step 4	switch(config-if)# **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]	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. 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	(Optional) switch(config-if)# **show interface** *type slot/port*	Displays interface information.
Step 6	(Optional) switch(config-if) **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce an error during policy programming to ensure

	Command or Action	Purpose
		that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 7	(Optional) switch(config-if) **no shutdown**	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	(Optional) switch(config-if) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Use the **no channel-group** command to remove the port from the channel group.

Table 36: Removing a Port From the Channel Group

Command	Purpose
no channel-group	Removes the port from the channel group.

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)# no 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.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **bandwidth** *value*	Specifies the bandwidth, which is used for informational purposes. The range is from 1 to 80,000,000 kbs. The default value depends on the total active interfaces in the channel group.
Step 4	switch(config-if)# **delay** *value*	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. **Note** Prior to Cisco Release 4.2(1), the default delay value was 100 microseconds.
Step 5	switch(config-if)# **exit**	Exits the interface mode and returns to the configuration mode.
Step 6	(Optional) switch(config)# **show interface port-channel** *channel-number*	Displays interface information for the specified port channel.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	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.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **shutdown** | **no shutdown**	Shuts down the interface. No traffic passes and the interface displays as administratively down. The default is no shutdown. The **no shutdown** command opens the interface. The interface displays as administratively up. If there are no operational problems, traffic passes. The default is no shutdown.
Step 4	switch(config-if)# **exit**	Exits the interface mode and returns to the configuration mode.
Step 5	switch# **show interface port-channel** *channel-number*	Displays interface information for the specified port channel.
Step 6	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce an error during policy programming to ensure that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 7	(Optional) switch# **no shutdown**	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	(Optional) switch# **copy running-config startup-config**	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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	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	switch(config-if)# **description**	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	switch(config-if)# **exit**	Exits the interface mode and returns to the configuration mode.
Step 5	(Optional) switch(config-if)# **show interface port-channel** *channel-number*	Displays interface information for the specified port channel.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.

	Command or Action	Purpose
Step 3	switch(config-if)# **speed** {**10** | **100** | **1000** | **auto**}	Sets the speed for the port-channel interface. The default is auto for autonegotiation.
Step 4	switch(config-if)# **duplex** {**auto** | **full** | **half**}	Sets the duplex for the port-channel interface. The default is auto for autonegotiation.
Step 5	switch(config-if)# **exit**	Exits the interface mode and returns to the configuration mode.
Step 6	(Optional) switch# **show interface port-channel** *channel-number*	Displays interface information for the specified port channel.
Step 7	(Optional) switch# **copy running-config startup-config**	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 Flow Control

You can enable or disable the capability of the port-channel interfaces that run at 1 Gb or higher to send or receive flow-control pause packets. For port-channel interfaces that run at lower speeds, you can enable or disable only the capability of the port-channel interfaces to receive pause packets.

Note The settings have to match at both the local and remote ends of the link so that flow control can work properly.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **flowcontrol** {**receive** | **send**} {**desired** | **off** | **on**}	Sets the flow control parameters for sending and receiving the pause packets for the port-channel interface. The default is desired.

	Command or Action	Purpose
Step 4	switch(config-if)# **exit**	Exits the interface mode and returns to the configuration mode.
Step 5	(Optional) switch# **show interface port-channel** *channel-number*	Displays interface information for the specified port channel.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the port-channel interface for port channel group 2 to send and receive pause packets:

```
switch# configure terminal
switch(config)# interface port-channel 2
switch(config-if)# flowcontrol receive on
switch(config-if)# flowcontrol send on
```

Configuring Load Balancing Using Port Channels

You can configure the load-balancing algorithm for port channels that applies to the entire device or to only one module regardless of the VDC association. Module-based load balancing takes precedence over device-based load balancing.

Before you begin

- Enable LACP if you want LACP-based port channels.
- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **port-channel load-balance** *method* {**dst ip** | **dst ip-l4port-vlan** | **dst ip-vlan** | **dst mac** | **dst l4port** | **dst ip-l4port** | **src-dst ip** | **src-dst mac** | **src-dst l4port** | **src-dst ip-l4port** | **src-dst ip-vlan** | **src-dst ip-l4port-vlan** | **src ip** | **src**	Specifies the load-balancing algorithm for the device or module. The range depends on the device. The default for Layer 3 is **src-dst ip** for both IPv4 and IPv6, and the default for non-IP is **src-dst mac**.

	Command or Action	Purpose
	ip-l4port-vlan | **src ip-l4port** | **src ip-vlan** | **src mac** | **src l4port** | **hash-modulo** [**force**]} [**gtp-teid**] [**module** *module-number* | **fex** {*fex-range* | **all**}] [**asymmetric**] [**rotate** *rotate*]	**Note** The **asymmetric** keyword is valid with the **src-dst ip** command and F2 or F2e modules only. As F2 or F2e modules are symmetric by default, the **asymmetric** keyword prevents a traffic-drop occurring during bi-directional flow. A warning message prompts you that an F2 or F2e module needs to be enabled. This improves load-balancing and avoids any disruption to the system. Use the **no port-channel load-balance src-dst mac asymmetric** command to revert back to the default system settings (symmetrical). **Note** If a module-based configuration already exists, it takes precedence over the default system settings. Use the **no port-channel load-balance src-dst mac asymmetric module** command at module level to revert back to system level settings (symmetrical). **Note** The **module**, **asymmetric**, and **rotate** keywords are invalid with the **hash-modulo** command. When the **gtp-teid** keyword is specified in a packet that includes a GTP header field, the port-channel member selected depends not only on the already specified packet header fields such as MAC address, IP address, and L4 ports, but also on the 32-bit Tunnel Endpoint Identifier (TEID) header field. The packet must enter a port on an M3 module for the TEID header field to be used in the port-channel load-balancing. When the **gtp-teid** keyword is specified in a packet, the packet's TEID header field is used in port-channel member selection only if the packet contains an IPv4 or IPv6 header field followed by a UDP header field with the destination port 2152 and a GTP version 1 header field with the Protocol Type 1. All the other GTP header fields are considered GTP control messages. To avoid reordering of the GTP control messages in the network between GTP endpoints, NX-OS does not include the TEID header fields of the GTP control messages in its channel member selection.

	Command or Action	Purpose
		Note The **gtp-teid** keyword is supported only on M3 modules and does not affect the behavior of the other modules. Starting from Cisco NX-OS Release 8.3(1), the **gtp-teid** keyword is also supported on F4 modules.
Step 3	(Optional) **show port-channel load-balance**	Displays the port-channel load-balancing algorithm.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Use the **no port-channel load-balance** to restore the default load-balancing algorithm of src-dst mac for non-IP traffic and src-dst ip for IP traffic.

Command	Purpose
no port-channel load-balance	Restores the default load-balancing algorithm.

Example

This example shows how to configure source IP load balancing for port channels on module 5:

```
switch# configure terminal
switch(config)# port-channel load-balance src-ip-l4port module 5
```

This example shows how to configure different combinations for symmetric port channel load balancing for a port channel connected to switch1 and switch2. Use the same **rotate** *rotate-value* as listed in the following configuration combinations.

```
 ! Configure port-channel hash distribution at the global level!

switch1(config)# port-channel hash-distribution fixed
Switch2(config)# port-channel hash-distribution fixed


! Configure symmetric port-channel load balancing combinations on both
switch1 and switch2 of a port channel.!

!Combination 1!

switch1(config)# port-channel load-balance src ip
Switch2(config)# port-channel load-balance dst ip rotate 4

!Combination 2!

switch1(config)# port-channel load-balance dst ip
Switch2(config)# port-channel load-balance src ip rotate 4

!Combination 3!

switch1(config)# port-channel load-balance src ip-l4port
Switch2(config)# port-channel load-balance dst ip-l4port rotate 6
```

```
!Combination 4!

switch1(config)# port-channel load-balance dst ip-l4port
Switch2(config)# port-channel load-balance src ip-l4port rotate 6

!Combination 5!

switch1(config)# port-channel load-balance src ip-l4port vlan
Switch2(config)# port-channel load-balance dst ip-l4port rotate 8

!Combination 6!

switch1(config)# port-channel load-balance dst ip-l4port vlan
Switch2(config)# port-channel load-balance src ip-l4port rotate 8

!Combination 7!

switch1(config)# port-channel load-balance src ip-vlan
Switch2(config)# port-channel load-balance dst ip-vlan rotate 8

!Combination 8!

switch1(config)# port-channel load-balance dst ip-vlan
Switch2(config)# port-channel load-balance src ip-vlan rotate 8

!Combination 9!

switch1(config)# port-channel load-balance src l4port
Switch2(config)# port-channel load-balance dst l4port rotate 2

!Combination 10!

switch1(config)# port-channel load-balance dst l4port
Switch2(config)# port-channel load-balance src l4port rotate 2

!Combination 11!

switch1(config)# port-channel load-balance src mac
Switch2(config)# port-channel load-balance dst mac rotate 6

!Combination 12!

switch1(config)# port-channel load-balance dst mac
Switch2(config)# port-channel load-balance src mac rotate 6
```

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.

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature lacp**	Enables LACP on the device.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	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.

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.

	Command or Action	Purpose
Step 3	switch(config-if)# **channel-group** *number* **mode** {**active** | **on** | **passive**}	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	(Optional) switch(config-if)# **show port-channel summary**	Displays summary information about the port channels.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	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

From Cisco NX-OS Release 5.1, you can configure the LACP minimum links feature. Although minimum links and maxbundles work only in LACP, you can enter the commands for these features for non-LACP port channels, but these commands are nonoperational.

Before you begin

Ensure that you are in the correct port-channel interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **lacp min-links** *number*	Specifies the port-channel interface to configure the number of minimum links and enters the

	Command or Action	Purpose
		interface configuration mode. The range is from 1 to 16.
Step 4	(Optional) switch(config-if)# **show running-config interface port-channel** *number*	Displays the port-channel minimum links configuration.

Use the **no lacp min-links** command to restore the default port-channel minimum links configuration.

Table 37: Restoring the Default Port-Channel Minimum Links Configuration

Command	Purpose
no lacp min-links	Restores the default port-channel minimum links configuration.

Example

This example shows how to configure the minimum number of port-channel interfaces on module 3:

```
switch# configure terminal
switch(config)# lacp min-links 3
```

Configuring the LACP Port-Channel MaxBundle

From Cisco NX-OS Release 5.1, you can configure the LACP maxbundle feature. Although minimum links and maxbundles work only in LACP, you can enter the commands for these features for non-LACP port channels, but these commands are nonoperational.

Before you begin

Ensure that you are in the correct port-channel interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **lacp max-bundle** *number*	Specifies the port-channel interface to configure max-bundle, and enters the interface configuration mode.

	Command or Action	Purpose
		The default value for the port-channel max-bundle is 16. The allowed range is from 1 to 16.
Step 4	(Optional) switch(config-if)# **show running-config interface port-channel** *number*	Displays the port-channel minimum links configuration.

Use the **no lacp max-bundle** command to restore the default port-channel max-bundle configuration.

Table 38: Restoring the Default Port-Channel Max-Bundle Configuration

Command	Purpose
no lacp max-bundle	Restores the default port-channel max-bundle configuration.

Example

This example shows how to configure the port channel interface max-bundle on module 3:

```
switch# configure terminal
switch(config)# 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. In-service software upgrade (ISSU) and stateful switchover (SSO) are not supported with the LACP fast rate timer.

Note The number of interfaces validated with LACP Fast Timers in Cisco NX-OS Release 8.2(4) are:

- 250 physical member ports with port-channel in Layer 3 mode.
- 100 physical member ports with port-channel in Layer 2 mode with 1000 RSTP instances active on the system.

Before you begin

- Ensure that you have enabled the LACP feature.

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	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.

You can reuse the same configuration for the system priority values in more than one VDC.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **lacp system-priority** *priority*	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.

	Command or Action	Purpose
		Note Each VDC has a different LACP system ID because the software adds the MAC address to this configured value.
Step 3	(Optional) switch(config)# **show lacp system-identifier**	Displays the LACP system identifier.
Step 4	(Optional) switch(config)# **show running-config interface port-channel** *number*	Displays the port-channel minimum links 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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **lacp port-priority** *priority*	Configures the port priority for use with LACP. Valid values are from1 through 65535, and higher numbers have a lower priority. The default value is 32768.
Step 4	(Optional) switch(config-if)# **show running-config interface port-channel** *number*	Displays the port-channel minimum links 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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable LACP.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **shutdown**	Administratively shuts down the port channel.
Step 4	switch(config-if)# **no lacp graceful-convergence**	Disables LACP graceful convergence on the port channel.
Step 5	switch(config-if)# **no shutdown**	Brings the port channel administratively up.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	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
```

Re-Enabling LACP Graceful Convergence

If the default LACP graceful convergence is once again required, you can re-enable convergence.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **shutdown**	Administratively shuts down the port channel.
Step 4	switch(config-if)# **lacp graceful-convergence**	Enables LACP graceful convergence on the port channel.
Step 5	switch(config-if)# **no shutdown**	Brings the port channel administratively up.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	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 Port

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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Enable LACP.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **shutdown**	Administratively shuts down the port channel.
Step 4	switch(config-if)# **no lacp suspend-individual**	Disables LACP individual port suspension behavior on the port channel.
Step 5	switch(config-if)# **no shutdown**	Brings the port channel administratively up.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	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
```

Re-Enabling LACP Port

You can re-enable the default LACP individual port suspension.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **shutdown**	Administratively shuts down the port channel.
Step 4	switch(config-if)# **lacp suspend-individual**	Enables LACP individual port suspension behavior on the port channel.
Step 5	switch(config-if)# **no shutdown**	Brings the port channel administratively up.

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

Example

This example shows how to re-enable 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

From Cisco NX-OS Release 6.1(1), the adaptive and fixed hash distribution configuration is supported at 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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-channel hash-distribution** {**adaptive** | **fixed**}	Specifies the port-channel hash distribution at the global level. • **adaptive**—This is the default mode. RBH values are asymmetric. • **fixed**—Peer port connections must be in an ascending order. RBH values are distributed symmetrically as per the ascending order of the port. The number of buckets in each port is equal. While configuring this command, the following warning is displayed:

	Command or Action	Purpose
		This global 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	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure adaptive hash distribution at the global level:

```
configure terminal
 port-channel hash-distribution adaptive
show port-channel rbh-distribution
ChanId    Member port    RBH values         Num of buckets
--------  -------------  -----------------  ----------------
 3022      Eth15/5        0                  1
 3022      Eth15/21       4                  1
 3022      Eth15/6        1                  1
 3022      Eth15/13       2                  1
 3022      Eth15/14       3                  1
```

Configuring Port-Channel Hash Distribution at the Port-Channel Level

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** {*channel-number* | *range*}	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **[no] port-channel hash-distribution** {**adaptive** | **fixed**}	Specifies the port-channel hash distribution at the global level. • **adaptive**—This is the default mode. RBH values are asymmetric. • **fixed**—Peer port connections must be in an ascending order. RBH values are distributed symmetrically as per the ascending order of the port. The number of buckets in each port is equal. While configuring this command, the following warning is displayed: The command does not take effect until the next member link event (link down/up/no

	Command or Action	Purpose
		shutdown/shutdown). Do you still want to continue(y/n)? [yes]
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure fixed hash distribution at the port channel level:

```
configure terminal
 interface port-channel 3010
  port-channel hash-distribution fixed
show port-channel rbh-distribution interface port-channel 3021
ChanId    Member port    RBH values         Num of buckets
--------  -------------  -----------------  ----------------
 3021      Eth15/23       0                  1
 3021      Eth15/24       1                  1
 3021      Eth15/25       2                  1
 3021      Eth15/26       3                  1
```

Configuring RBH Modulo Mode

Enabling RBH modulo mode flaps all port channels.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-channel load-balance hash-modulo**	Enables the RBH modulo mode. This command reinitializes all port channels so there is an option to continue or not continue. **Note** This command is rejected if the current system-wide module types include the M1-Series module. To remove the M1-Series module type from the system-wide configuration, enter the **system module-type f1, f2, m1xl, m2xl** command.
Step 3	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable the RBH modulo mode:

```
switch# configure terminal
switch(config)# port-channel load-balance hash-modulo
```

Configuring Minimum Links on the FEX Fabric Port Channel

From Cisco NX-OS Release 6.1(3), you can configure a minimum number of links for the FEX fabric port channel so that when a certain number of FEX fabric port-channel member ports go down, the host-facing interfaces of the FEX are suspended.

Before you begin

Ensure that you are in the correct port-channel interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *number*	Specifies the interface to configure and enters the interface configuration mode.
Step 3	switch(config-if)# **switchport**	Configures the interface as a Layer 2 access port.
Step 4	switch(config-if)# **switchport mode fex-fabric**	Sets the port channel to support an external Fabric Extender.
Step 5	switch(config-if)# [**no**] **port-channel min-links** *number*	Configures the minimum number of links on the FEX fabric port channel. The range is from 1 to 16.
Step 6	(Optional) switch(config-if)# **show port-channel summary**	Displays summary information about the port channels.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the minimum number of links for the FEX fabric port channel:

```
switch# configure terminal
switch(config)# interface port-channel 100
switch(config-if)# switchport
switch(config-if)# switchport mode fex-fabric
```

```
switch(config-if)# port-channel min-links 3
switch(config-if)# show port-channel summary
Flags: D - Down P - Up in port-channel (members) I - Individual
H - Hot-standby (LACP only) s - Suspended r - Module-removed
S - Switched R - Routed U - Up (port-channel)
M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port- Type Protocol Member Ports Channel
--------------------------------------------------------------------------------
101 Po101(SM) Eth NONE Eth10/46(P) Eth10/47(P) Eth10/48(P)
```

Configuring Random Load Balance

Configuring Random Load Balance on a Port Channel

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure interface port-channel.

switch(config)# **interface port-channel** *port-channel-number*

Step 3 Configure random load balance for the port-channel interface. Use the **no** form of the following command to disable the random load balance feature.

switch(config-if)# **egress port-channel load-balance random**

Note This will override the default system or module-wide port-channel load balance settings. To configure random load balancing for ingress traffic, configure the **egress port-channel load-balance random** command on an switch virtual interface (SVI) on Layer 3.

Configuring Random Load Balance on an Interface

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure a port-channel interface:

switch(config)# **interface** *interface-name*

Step 3 Configure random load balance for the interface. Use the **no** form of the following command to disable the random load balance feature.

switch(config-if)# **egress port-channel load-balance random**

Note The ingress Layer 3 interface or a port-channel interface performs random load balance on the Layer 2 or Layer 3 egress interface and port-channel interface.

Configuring random load balance on a single physical interface is useful in scenarios where traffic comes in from an ingress Layer 3 interface and goes out of a port-channel interface.

Configuring Random Load Balance for a VLAN

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure a VLAN:

switch(config)# **vlan** *vlan-id*

Step 3 Enter VLAN configuration mode:

switch(config-vlan)# **vlan configuration** *vlan-id*

Step 4 Configure random load balance for the VLAN. Use the **no** form of the following command to disable the random load balance feature.

switch(config-if)# **egress port-channel load-balance random**

Note Random load balance is applied on all the Layer 2 ingress interfaces under the VLAN. The ingress interfaces perform random load balance on all the Layer 2 or Layer 3 port-channel egress interfaces.

Configuring Random Load Balance for an SVI

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure a switch virtual interface (SVI):

switch(config)# **vlan** *vlan-range*

Step 3 Enter VLAN configuration mode:

switch(config)# **vlan configuration** *vlan-range*

Step 4 Configure random load balance for the SVI for ingress traffic. Use the **no** form of the following command to disable the random load balance feature.

switch(config-vlan-config)# **egress port-channel load-balance random**

Example: Configuring Random Load Balance

This example shows how to configure random load balance on a port-channel interface:

```
configure terminal
    interface port-channel 44
       egress port-channel load-balance random
```

This example shows how to configure random load balance on a physical interface:

```
configure terminal
    interface Ethernet6/1
       egress port-channel load-balance random
```

This example shows how to configure random load balance on a VLAN:

```
configure terminal
    vlan 100
       vlan configuration 100
         egress port-channel load-balance random
```

This example shows how to configure random load balance on a switch virtual interface (SVI) for ingress traffic:

```
configure terminal
    vlan 2-10
    vlan configuration 2-10
       egress port-channel load-balance random
```

Verifying Port-Channel Configurations

Use the following commands to verify port-channel configurations:

Table 39: Verifying Port-Channel Configurations

Command	Purpose
show interface port-channel *channel-number*	Displays the status of a port-channel interface.
show feature	Displays enabled features.

Command	Purpose
load-interval {**interval** *seconds* {**1** | **2** | **3**}}	From Cisco NX-OS Release 4.2(1) for the Cisco Nexus 7000 Series devices, 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 rbh distribution	Displays the distribution of RBH values across port-channel interfaces.
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.
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.
show interface status error policy [**detail**]	Displays errors on interfaces and VLANs that are inconsistent with hardware policies. The **detail** command displays the details of the interfaces and VLANs that receive an error.

For more information about these commands, see the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference.

Monitoring the Port-Channel Interface Configuration

Use the following commands to display port-channel interface configurations:

Table 40: Monitoring the Port-Channel Interface Configuration

Command	Purpose
clear counters interface port-channel *channel-number*	Clears the counters.

Command	Purpose
clear lacp counters [**interface port-channel** *channel-number*]	Clears the LACP counters.
load-interval {**interval** *seconds* {**1** | **2** | **3**}}	From Cisco NX-OS Release 4.2(1) for the Cisco Nexus 7000 Series devices, 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.

See the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference for information about these commands.

Configuration Examples 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

Table 41: Related Documents

Related Topic
Cisco Nexus 7000 Series NX-OS Interfaces Command Reference
Cisco Nexus 7000 Series NX-OS System Management Configuration Guide
Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide
Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 7000 Series Switches, Release 6.x
Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide
Cisco NX-OS Licensing Guide
VLANs, MAC address tables, private VLANs, and the Spanning Tree Protocol. Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide
Cisco Nexus 7000 Series NX-OS FabricPath Command Reference
Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide
Cisco Nexus 7000 Series NX-OS Release Notes

Standards

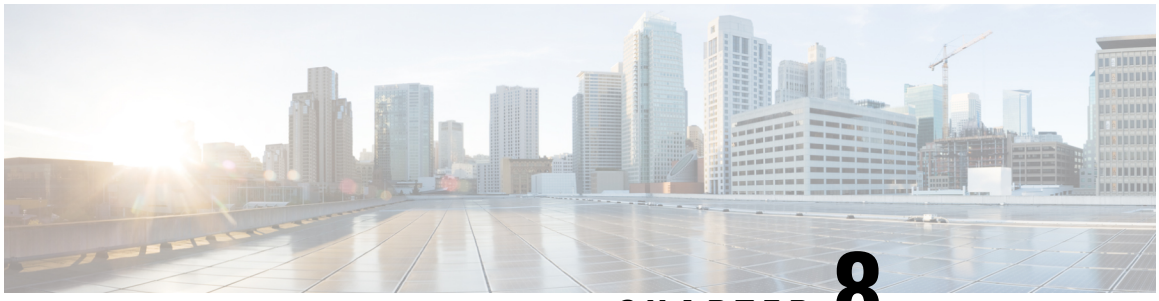
Table 42: Standards

Standards	Title
IEEE 802.3ad	—

MIBs

Table 43: MIBs

MIBs	MIBs Link
• IEEE8023-LAG-CAPABILITY • CISCO-LAG-MIB	To locate and download MIBs, go to: http://www.cisco.com/public/sw-center/netmgmt/cmtk/m



CHAPTER 8

Configuring vPCs

This chapter describes how to configure virtual port channels (vPCs) on Cisco NX-OS devices.

Note From Cisco NX-OS Release 5.1(1), vPCs have been enhanced to interoperate with FabricPath. To configure vPCs with FabricPath networks, see the Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide.

From Cisco NX-OS Release 5.1(1), you can use any of the 10-Gigabit Ethernet (10GE) interfaces, or higher, on the F-series modules or the 10-Gigabit Ethernet interfaces, or higher, on the M-series modules for the vPC peer link on an individual switch, but you cannot combine member ports on an F module with ports on an M module into a single port channel on a single switch. 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. See the Cisco NX-OS FCoE Configuration Guide for Cisco Nexus 7000 and Cisco MDS 9000 for more information about shared interfaces.

The port-channel compatibility parameters must also be the same for all vPC member ports on both peers and therefore you must use the same type of module in each chassis.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes

for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Feature History for Configuring vPCs

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 44: Feature History for Configuring vPCs

Feature Name	Release	Feature Information
Dynamic Routing over vPC	8.4(1)	Added support for Dynamic Routing over vPC feature on Cisco Nexus F4 Series modules for IPv4 and IPv6 unicast traffic.
vPC support on M3 modules	7.3(0)DX(1)	Added support for vPCs on M3 modules.
Hitless vPC Role Change	7.3(0)D1(1)	Added support for switching vPC roles without impacting traffic flows.
vPC Shutdown	7.2(0)D1(1)	Added the **shutdown** command that shuts down the peer to isolate it for debugging, reloading, or physically removing it from the vPC complex, and enables the peer vPC switch to take over as the primary peer.
Physical Port-based vPC on F3	7.2(0)D1(1)	Added support for Physical Port-based vPCs for F3.
1500 host vPC for FEX (Physical Port-based vPC on FEX)	7.2(0)D1(1)	Added support for 1500 host vPC for FEX (Physical Port-based vPC on FEX).
vPC Configuration Synchronization	7.2(0)D1(1)	vPC Configuration Synchronization feature synchronizes the configurations of one switch automatically to other similar switches.
Layer 3 over vPC for F2E and F3 modules	7.2(0)D1(1)	Added support for this feature.
Physical Port-based vPC on F2	6.2(6)	Added support for Physical Port-based vPCs for F2.
LAN shutdown	6.2(6)	Added the **shutdown lan** command to support this feature.
FCoE over Physical Port-based vPCs	6.2(6)	Added support for this feature.
Physical Port-based vPCs	6.2(6)	Added support for Physical Port-based vPCs on the physical interface of vPC peer devices.

Feature Name	Release	Feature Information
vPCs	6.2(2)	Added the **mode auto** command to enable certain commands for vPCs simultaneously.
vPCs	6.1(3)	Added the **multicast load-balance** command that allows two peers to be partially designated forwarders when both vPC paths are up.
vPCs	5.2(1)	Support increased to 528 vPCs.
vPCs	5.2(1)	Added the **vpc orphan-ports suspend** command to suspend orphan ports on the vPC secondary device when the vPC fails.
vPCs	5.2(1)	Added the **auto-recovery** command to improve speed and reliability of vPC recovery after an outage. The **reload restore** command is deprecated.
vPCs	5.2(1)	Added per-VLAN consistency checking so that only those VLANs with inconsistent configuration are suspended.
vPCs	5.2(1)	Added the **graceful consistency-check** command to enable the vPC primary device to forward traffic when inconsistent configuration is detected between the peers.
vPCs	5.0(2)	Added the **peer-switch** command to enable the vPC switch pair to appear as a single STP root in the Layer 2 topology.
vPCs	5.0(2)	Added the **reload restore** command to configure the vPC switch to assume its peer is not functional and to bring up the vPC.
vPCs	4.2(1)	Added the **delay restore** command to delay the bringup of the vPC secondary device after reload until the routing table can converge.
vPCs	4.2(1)	Added the **dual-active exclude interface-vlan** command to ensure that VLAN interfaces remain up if the vPC peer link fails.
vPCs	4.2(1)	Added the **peer-gateway** command to ensure that all packets use the gateway MAC address of the device.
vPCs	4.2(1)	Support increased to 256 vPCs.
vPCs	4.1(4)	Support increased to 192 vPCs.
vPCs	4.1(2)	These features were introduced.

Information About vPCs

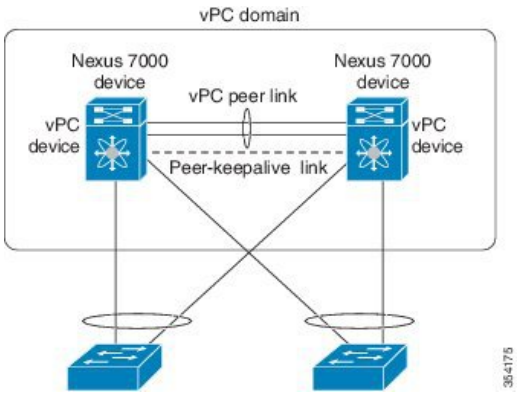
A virtual port channel (vPC) allows links that are physically connected to two Cisco Nexus 7000 Series devices to appear as a single port channel by a third device. You can create a vPC by placing a physical port into a port-channel and assign the port-channel as vPC. For F2, F3, and FEX modules, a streamlined vPC configuration is introduced allowing to assign the physical ports as vPC members without requirement to associate those to a local port-channel.

vPC+

A virtual port channel+ (vPC+) is an extension to virtual port channels (vPCs) that run CE only. A vPC+ domain allows a classical Ethernet (CE) vPC domain and a Cisco FabricPath cloud to interoperate and also provides a First Hop Routing Protocol (FHRP) active-active capability at the FabricPath to Layer 3 boundary. A vPC+ domain enables Cisco Nexus 7000 Series enabled with FabricPath devices to form a single vPC+, which is a unique virtual switch to the rest of the FabricPath network. For more detailed information on vPC+ see the Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide.

Note You cannot configure a vPC+ domain and a vPC domain in the same VDC.

Figure 11: vPC Architecture



You can use only Layer 2 port channels in the vPC. A vPC domain is associated to a single Virtual Device Context (VDC), so all vPC interfaces belonging to a given vPC domain must be defined in the same VDC. You must have a separate vPC peer link and peer-keepalive link infrastructure for each VDC deployed. Consolidating a vPC pair (two vPC peer devices of the same domain) in two VDCs of the same physical device is not supported. The vPC peer link must use at least 10-Gigabit Ethernet ports for both ends of the link or the link will not form.

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, the F-series line cards can have 16 active links and M-series line cards can have 8 active links in a single port

channel. When you configure the port channels in a vPC—including the vPC peer link channels—using LACP, F-series card each device can have eight 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.)

You can use the **lacp graceful-convergence** command to configure port channel Link Aggregation Control Protocol (LACP) graceful convergence. You can use this command only on a port-channel interface that is in an administratively down state. You cannot configure (or disable) LACP graceful convergence on a port channel that is in an administratively up state.

You can use the **lacp suspend-individual** command to enable LACP port suspension on a port channel. LACP sets a port to the suspended state if it does not receive an LACP bridge protocol data unit (BPDU) from the peer ports in a port channel. This can cause some servers to fail to boot up as they require LACP to logically bring up the port.

Note You must enable the vPC feature before you can configure or run the vPC functionality.

From Cisco NX-OS Release 4.2, the system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. See the Cisco Nexus 7000 Series NX-OS System Management Configuration Guide for information about rollbacks and checkpoints.

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 7000 Series chassis by using two or more 10-Gigabit Ethernet ports in dedicated port mode. To ensure that you have the correct hardware to enable and run a vPC from Cisco NX-OS Release 4.1(5), 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 7000 Series chassis, you configure another port channel again using two or more 10-Gigabit Ethernet ports 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. If you are not using the correct module, the system displays an error message.

Note 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.

From Cisco NX-OS Release 4.2, 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. 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 can create a track object and apply that object to all links on the primary vPC peer device that connect to the core and to the vPC peer link. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for information about the track interface command.

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 Always attach all vPC devices using port channels to both vPC peer devices.

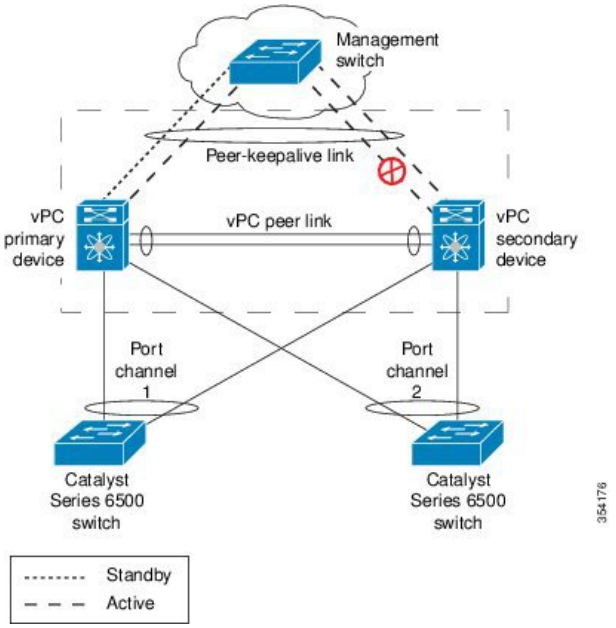
vPC Terminology

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 states between the vPC peer devices. Both ends must be on 10-Gigabit Ethernet interfaces.
- 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 7000 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 the figure below).

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 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 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 Links

A vPC peer link is the link that is used to synchronize the states between the vPC peer devices. Both ends of the link must be on 10-Gigabit Ethernet interfaces.

- Keeps both vPC peer switches synchronized for control plane information (such as the vPC state, consistency parameters, and MAC addresses).
- Forwards data packets to the vPC peer switch, when the local vPC is down.
- A single vPC domain between two VDCs on the same physical Cisco Nexus 7000 device is not supported.

Note You must configure the peer-keepalive link before you configure the vPC peer link or the peer link does not come up. (See the "Peer-Keepalive Link and Messages" section for information about the vPC peer-keepalive link and messages.)

Note Starting from Cisco NX-OS Release 8.0(1) you cannot configure vPC peer-link on a port-channel with non-default MTU configuration. The following error message is displayed if you try to configure:

```
ERROR: Cannot configure peer-link since mtu is non-default
```

To configure peer-link, remove the non-default MTU configuration and re apply the **vpc peer-link** command. By default packets of all sizes are allowed in peer-link.

You can configure a vPC peer link to configure two devices as vPCs peers. You must use the module in order to configure a vPC peer link.

We recommend that you use the dedicated port mode when you configure a vPC peer link. For information about the dedicated port mode, see "Configuring Basic Interface Parameters."

vPC Peer Link and I/O Modules Support in Cisco NX-OS Release 6.2

You can configure F2e VDCs. The VDC type for two vPC peer devices must match when the F2 Series module and the F2e Series module are used in the same VDC or system. For an F2 Series module and an F2e Series module in the same topology, the features related to the F2 Series module will only apply.

After ISSU to Cisco NX-OS Release 6.2(2), F2 VDCs will automatically change to F2 F2e VDCs, regardless of the existence of an F2e Series module.

The table below displays the I/O modules that are supported on both sides of a vPC peer link in Cisco NX-OS Release 6.2.

Table 45: I/O Module Combinations Supported on Both Sides of a vPC Peer Link, Cisco NX-OS Release 6.2 and Later

vPC Primary	vPC Secondary
M1 I/O module	M1 I/O module
M2 I/O module	M2 I/O module
M3 I/O module	M3 I/O module
F2 I/O module	F2 I/O module
F2 I/O module	F2e I/O module
F2e I/O module	F2e I/O module
F2e I/O module	F2 I/O module
F3 I/O module	F3 I/O module

vPC Peer Link and I/O Modules Support in Cisco NX-OS Release 6.1 and Earlier Releases

In Cisco NX-OS Release 6.1 and earlier releases, only identical I/O modules on either side of a vPC peer link are supported. Using different I/O modules on either side of a vPC peer link is not supported. Mixing I/O modules on the same side of a port channel is also not supported. The table above displays the I/O modules that are supported on both sides of a vPC peer link.

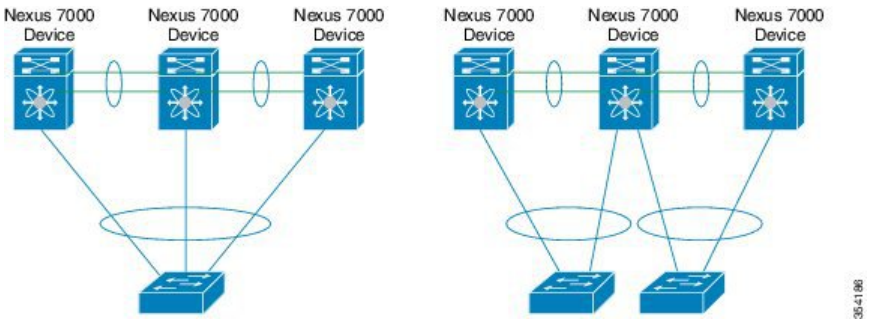
While using port channels, we recommended that you use identical line cards on both sides.

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.

The figure below 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 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 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.

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.

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

Configuration information flows across the vPC peer links using the Cisco Fabric Services over Ethernet (CFSoE) protocol. (See the "Cisco Fabric Services Over Ethernet" section on page 7-30 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 "Cisco Fabric Services Over Ethernet" section on page 7-30 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.

We recommend that you create and configure a separate VRF and configure a Layer 3 port on each vPC peer device in that VRF for the vPC peer-keepalive link. The default ports and VRF for the peer-keepalive are the management ports and VRF.

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 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.
  - When the port-channel is designated as the vPC peer link, the spanning-tree port type network command is added and so the port-channel becomes the bridge assurance port.
  - 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:

- If LACP is used as port-channel aggregation protocol, UDLD is not required in a vPC domain.
- If LACP is not used as the port-channel aggregation protocol (static port-channel), use UDLD in normal mode on vPC member ports.
- If STP is used without Bridge Assurance and if LACP is not used, use UDLD in normal mode on vPC orphan ports.

See the "Configuring UDLD Mode" section for information about configuring UDLD.

Configuring Layer 3 Backup Routes on a vPC Peer Link

You can use VLAN network interfaces on the vPC peer devices for such applications as HSRP and PIM. You can use a VLAN network interface for routing from the vPC peer devices.

Note 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 "Configuring Layer 3 Interfaces."

From Cisco NX-OS Release 6.2(2), if the vPC peer link is on an F2e-Series module in a mixed chassis with an M-Series module and an F2e-Series module, do not use the Layer 3 backup routing path over the vPC peer link; instead deploy a dedicated Layer 3 backup routing path using an additional inter-switch port channel.

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.

From Cisco NX-OS Release 4.2(1), you can ensure that specified VLAN interfaces do not go down on the vPC secondary device when the vPC peer link fails.

Use the **dual-active exclude interface-vlan** command to configure this feature.

Note From Cisco NX-OS Release 7.2(0)D1(1), when you attach a Layer 3 device to a vPC domain, the peering of routing protocols using a VLAN also carried on the vPC peer link is not supported. If routing protocol adjacencies are needed between vPC peer devices and a generic Layer 3 device, you must use physical routed interfaces for the interconnection. Use of the vPC peer-gateway feature does not change this requirement.

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.

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 peer link itself to send and receive vPC peer-keepalive messages. For more information about configuring VRFs, see the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide.

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. You can configure a hold-timeout and a timeout value simultaneously.

Hold-timeout value—The hold-timeout value range is between 3 to 10 seconds, with a default value of 3 seconds. This timer starts when the vPC peer link goes down. The purpose of the hold-timeout period is to prevent false-positive cases.

If you configure a hold-timeout value that is lower than the timeout value, then the vPC system ignores vPC peer-keepalive messages for the hold-timeout period and considers messages for the reminder of the timeout period. If no keepalive message is received for this period, the vPC secondary device takes over the role of the primary device. For example, if the hold-timeout value is 3 seconds and the timeout value is 5 seconds, for the first 3 seconds vPC keepalive messages are ignored (such as, when accommodating a supervisor failure for a few seconds after peer link failure) and keepalive messages are considered for the remaining timeout period of 2 seconds. After this period, the vPC secondary device takes over as the primary device, in case there is no keep alive message.

Timeout value—The timeout value range is between 3 to 20 seconds, with a default value of 5 seconds. This timer starts at the end of the hold-timeout interval. If you configure a timeout value that is lower than or equal to the hold-timeout value, then the timeout duration is initiated after the hold-timeout period. For example, if the timeout value is 3 seconds and the hold-timeout value is 5 seconds, the timeout period starts after 5 seconds

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.

Use the 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.

This is an example of configuring an interface as a trusted port:

```
(config)# class-map type qos match-all trust-map
(config-cmap-qos)# match cos 4-7

(config)# policy-map type qos ingresspolicy
(config-pmap-qos)# class trust-map

(config)# interface Ethernet8/11
(config-if)# service-policy type qos input ingresspolicy
```

See the Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide for complete information about configuring trusted ports and precedence.

vPC Peer Gateway

From Cisco NX-OS Release 4.2(1), 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 From Cisco NX-OS Release 6.2(2), you can use the mode auto command to automatically enable this feature. See the "Enabling Certain vPC Commands Automatically" section for more information about using this command.

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 7000 Series and Cisco Nexus 7700 Series devices rather than the common HSRP gateway. This behavior is noncompliant with some basic Ethernet RFC standards. Packets that reach a vPC device for the nonlocal router MAC address are sent across the 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 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.

Note From Cisco NX-OS Release 5.1(3) and above, when a VLAN interface is used for Layer 3 backup routing on the vPC peer devices and an F1 line card is used as the peer link, the VLAN must be excluded from the peer-gateway feature, if enabled, by running the peer-gateway exclude-vlan vlan-number command. For more information about backup routes, see the "Configuring Layer 3 Backup Routes on a vPC Peer Link" section.

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.

Dynamic Routing over vPC

Dynamic Routing over vPC feature is supported on F2E, F3, and M3 series modules (for IPv4 and IPv6 Unicast traffic). From Cisco NX-OS Release 8.4(1), the dynamic routing over vPC feature is supported on F4 Series modules.

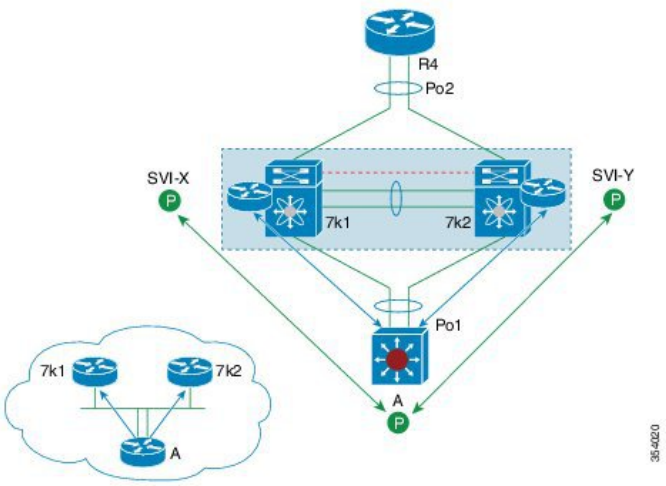
This feature enables L3 routing protocols such as OPSF to form adjacency with the two vPC peer chassis. The equal routing cost matrices must be configured on applicable interface on each of the vPC peers, failure to do so can result in blocking the traffic. Asymmetric routing feature has to be implemented to address this issue and to configure Dynamic Routing over vPC. Additionally, when Dynamic Routing over vPC is enabled a warning log message is printed.

Layer 3 over vPC for F2E, F3 Modules

This section describes the Layer 3 over vPC for F2E, F3 and M3 Modules feature and how to configure it. Starting from Cisco NX-OS Release 7.2(0)D1(1), Layer 3 over vPC is available on F2E and F3 Series modules. Using this feature, a Layer 3 device can form peering adjacency between both the vPC peers in a vPC complex. vPC peers must have identical VLANs. The TTL of the traffic sent over a peer link does not decrement. The peer-gateway feature should be enabled on all I/O modules before configuring the Layer 3 over vPC feature. The peer-gateway feature allows the vPC peer (SVI-X) (refer the figure below) to forward packets on behalf of other peer (SVI-Y). This feature saves bandwidth by avoiding traffic over the peer link. You can set up peer adjacency between Layer 3 device and vPC peer without separate Layer 3 links. Both bridged and routed traffic can flow over the same link.

Routing adjacency between Layer 3 device and vPC peer is formed without a non-vPC VLAN. Adjacency is formed on the vPC VLAN. Routing adjacency between a Layer 3 device and a vPC peer is formed without Layer 3 inter-switch links between the vPC peers. Adjacency is formed on the vPC peer-link. There is faster convergence when a link or device fails for all traffic. vPC loop avoidance mechanism is available for all traffic.

Figure 14: Layer 3 Over vPC Solution



Layer 3 over VPC Support in Cisco NX-OS Release 7.2(0)D1(1)

The following are figures illustrates the Layer 3 over VPC Support in Cisco NX-OS Release 7.2(0)D1(1):

Figure 15: Supported: Peering Over a vPC Interconnection Where the Router Peers with Both the vPC Peers.

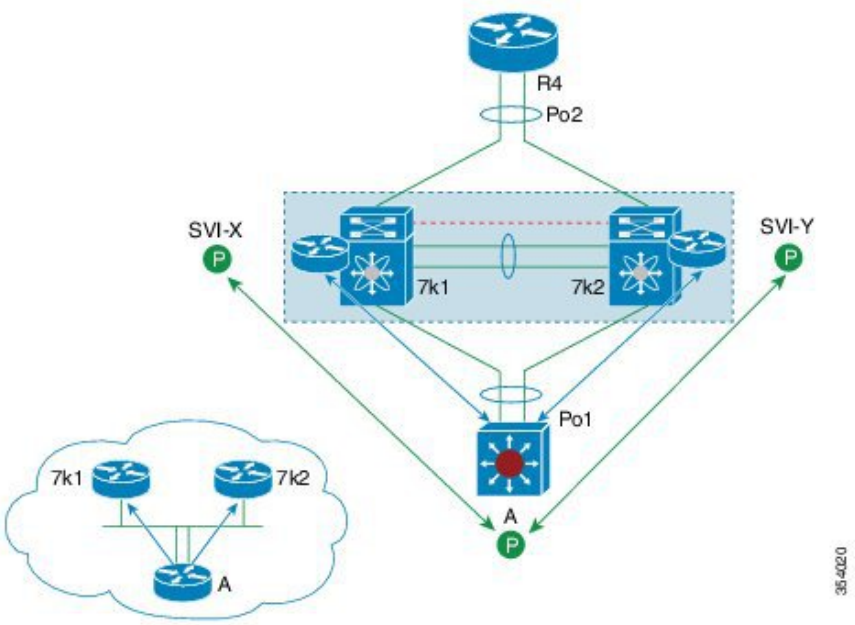


Figure 16: Supported: Peering Over an STP Interconnection Using a vPC VLAN Where the Router Peers with Both the vPC Peers.

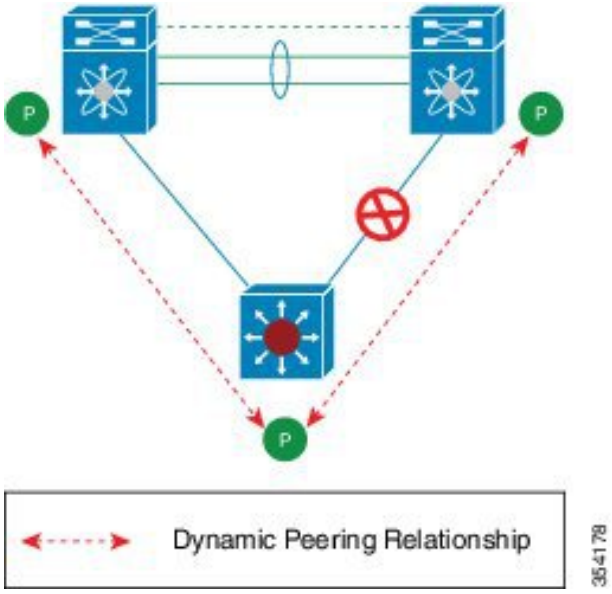


Figure 17: Supported: Peering Over an Orphan Device with Both the vPC Peers.

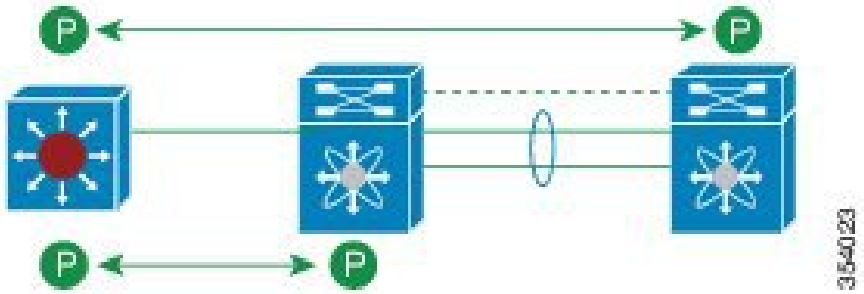


Figure 18: Supported: Peering Over a vPC Interconnection Where Each Nexus Device Peers with Two vPC Peers.

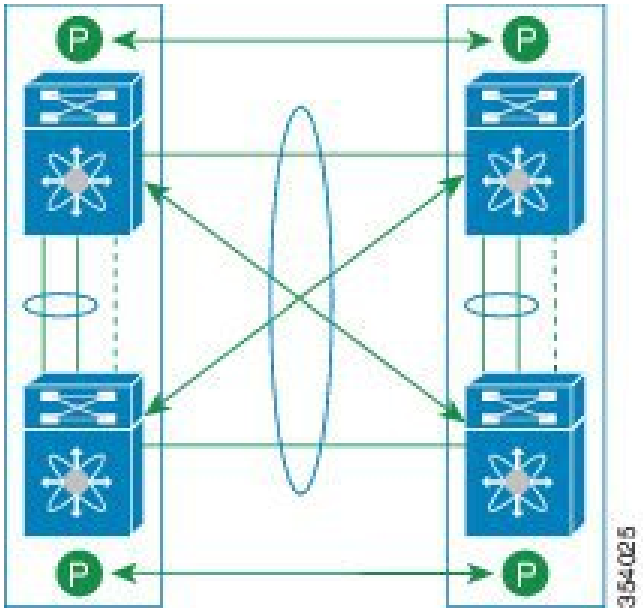
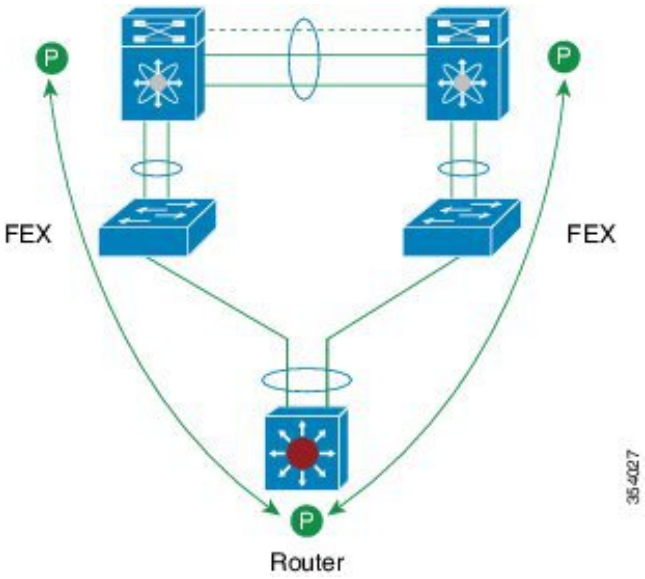


Figure 19: Supported: Peering with vPC Peers Over FEX vPC Host Interfaces



The FEX is connected to Nexus in straight-through topology. The router peers with both Nexus boxes over satellite ports. Layer 3 over vPC in FEX Active-Active mode vPC is not supported.

Figure 20: Unsupported: Peering Across vPC Interfaces with Unequal Layer 3 Metrics

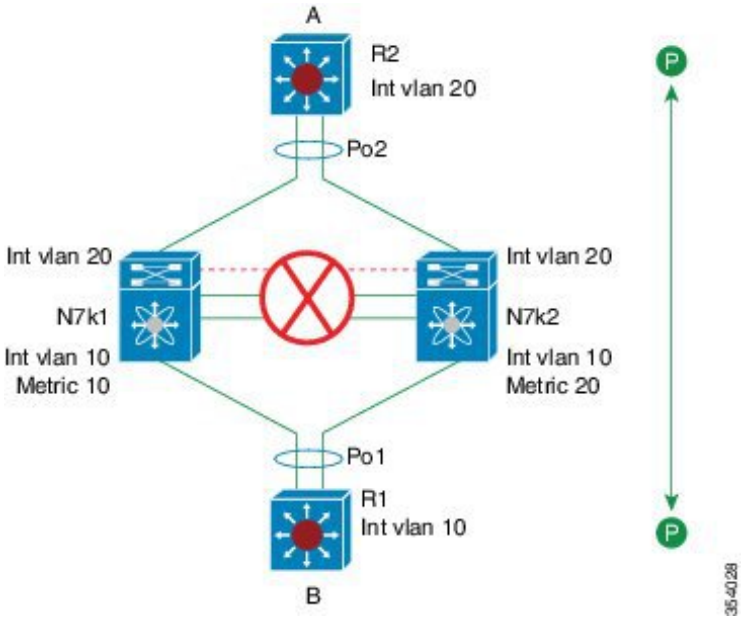
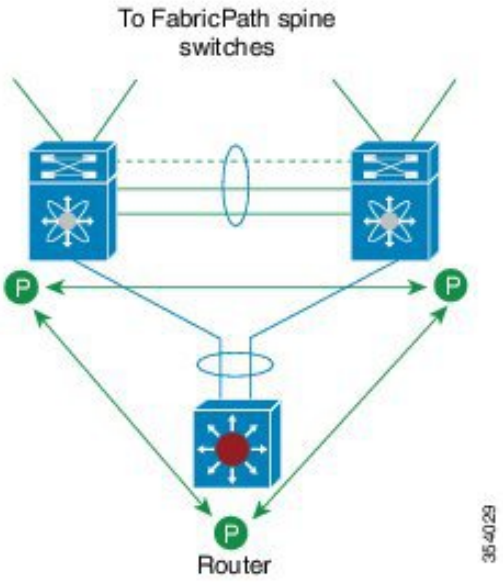


Figure 21: Unsupported: Peering Over vPC+ Interfaces in Cisco NX-OS 7.2(0)D1(1)



Peering with vPC peers over vPC+ interfaces is unsupported.

Figure 22: Unsupported: Peering with vPC+ Peers an STP Interconnection Using a vPC+ VLAN

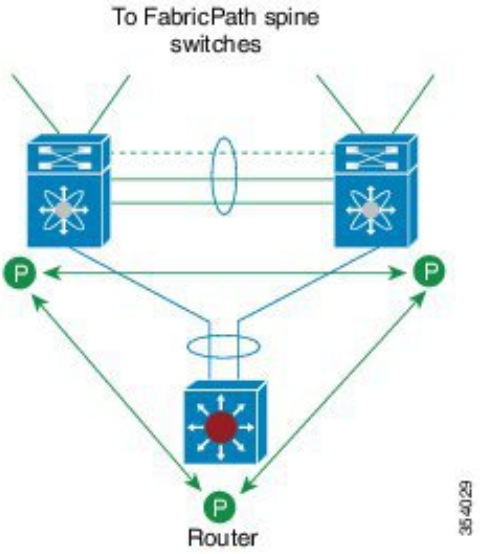


Figure 23: Unsupported: Route Peering with Orphan Device with Both the vPC+ Peers

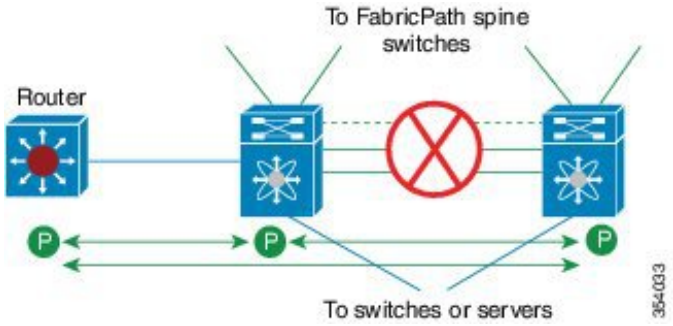
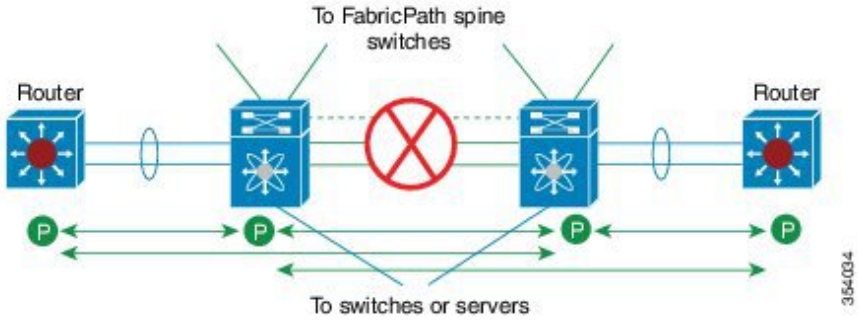


Figure 24: Unsupported: Peering Over PC Interconnection and Over vPC+ Peer Link Using vPC VLAN



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 VDC.

You must explicitly configure the port channel that you want to act as the peer link on each device. You associate the port channel that you made a 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. All ports in the vPC on each of the vPC peer devices must be in the same VDC. 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.

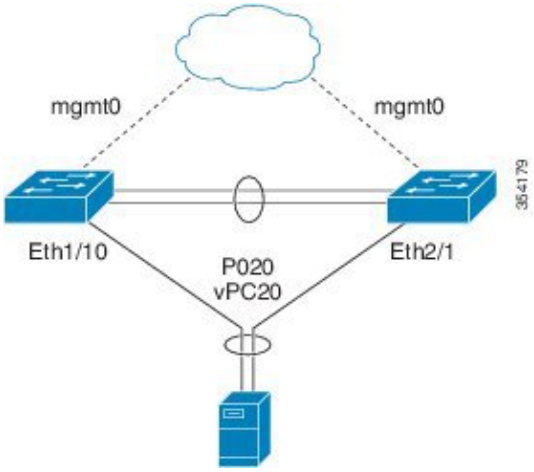
See the "Cisco Fabric Services Over Ethernet" section for more information about displaying the vPC MAC table. After you create a vPC domain, the Cisco NX-OS software creates a system priority for the vPC domain. You can also configure a specific system priority for the vPC domain.

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 figure below shows a basic configuration in which the Cisco Nexus 7000 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.

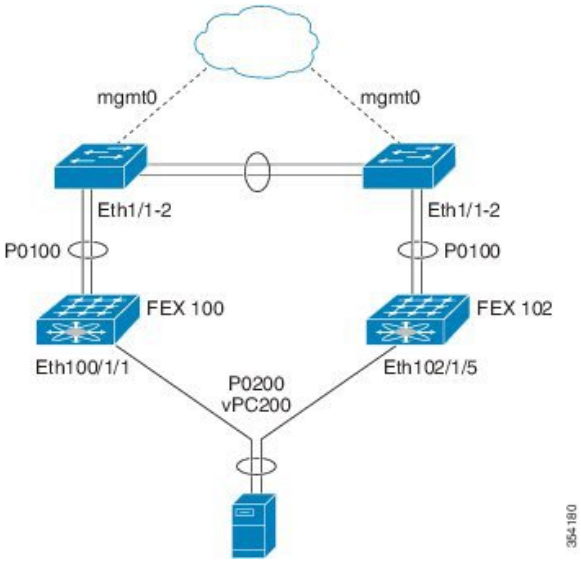
Figure 25: Switch vPC Topology



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.

From Cisco NX-OS Release 5.2(1), you can configure a vPC from the peer devices through Fabric Extenders (FEXs), as shown in the figure below.

Figure 26: FEX Straight-Through Topology (Host vPC)



In the figure, each FEX is single-homed (straight-through FEX topology) with a Cisco Nexus 7000 Series device. The host interfaces on this FEX are configured as port channels and those port channels are configured as vPCs. Eth100/1/1 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 Cisco Nexus 2000 Series Fabric Extender Software Configuration Guide for Cisco Nexus 7000 Series Switches, Release 7.x for more information about configuring FEX ports.

Physical Port-based vPCs

Physical port-based vPCs are vPCs directly configured on the physical interface of a vPC device. Physical port-based vPCs can optionally run Link Aggregation Control Protocol (LACP) to the downstream device. Physical port-based vPCs are supported on F2 and F2E modules. The vPC configuration is applied directly on the physical port member. You can also enable LACP protocol on the physical interface configured with vPC. From Cisco NX-OS Release 7.2(0)D1(1), Physical port-based vPCs are supported on F3 and FEX modules as well.

Physical Port as vPCs members for F2, F3, and FEX

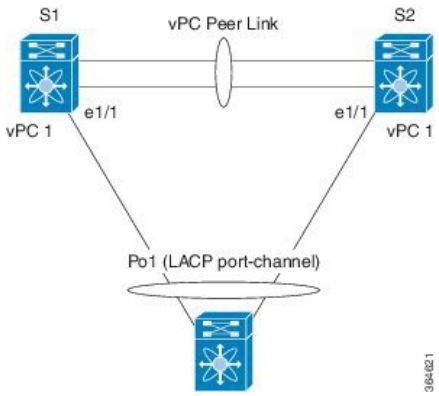
This section describes streamlined configuration of vPC which directly assigns physical port as vPC members instead of bundling into a local port-channel, which is configured as vPC. This capability specific to vPC with member port connected to F2, F3, or FEX modules.

- Enables simple configuration as the user does not create a port-channel to enable the vPC configuration. The vPC configuration is applied directly on the physical port member.
- Supports vPC setup that has only one 10 Gigabit Ethernet, 40 Gigabit Ethernet, or 100 Gigabit Ethernet port in each leg of the vPC. Creation of port-channel for a vPC setup in such case is not optimal. This feature is best suited for port-channel vPC with only one interface.

- Enhances scalability enabling future support for more physical ports.
- Provides accounting logs and system logs for the physical port, rather than the port-channel.
- Supports large FEX setups. This feature is best suited for port-channel vPC with only one interface.
- Expands the limits of vPC by decoupling the configuration and deployment from the port-channel constructs.
- Enables additional enhancement to extend FCOE support on physical port on the vPC, thus enabling multipathing for the Ethernet traffic while preserving existing constructs for FCOE support.

Note The **fabricpath multicast load-balance** command must be enabled before configuring Physical Port vPC+. This requirement applies to regular front panel and FEX ports.

Figure 27: Physical Port vPC Topology



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 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 "Cisco Fabric Services Over Ethernet" 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.

The compatibility check process for vPCs differs from the compatibility check for 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)

The following parameters were added in Cisco NX-OS Release 6.2(6) for Physical Port-based vPCs:

- Native VLAN
- Port mode
- Interface type
- VLAN xLT mapping
- vPC card type
- Shared mode

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
- 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)
- 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)
- Gateway Load-Balancing Protocol (GLBP)
- 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

In releases earlier than Cisco NX-OS Release 5.2(1), when a consistency check detects a mismatch in a parameter from the list of parameters that must be identical, the vPC peer link and vPC are prevented from coming up. If a parameter mismatch is configured after the vPC is already established, the vPC moves into suspend mode and no traffic flows on the vPC.

From Cisco NX-OS Release 5.2(1), 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.

Use the **graceful consistency-check** command to configure this feature.

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. In releases earlier than Cisco NX-OS Release 5.2(1), if the configuration of any enabled VLAN is inconsistent across the peer devices, the vPC is prevented from establishing or moves into a suspended mode.

From Cisco NX-OS Release 5.2(1), 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. See the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide for more information about STP port types.

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.

vPC Shutdown

The vPC Shutdown feature enables a user to isolate a switch from a vPC complex before it is debugged, reloaded, or even removed physically, so that the vPC traffic passing through the peer vPC switch in the vPC complex is not affected.

When the user executes the **shutdown** command, the MCEC module (MCECM) stops sending out-of-band (OOB) keep-alive messages and also brings down all the vPC ports, SVIs, and the peer-link. On detection of the peer-link going down and the non-availability of the keep-alive messages, the peer vPC switch takes over as the primary peer. As the keep-alive messages are not received, the peer vPC switch does not bring up the vPC peer-link even after a flap. The isolated vPC switch keeps all the vPCs down as the peer-link is down. The vPC orphan port suspends configured orphan ports.

When the user executes the **no** form of this command, the switch is brought back into the vPC complex with minimal disruption of the network traffic. Executing the **no** form of this command, starts the keepalives, brings up the peer links, and consecutively brings up all the vPCs.

When executed on the primary switch, the **shutdown** command dual-active status is established.

Orphan ports lose connectivity when the vPC **shutdown** command is executed.

Cisco NX-OS services saves the **shutdown** command in the persistent storage service (PSS). The command is restored when the switch reloads. The **shutdown** command is saved as vPC configuration. The **shutdown** command executed again along with the vPC configuration, if it has been copied to the startup configuration. The **shutdown** command is restored when the switch reloads

Version Compatibility Among vPC Switches After vPC shutdown Command

It is possible that the vPC operating version of an isolated vPC peer switch that comes up after debugging or after an ISSU, is different from that the peer switch. When the **no shutdown** command is applied, the vPC peer-link comes up with both the switches having as their versions the lower of the two versions.

Role of STP in vPC Shutdown

The STP synchronizes the port states to the vPC peer causing the new primary vPC peer to take over from the current state, when the role switchover happens. If the MCECM take more than 6 seconds to detect the role change and notify the STP, then the STP bridge protocol data units (BPDUs) that are sent on the vPC are timed out. To avoid this, it is recommended to configure STP peer switch feature so that both vPC switches send BPDUs over the vPC ports.

vPC shutdown Command for a Switch in FEX Active-Active Mode

If you configure the **shutdown** command on the switch to which a dual-homed FEX is connected in a vPC, the FEX goes offline on that switch. An ISSU of the isolated switch does not update the software image on the FEX. You cannot use the vPC **shutdown** command to perform ISSU by isolating and upgrading each switch for FEX Active-Active.

Consider the following FEX Active-Active scenario where peers Peer 1 and Peer 2 are involved:

- The inactive peer, that is Peer 2, is offline because of reasons such as the VPC shutdown command
- An ISSU has been performed on the active peer, that is Peer 1, for upgrading from one software image version to a higher version

All line cards and the remote line cards, including FEX Active-Active, upgrade to higher version of the software image. This happens because the FEX Active-Active is offline on the inactive peer.

Consecutively, when the inactive peer becomes online due to the VPC no shutdown command, this peer will still run the lower version of the software image. In such as case, the status of FEX Active-Active toggles between AA version mismatch and Offline in this peer. This is because both the peers run different versions of the software image. To avoid this situation, the user should not bring up the Peer 2, or execute the VPC shutdown command on it, until the Peer 2 is also upgraded to higher version software image.

Role of the Layer 2 MCECM in vPC Shutdown

When you execute the **shutdown** command, the Multichassis EtherChannel Module (MCECM) stops the keep-alive messages and brings down the peer-link. If the vPC peer switch does not receive keep-alive messages in 5 seconds, it assumes the primary role.

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.

Configuring vPC Peer Links and Links to the Core on a Single Module

Note 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.

From Cisco NX-OS Release 4.2, 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 7000 Series NX-OS Unicast Routing Configuration Guide for information about configuring object tracking and track lists.

See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for information about configuring object tracking.

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. Note that the Boolean AND operation is not supported with vPC object tracking.

A vPC deployment with a single Cisco Nexus 7000 Series M132XP-12 module or M108XP-12 module, where the L3 core uplinks and vPC peer-link interfaces are localized on the same module, is vulnerable to access layer isolation if the 10-Gbps module fails on the primary vPC (vPC member ports are defined on both 1-Gbps line cards and on 10-Gbps line card).

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.

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.

With M Series modules and LACP, a vPC peer link supports 16 LACP interfaces: 8 active links and 8 hot standby links. You can configure 16 LACP links on the downstream vPC channel: 8 active links and 8 hot standby links. If you configure the port channels without using LACP, you can have only 8 links in each channel. With F-Series line cards, a vPC peer link and downstream vPC channels support up to 16 active LACP links. You can have 16 links in each channel even if the port channels are not configured using LACP.

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.

When the port-channel is designated as the vPC peer link, the spanning-tree port type network command is added and so the port-channel becomes the bridge assurance port. We 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 7000 Series NX-OS Layer 2 Switching Configuration Guide for information about STP enhancement features and PVST simulation.

Note

- Bridge Assurance is enabled automatically on vPC peer-link during creation of link. It is recommended to retain Bridge assurance on the peer-link.
- Bridge Assurance on VPC is not supported.

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 "Cisco Fabric Services Over Ethernet" 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 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 7000 Series NX-OS Layer 2 Switching Configuration Guide for an example.

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 7000 Series NX-OS Layer 2 Switching Configuration Guide for more information about STP port types.

Note If you bridge two VLANs on a Nexus 7000 peer-switch, with an Adaptive Security Appliance (ASA) in a transparent mode, the switch puts one of the VLAN in a STP dispute. To avoid this, disable peer-switch or STP on the ports.

vPC Peer Switch

The vPC peer switch feature is enabled on Cisco NX-OS Release 5.0(2) to address performance concerns around STP convergence. This feature allows a pair of Cisco Nexus 7000 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 the 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 vPC topology (non-hybrid), in which all the devices belong to the vPC topology.

Note The Peer-switch feature on networks that use vPC and STP-based redundancy is not supported. If the vPC peer-link fails 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 traffic going to the first vPC peer and the other half traffic to the second vPC peer. With peer link failure, there is no impact to the north/south traffic but the east/west traffic is lost.

See the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide for information about STP enhancement features and Rapid PVST+.

vPC Peer Link's Designated Forwarder

From Cisco NX-OS Release 6.0, Cisco NX-OS provides a way to control two peers to be partially designated forwarders when both vPC paths are up. When this control is enabled, each peer can be the designated forwarder for multi-destination southbound packets for a disjoint set of RBHs/FTAGs (depending on the hardware). The designated forwarder is negotiated on a per-vPC basis. This control is enabled with the **fabricpath multicast load-balance** command which is configured under vPC domain mode, for example:

```
switch(config)# vpc domain 1
switch(config-vpc-domain)# fabricpath multicast load-balance
```

From Cisco NX-OS Release 6.2(2), this feature is automatically enabled when the **mode auto** command is used. See the "Enabling Certain vPC Commands Automatically" section for more information about using this command.

Note Only an F2-series module supports multicast load balancing. On an F1-series module, the configuration is supported, but load balancing does not occur.

Note The **fabricpath multicast load-balance** command is required for configuring vPC+ with FEX ports.

See the Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide for more detailed information on enabling designated forwarders on vPCs.

vPC and ARP or ND

A feature was added in the Cisco NX-OS Release 4.2(6) 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 peer link port channel flaps or when a vPC peer comes back online.

Note From Cisco NX-OS Release 6.2(2), you can use the mode auto command to automatically enable this feature. See the "Enabling Certain vPC Commands Automatically" section for information about using this command.

vPC Multicast—PIM, IGMP, and IGMP Snooping

Note The Cisco NX-OS software for the Nexus 7000 Series devices does not support Product Independent Multicast (PIM), Source-Specific Multicast(SSM) or Bidirectional (BIDR) on a vPC. The Cisco NX-OS software fully supports PIM Any Source Multicast (ASM) on a vPC.

A PIM adjacency between an 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.

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.

See the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference for information about commands that display information on a vPC and multicast.

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 neighbor relationship between a vPC VLAN (a VLAN that is carried on a vPC peer link) and a downstream vPC-attached Layer 3 device is not supported, which can result in dropped multicast packets. If a PIM neighbor relationship is required with a downstream Layer 3 device, a physical Layer 3 interface must be used instead of a vPC interface.

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 7000 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 (in non-F2 mode), 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 inoperational 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.

PIM DUAL DR and IP PIM PRE-BUILD SPT with VPC Peer Link on F2 Modules

In the vPC implementation in F2-mode, because of a hardware limitation, the PIM dual DR mode is disabled. As a result, only the PIM DR adds the OIF, and the states are shown in this example:

```
Case 1: One OIF
===============
VPC Device1 (say this is PIM DR on oif1):
-----------------------------------------
(*,G)
  oif1 (igmp)

VPC Device2:
------------
(*,G) will not be created.
```

When the source traffic is received, only vPC Device 1 adds the (S,G) route.

```
VPC Device1 (say this is PIM DR on oif1):
-----------------------------------------
(*,G)
  oif1 (igmp)
(S,G)
  oif1 (mrib)

VPC Device2:
------------
(*, G) will not be created.
(S, G) will not be created.
```

In this case (with F2 mode), even if you enter the **ip pim pre-build-spt** command, no value is added because the corresponding (S,G) route is not created in the first place.

```
Case 2: Two OIFs
================
VPC Device1 (say this is PIM DR on oif1):
-----------------------------------------
(*,G)
  oif1 (igmp)

VPC Device2 (say this is PIM DR on oif2):
-----------------------------------------
(*,G)
  oif2 (igmp)
```

When the source traffic is received, associated OIFs are inherited by the (S,G) routes as shown in this example:

```
VPC Device1 (say this is PIM DR on oif1):
-----------------------------------------
(*,G)
  oif1 (igmp)

(S,G)
  oif1 (mrib)

VPC Device1 (say this is PIM DR on oif2):
-----------------------------------------
(*,G)
  oif2 (igmp)
```

```
(S,G)
  oif2 (mrib)
```

In the case of a vPC peer link with F2 modules, you do not need to enter the **ip pim pre-build-spt** command because PIM sends (S,G) joins upstream because associated routes have a non-NULL oiflist.

Note Do not enter the **ip pim pre-build-spt** command if the vPC feature is enabled in F2 mode.

vPC Peer Links and Routing

The First Hop Routing Protocols (FHRPs) interoperate with vPCs. The Hot Standby Routing Protocol (HSRP), Gateway Load Balancing Protocol (GLBP), 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. For GLBP, the forwarders on both vPC peer devices forward traffic.

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.

From Cisco NX-OS Release 4.2(1), 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 to the **delay restore** command.

See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for more information about FHRPs and routing.

Cisco Fabric Services Over Ethernet

The Cisco Fabric Services over Ethernet (FSoE) is a reliable state transport mechanism that is used to synchronize the actions of the vPC peer devices. Cisco FSoE carries messages and packets for many features linked with vPC, such as STP and IGMP. Information is carried in Cisco Fabric Service or Cisco FSoE protocol data units (PDUs).

When you enable the vPC feature, the device automatically enables Cisco FSoE, and you do not have to configure anything. Cisco FSoE distributions for vPCs do not need the capabilities to distribute over IP or the FS regions. You do not need to configure anything for the Cisco FSoE feature to work correctly on vPCs.

The Cisco FSoE transport is local to each VDC.

You can use the **show mac address-table** command to display the MAC addresses that Cisco FSoE synchronizes for the vPC peer link.

Note Do not enter the **no cfs eth distribute** or the **no cfs distribute** command. You must enable Cisco FSoE 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 Cisco FSoE.

Cisco Fabric Service also transports data over TCP/IP. See the Cisco Nexus 7000 Series NX-OS System Management Configuration Guide for more information about Cisco Fabric Service over IP.

Note The software does not support Cisco Fabric Service 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 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 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. From Cisco NX-OS Release 5.2(1), 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.

Fibre Channel over Ethernet over Physical Port-based vPCs

The Fibre Channel over Ethernet (FCoE) over Physical Port Virtual Port Channels (vPCs) feature extends the shared model for physical Ethernet interfaces to vPC interfaces.

Each Ethernet interface that forms a vPC leg is shared between the storage virtual device context (VDC) and the Ethernet VDC. The shared Ethernet interface carries both FCoE and LAN traffic. Mutually exclusive FCoE and LAN VLANs are allocated to carry the traffic on the vPC leg; FCoE traffic is carried by the FCoE VLAN and LAN traffic is carried by the LAN VLAN.

Shutdown LAN

Certain configuration and network parameters must be consistent across peer switches in order for physical port vDCs to work. If an inconsistency impacting the network (Type 1) is detected, the secondary vPC leg (the physical link between the access switch and the host) is brought down. With FCoE over physical port vPC, vPC legs carry both FCoE and LAN traffic so that the FCoE and LAN link are both brought down. The shutdown LAN feature enables you to shut down or bring up only the LAN VLANs on an Ethernet interface.

vPC Recovery After an Outage

In a data center outage, both of the Cisco Nexus 7000 Series devices that include a vPC get reloaded. Occasionally only one peer can be restored. With no functioning peer-keepalive or peer link, the vPC cannot function normally, but depending on your Cisco NX-OS release, a method might be available to allow vPC services to use only the local ports of the functional peer.

Restore on Reload

Note From Cisco NX-OS Release 5.2(1), the **reload restore** command and method is deprecated. We recommend that you use the **auto-recovery** command and method.

From Cisco NX-OS Release 5.0(2), you can configure the Cisco Nexus 7000 Series device to restore vPC services when its peer fails to come online by using the reload restore command. You must save this setting in the startup configuration. On reload, the Cisco NX-OS software starts a user-configurable timer (the default is 240 seconds). If the peer link port comes up physically or if the peer-keepalive is functional, the timer is stopped and the device waits for the peer adjacency to form.

If at timer expiration no peer-keepalive or peer link up packets were received, the Cisco NX-OS software 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 STP primary regardless of its role priority and also acts as the master for LACP port roles.

Autorecovery

From Cisco NX-OS Release 5.2(1), you can configure the Cisco Nexus 7000 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 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 reinitialize 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 master for LACP port roles.

From Cisco NX-OS Release 6.2(2), you can use the **mode auto** command to automatically enable this feature. See the "Enabling Certain vPC Commands Automatically" section for information about using this command.

From Cisco NX-OS Release 7.2(0)D1(1), the secondary device assumes primary role, if the primary peer is down and 15 keep-alives messages are lost.

From Cisco NX-OS Release 7.2(0)D1(1), to enable the secondary peer to take over as the primary peer if the secondary peer misses 15 keep-alives from primary peer, you can configure **auto-recovery** command. When the switch reloads, the auto-recovery timer starts, and the switch takes on the primary STP role if the peer switch does not respond to it.

When vPC shutdown command is configured, auto-recovery is blocked.

From Cisco NX-OS Release 6.2.(2), for auto recovery to occur during the initial boot, the logical peer link must be down, and no peer keepalive messages must be received. In earlier releases, auto recovery did not occur if peer kepalive messages were not received and the physical peer link was set to Up status.

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.

See the Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide for complete information about high-availability features.

Hitless vPC Role Change

The vPC hitless role change feature provides a framework to switch vPC roles between vPC peers without impacting traffic flows. The vPC role swapping 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 when the **vpc role preempt** command is executed.

Use Case Scenario for Hitless 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 configuring the **vpc role preempt** command. Configure **no port-channel limit** under the vpc domain command before configuring the **vpc role preempt** command.

- 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 the **vpc role preempt** command to restore the device roles to be primary and secondary.

vPC Configuration Synchronization

Virtual port channels (vPC) topologies require identical configurations on peer switches. As a result, you must repeat configurations on both peer switches. This process, which can cause errors due to misconfigurations or omissions, can result in additional service disruptions because of mismatched configurations. Configuration synchronization eliminates these problems by allowing you to configure one switch and automatically synchronize the configuration on the peer switch.

In a vPC topology, each Cisco Nexus 7000 Series switch must have some matching parameters. You can use a vPC consistency check to verify that both Cisco Nexus 7000 Series switches have the same configuration (Type 1 or Type 2). If they do not match, depending on whether it is a global (for example, spanning-tree port mode), a port-level (for example, speed, duplex, or channel-group type), or even a port-channel interface, the vPC can go into a suspended state or a VLAN can go into a blocking state on both peer switches. As a result, you must ensure that the configuration from one switch is copied identically to the peer switch.

Configuration synchronization allows you to synchronize the configuration between a pair of switches in a network. Configuration synchronization and vPCs are two independent features and configuration synchronization does not eliminate vPC consistency checks. The checks will continue. If there is a configuration mismatch, the vPC can still go into a suspended state.

In a FEX Active-Active setup:

- All the Host Interfaces (HIFs) ports are mapped to the internal vPC.
- The vPC Config-Sync feature listens to the internal vPC creation notification and triggers a merge of the HIF port configuration.
- All the future HIF configuration are synchronized with the peer switch, if the merge is successful.
- The status of HIF is marked as "peer out of synchronization" and the configuration of the interface is not synchronized, if the merge fails.
- We recommend that you disable **vpc-config-sync** command before starting ASCII configuration. After the ASCII configuration is completed, enable **config-sync** command for regular operation.

Note

- vPC peer-link should be configured and up state.
- You cannot chose which commands are synchronized.

Benefits of vPC Configuration Synchronization

Configuration synchronization benefits are as follows:

- Provides a mechanism to synchronize configuration from one switch to another switch.
- Merges configurations when connectivity is established between peers.
- Provides mutual exclusion for commands.
- Supports existing session and port profile functionality.
- Provides minimal user intervention.
- Minimizes the possibility of user error.

Supported Commands for vPC Configuration Synchronization

The following types of commands are enabled for configuration synchronization:

Note The **show vpc config-sync cli syntax** command lists all the commands that are enabled for configuration synchronization. You cannot choose which commands are synchronized. For more information, see the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference.

- Type-1 configurations:
  - Global configurations
  - vPC member port-channel configurations
- vPC configurations.

Note The configurations can be given on either of the vPC peer switches.

Guidelines and Limitations for vPCs

vPCs have the following configuration guidelines and limitations:

- Enable vPCs before you configure them.
- Configure the peer-keepalive link and messages before the system can form the vPC peer link.
- Routing over vPC is supported only on F2E and F3 modules prior to Cisco NX-OS Release 8.1(1). Starting from Cisco NX-OS Release 8.1(1), routing over vPC is also supported on M3 series modules for IPv4 unicast traffic. Starting from Cisco NX-OS Release 8.2(1), routing over vPC is also supported on M3 series modules for IPv6 unicast traffic. Routing over vPC is supported on F4 series modules from Cisco NX-OS Release 8.4(1).

- Configure a separate Layer 3 link for routing from the vPC peer devices, rather than using a VLAN network interface for this purpose.
- All ports for a given vPC must be in the same VDC.
- Physical port vPC is not supported with VDCs containing F4 modules. If you have a mixed VDC with F3 and F4 modules, physical port vPC is not supported even when the FEXs are connected to F3 modules.
- Assign a unique vPC domain ID for each respective vPC to configure multilayer (back-to-back) vPCs.
- DHCP Relay is supported.
- We recommend that you configure all the port channels in the vPC using LACP with the interfaces in active mode.
- When a pair of Cisco Nexus 7000 series switches is connected to a downstream device in a vPC setup, and the vPC domain Id is changed, the LACP port channel configuration on one of the switches might go in hot stand-by mode. To avoid the above scenario, we recommend that you remove the vPC configurations and reconfigure the vPC configurations.
- Configure both vPC peer devices; the configuration is not sent from one device to the other.
- Only Layer 2 port channels can be in vPCs.
- vPC peers can operate dissimilar versions of NX-OS software only during the upgrade or downgrade process.
- Different versions of NX-OS software on vPC peer switches is not supported.
- IPv6 multicast on a vPC is not supported.
- Back-to-back, multilayer vPC topologies require unique domain IDs on each respective vPC.
- When using vPCs, we recommend that you use default timers for FHRP (HSRP, VRRP, GLBP), and PIM configurations. There is no advantage in convergence times when using aggressive timers in vPC configurations.
- Configure **vpc orphan-ports suspend** command on all non-vPC-interfaces (port channel or ethernet) that carry vPC peer-link VLAN traffic. During vPC shutdown, vPC manager brings down vPC interfaces, vPC interface VLANs and non-vPC interfaces with **vpc orphan-ports suspend** configuration.
- 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 7000 Series NX-OS Unicast Routing Configuration Guide for further details about OSPF.
- When you configure a static MAC address on a vPC switch, ensure to configure a corresponding static MAC address on the other vPC switch. If you configure the static MAC address only on one of the vPC switches, the other vPC switch will not learn the MAC address dynamically.
- In a vPC topology, when a Multichassis EtherChannel Trunk (MCT) link is shut down on a vPC primary switch, and is followed by the vPC primary switch reload, the vPC secondary switch's ports do not come up immediately. This may cause a drop in traffic.

- 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.
- From Cisco NX-OS Release 7.2(0)D1(1), when you attach a Layer 3 device to a vPC domain, the peering of routing protocols using a VLAN also carried on the vPC peer link is not supported. If routing protocol adjacencies are needed between the vPC peer devices and a generic Layer 3 device, you must use physical routed interfaces for the interconnection. Use of the vPC peer-gateway feature does not change this requirement.
- From Cisco NX-OS Release 8.1(x), in a vPC topology, non-MAC-in-MAC-encapsulated traffic can be lost if all the following conditions are met:
  - The non-MAC-in-MAC-encapsulated traffic that is routed through FabricPath enabled VLANs.
  - The packets have to hit the vPC switch from a non-core interface (an orphan port or from one of the hosts hanging off the vPC leg).
  - The packet must be destined to one of the hosts hanging off the vPC leg. It has to be an Layer 3 routing case.
  - The **no port-channel limit** command is configured under vPC.
  - The vPC leg connecting to the vPC host is down and the traffic is routed through the vPC peer link.
  - The vPC peer link is on M3 line card modules.

  In such a scenario, we recommend that you do not configure the **no port-channel limit** command under vPC.
- The STP port cost is fixed to 200 in a vPC environment.
- You might experience minimal traffic disruption while configuring vPCs.
- Jumbo frames are enabled by default on the vPC peer link.
- Routing protocol adjacency over a fabric path VLAN is not supported.
- The software does not support BIDR PIM or SSM on vPCs.
- The software does not support DHCP snooping, DAI, or IPSG in a vPC environment.
- The software does not support CFS regions.
- Port security is not supported on port channels.
- BFD for HSRP is not supported in a vPC environment.
- A single vPC domain between two VDCs on the same physical Cisco Nexus 7000 device is not supported.
- When Layer 3 over vPC feature is enabled using the **layer3 peer-router** command, BFD enabled with echo function is not supported on a switched virtual interface (SVIs) using vPC VLANs that are part of a vPC peer-link.

Auto recovery has the following limitations and guidelines:

- In Cisco NX-OS Release 6.2(2) and later releases, auto recovery is enabled by default. If you already enabled auto recovery in an earlier release and you upgrade to Release 6.2(2) or a later release, auto

recovery will remain enabled after the upgrade. If you want to disable auto recovery in Release 6.2(2) or a later release, you must use the **auto-recovery disable** command to explicitly disable auto recovery.

- From Cisco NX-OS Release 6.2.(2), for auto recovery to occur during the initial boot, the logical peer link must be down and no peer keepalive messages must be received. In releases earlier than 6.2.2, if peer keepalive messages were not received and the physical peer link was set to UP status, auto recovery did not occur.

Physical Port-based vPCs have the following guidelines and limitations:

- Physical Port-based vPCs are supported only on Nexus F2, F2e, and F3 Series modules.
- Physical port vPC is not supported with VDCs containing M3 modules.
- Physical port vPC is supported with vPC+ only on Nexus F2, F2e, and F3 Series modules.
- Physical port vPC is supported on a Fabric Extender (FEX) interface.
- Physical port vPC peer-link must be configured on Cisco Nexus F2, F2E, or F3 Series modules. It cannot be configured on a M Series module.
- Link Aggregation Control Protocol (LACP) cannot be enabled on a physical port without vPC.
- Same vPC configuration cannot be applied to multiple physical ports.
- Physical port vPC does not support ASCII-replay. When ASCII-replay occurs during a non-ISSU upgrade or downgrade between incompatible images, the Physical Port-based vPCs on the peer that is not undergoing upgrade will also go down temporarily.
- STP port-type network is not supported for vPC port-channels and STP port-type network is not supported, when **vpc role preempt** is configured on vPC port-channels.

FCoE over physical port vPC has the following guidelines and limitations:

- FCoE is supported only on trunk ports.
- FCoE is supported only for shared interfaces.
- FCoE is not supported on port channel vPCs.
- FCoE over a physical port vPC is supported in storage VDCs of type F2 only.
- FCoE over a physical port vPC is not supported in storage VDCs because Layer 2 multipathing over Physical Port-based vPCs are supported only for LAN.
- FCoE over a VPC+ is not supported.
- The shutdown LAN configuration is supported on shared interfaces only.
- The Link Layer Discovery Protocol (LLDP) must be enabled in the Ethernet VDC for shutdown LAN.

Hitless vPC role change feature has the following guidelines and limitations:

- vPC STP hitless role change feature is supported only from Cisco Nexus 7.3(0)D1(1) release onwards.
- vPC role change can be performed from either of the peer devices.
- 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.

- On vPC+, enable the **fabricpath multi path load-balance** command before configuring the vPC hitless role change feature. The Forwarding Tag (FTag) scheme is used in vPC+ to seamlessly configure the role change. To ensure FTag scheme is used, you need to enable the **no port channel limit** command on vPC+ as it has dependencies on the **fabricpath multi path load-balance** command.
- Enable the **no port channel limit** command on vPC+ before configuring the vPC hitless role change feature. If this command is not enabled, vPC hitless role change cannot be configured and an error message is displayed. Configure this command on both the vPC devices.

Note 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.
- vPC hitless role change cannot be performed if there are any Type 1 inconsistencies on the peer devices.
- When the peer-switch feature is enabled under a vPC domain, ensure that the vPC pair is configured as spanning-tree root for all the vPC VLANs.

Configuring vPCs

Enabling vPCs

Before you begin

- You must enable the vPC functionality before you can configure and use vPCs.
- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature vpc**	Enables vPCs on the device.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show feature**	Displays which features are enabled on the device.

	Command or Action	Purpose
Step 5	(Optional) switch# **copy running-config startup-config**	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
```

Disabling vPCs

Note When you disable the vPC functionality, the device clears all the vPC configurations.

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature vpc**	Disables vPCs on the device.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show feature**	Displays which features are enabled on the device.
Step 5	(Optional) switch# **copy running-config startup-config**	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
```

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 VDC. 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.

Before you begin

- Ensure that you are in the correct VDC (if you are not in the correct VDC, use the **switchto vdc** command).
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 4	(Optional) switch# **show vpc brief**	Displays brief information about each vPC domain.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create a vPC domain:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# exit
```

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
```

Configuring a vPC Keepalive Link and Messages

Note You must configure the vPC peer-keepalive link before the system can form the vPC peer link.

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 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 peer link itself to send vPC peer-keepalive messages. For information about creating and configuring VRFs, see the Cisco Nexus 7000 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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **peer-keepalive destination** *ip address* [**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* | **udp-port** *number* | **vrf** {*name* | **management** | **vpc-keepalive**}]	Configures the IPv4 address 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. Ensure that you either use IPv4 address to configure the peer-keepalive link. The management ports and VRF are the defaults.

	Command or Action	Purpose
		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 7000 Series NX-OS Unicast Routing Configuration Guide.
Step 4	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 5	(Optional) switch# **show vpc statistics**	Displays information about the configuration for the keepalive messages.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

For more information about configuring VRFs, see the Cisco Nexus 7000 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
```

Creating a vPC Peer Link

You create the vPC peer link by designating the port channel that you want on each device as the 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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.
- Ensure that you are using a Layer 2 port channel.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Selects the port channel that you want to use as the vPC peer link for this device, and enters interface configuration mode.
Step 3	(Optional) switch(config-if)# **switchport mode trunk**	Configures this interface in trunk mode.
Step 4	(Optional) switch(config-if)# **switchport trunk allowed vlan** *vlan-list*	Configures the permitted VLAN list.
Step 5	switch(config-if)# **vpc peer-link**	Configures the selected port channel as the vPC peer link, and enters vpc-domain configuration mode. **Note** When the port-channel is designated as the vPC peer link, the **spanning-tree port type network** command is added, so the port-channel becomes the bridge assurance port.
Step 6	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 7	(Optional) switch# **show vpc brief**	Displays information about each vPC, including information about the vPC peer link.
Step 8	(Optional) switch# **copy running-config startup-config**	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
```

Configuring Physical Port vPC on F2, F3, and FEX

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *name number*	Specifies the interface that you want to add to a physical port, and enters the interface configuration mode.
Step 3	switch(config-if)# **switchport**	Configures the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **vpc** *number*	Configures the selected physical interface into the vPC to connect to the downstream device, and enters interface vPC configuration mode. You can use any module in the device for the physical interface. The range is from 1 and 4096. **Note** The vPC number that you assign to the physical interface connecting to the downstream device from the vPC peer device must be identical on both vPC peer devices.
Step 5	Required: switch(config-if-vpc)# **lacp mode active**	Enables LACP on the physical port. **Note** Static mode can also be used.
Step 6	Required: switch(config-if-vpc)# **exit**	Exits the interface vPC configuration mode.
Step 7	Required: switch(config-if)# **exit**	Exits the interface configuration mode.
Step 8	Required: switch(config)# **exit**	Exits the global configuration mode.
Step 9	(Optional) switch# **show running-config interface** *name number*	Displays information about the interface.

Example

This example shows how to configure Physical Port vPC on F2, F3, and FEX modules:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport
switch(config-if)# vpc 10
switch(config-if-vpc)# lacp mode active
switch(config-if-vpc)# exit
switch(config-if)# exit
```

```
switch(config)# exit
switch# show running-config interface
```

This example shows how to verify the LACP mode:

```
switch# show running-config interface

   Interface Ethernet1/1
   no shutdown
   Switchport
    vpc 1
      lacp mode active
```

Creating VLAN on vPC

vPC VLAN is a VLAN that is allowed on vPC member port and vPC peer-link. When configuring large number of VLANs in a vPC environment, it is recommended to configure the VLANs simultaneously by specifying the range of VLANs, instead of configuring one VLAN at a time.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan 200-299**	Configures VLANs in the range 200 to 299 and enters the VLAN configuration mode.
Step 3	switch(config-vlan)# **exit**	Exits the VLAN configuration mode.

Example

This example shows how to configure 100 VLANs and name each of them:

```
switch# configure terminal
switch(config)# vlan 200-299
switch(config-vlan)# exit
switch(config)# vlan 201
switch(config-vlan)# name finance
switch(config-vlan)# exit
```

Configuring Layer 3 over vPC for F2E, F3 Modules

Before you begin

- Ensure that the peer-gateway in enabled and configured on both the peers and both the peers are running image that supports Layer 3 over vPC feature. If you enter the **layer3 peer-router** command without enabling the peer-gateway feature, a syslog message is displayed recommending you to enable the peer-gateway feature.
- Ensure that the peer link is up

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **layer3 peer-router**	Enables the Layer 3 device to form peering adjacency with both peers. **Note** Configure this command in both the peers.
Step 4	switch(config-vpc-domain)# **peer-gateway**	Enables Layer 3 forwarding for packets destined for the peer's gateway MAC address.
Step 5	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 6	(Optional) switch# **show vpc brief**	(Optional) Displays brief information about each vPC domain.
Step 7	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a Layer 3 over vPC for F2E, F3 modules:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# layer3 peer-router
switch(config-vpc-domain)# peer-gateway
switch(config-vpc-domain)# exit
```

This example shows how to verify if the Layer 3 over vPC for F2E, F3 modules feature is configured:

```
switch# show vpc brief
```

```
vPC domain id : 1
Peer status : peer adjacency formed ok
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : failed
Type-2 consistency status : success
vPC role : secondary
Number of vPCs configured : 2
Peer Gateway : Enabled
Peer gateway excluded VLANs : -
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Enabled (timeout = 240 seconds)
Operational Layer3 Peer : Enabled
```

Configuring a vPC Peer Gateway

From Cisco NX-OS Release 4.2(1) and later releases, you can configure vPC peer devices to act as the gateway for packets that are destined to the vPC peer device's MAC address.

When you attach a Layer 3 device to a vPC domain, the peering of routing protocols using a VLAN also carried on the vPC peer-link is not supported. If routing protocol adjacencies are needed between vPC peer devices and a generic Layer 3 device, you must use physical routed interfaces for the interconnection. Use of the vPC peer-gateway feature does not change this requirement.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **peer-gateway**	Enables Layer 3 forwarding for packets destined to the peer's gateway MAC address.
Step 4	(Optional) switch(config-vpc-domain)# **peer-gateway exclude-vlan** *backup-vlan-id*	From Cisco NX-OS Release 5.1(3), avoids software switching of transit VLAN traffic in a mixed chassis mode. See the "vPC Peer Gateway" section for more information.
Step 5	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.

	Command or Action	Purpose
Step 6	(Optional) switch# **show vpc brief**	Displays brief information about each vPC, including information about the vPC peer link..
Step 7	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring a Graceful Consistency Check

From Cisco NX-OS Release 5.2(1), 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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	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	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 5	(Optional) switch# **show vpc brief**	Displays brief information about each vPC domain.

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
```

Configuring vPC Shutdown

From Cisco NX-OS Release 7.2(0)D1(1), you can use the **shutdown** command to isolate a switch from a vPC complex before it is debugged, reloaded, or even removed physically, so that the vPC traffic passing through the peer vPC switch in the vPC complex is not affected.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **shutdown**	Shuts down the peer to isolate it for debugging, reloading, or physically removing it from the vPC complex, and enables the peer vPC switch to take over as the primary peer. Use the **no** form of this command to disable the feature.
Step 4	switch(config-vpc-domain)# **exit**	Exits vPC-domain configuration mode.

Example

This example shows how to enable the graceful consistency check feature:

```
switch# configure terminal
switch(config)# vpc domain 1
switch(config-vpc-domain)# shutdown
switch(config-vpc-domain)# exit
```

Configuring vPC Config Synchronization

Enabling vPC Configuration Synchronization

Before you begin

- You must create identical vPC domain IDs on both vPC peer switches.
- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **config-sync**	Enables vPC configuration synchronization. **Note** This command must be configured on both the primary and secondary switch.

The table below shows the process of configuration synchronization on switch 1 and switch 2:

Primary Switch	Secondary Switch

`switch-1# configure terminal`
`switch-1(config)# vpc domain 300`
`switch-1(config-vpc-domain)# config-sync`
	`switch-2# configure terminal`
	`switch-2(config)# vpc domain 300`
	`switch-2(config-vpc-domain)# config-sync`

Configuration synchronization is enabled on both switches in the same vPC domain.	
`switch-1# configure terminal` `switch-1(config)# spanning-tree mode mst`	
The above configuration is applied on the primary switch and is configuration synchronized to the secondary switch. The configuration is either successfully applied to both switches or will be failed on both.	

`switch-1# show running-config`
`...`
`spanning-tree mode mst`
`...`
	`switch-2# show running-config`
	`...`
	`spanning-tree mode mst`
	`...`

	`switch-2# configure terminal`
	`switch-2(config)# spanning-tree port type`
	`switch-2 default`

The configuration is applied on the secondary switch and is configuration synchronized to the primary switch. **Note** The configuration can be applied to either switch.	

`switch-1# show running-config`
`...`
`spanning-tree port type network default`
`...`
	`switch-2# show running-config`
	`...`
	`spanning-tree port type network default`
	`...`

Synchronizing Configuration for a Physical Port vPC

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Specifies the vPC physical port, and enters interface configuration mode.
Step 3	switch(config-if)# **vpc** *vpc-id* [**sync** {**export** | **import**}]	Moves port channel into a vPC and enters interface vPC configuration mode. The range is from 1 to 4096. • **sync export** enables the primary switch configuration to be exported to the secondary switch. • **sync import** enables the secondary switch configuration to be imported to primary switch.
Step 4	(Optional) switch(config-if)# **show running-config interface ethernet** *slot/port*	Displays the running configuration for the physical port.

Asymmetric Mapping

The table below shows the process of enabling configuration synchronization (asymmetric mapping) on the vPC physical port on the primary and the secondary switch:

Primary Switch	Secondary Switch

`switch-1# configure terminal`
`switch-1(config)# interface eth1/1`
`switch-1(config-if)# vpc 100`	

The physical port (ethernet1/1) is added to the vPC 100 domain on the primary switch. vPC 100 is not configured on the secondary switch. The configuration will not be synchronized until vPC 100 is added to the secondary switch.	

	`switch-2# configure terminal`
	`switch-2(config)# interface eth2/3`
	`switch-2(config-if)# vpc 100`

Following the configuration of vPC 100 to the secondary switch, the physical ports (interface ethernet2/3 on the secondary switch and interface ethernet1/1 on the primary switch) will be configuration synchronized.	

Symmetric Mapping

The table below shows the process of enabling configuration synchronization (symmetric mapping) on the vPC physical port on the primary and the secondary switch:

Primary switch	Secondary switch

`switch-1# configure terminal`
`switch-1(config)# interface eth1/1`
`switch-1(config-if)# vpc 100 symmetric`
	`switch-2# configure terminal`
	`switch-2(config)# interface eth1/1`

The physical port (ethernet1/1) is added to the vPC 100 domain on the primary switch. The physical port (ethernet 1/1) is also present on the secondary switch. The configuration of the physical port on both the primary and secondary switch will be kept in synchronization.	

`switch-1# show running-config interface eth1/10`
 	
`interface ethernet1/1`
`  switchport`
`  switchport mode trunk`
`  vpc 100`
	`switch-2# show running-config interface eth1/10`
	
	`interface ethernet1/1`
	`  switchport`
	`  switchport mode trunk`
	`  vpc 100`

Synchronizing Configuration of vPC Member Port Channel

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Selects the port channel that you want to use as the vPC peer link for this device, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport**	Configures the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **vpc** *vpc-id* [**sync** {**export** | **import**}]	Moves port channel into a vPC and enters interface vPC configuration mode. The range is from 1 to 4096. • **sync export** enables the primary switch configuration to be exported to the secondary switch.

	Command or Action	Purpose
		• **sync import** enables the secondary switch configuration to be imported to primary switch.
Step 5	(Optional) switch(config-if)# **show running-config interface port-channel** *channel-number*	Displays the running configuration for the port channel.

The table below shows the process of enabling configuration synchronization under port channel 10 on the primary and the secondary switch:

Primary Switch	Secondary Switch

`switch-1# configure terminal`
`switch-1(config)# interface port-channel 10`
`switch-1(config-if)# switchport`
`switch-1(config-if)# vpc 10`	

The configuration under port-channel 10 is configuration synchronized to the secondary switch. **Note** The **vpc** *number* command can be given first on either the primary or secondary switch.	

	`switch-2# show running-config interface po10`
	
	`interface port-channel10`
	`  switchport`
	`  vpc 10`

The configuration is applied on the secondary switch and is configuration synchronized to the primary switch. **Note** The configuration can be applied to either switch.	

	`switch-2# configure terminal`
	`switch-2(config)# interface port-channel 10`
	`switch-2(config-if)# switchport mode trunk`

The **show running-config interface port-channel** *channel-number* command shows that the configuration synchronization for port channel 10 is successful:	

`switch-1# show running-config interface port-channel 10`
 	
`interface port-channel10`
`  switchport`
`  switchport mode trunk`
`  vpc 10`
	`switch-2# show running-config interface port-channel 10`
	
	`interface port-channel10`
	`  switchport`
	`  switchport mode trunk`
	`  vpc 10`

Verifying vPC Configuration Synchronization

To verify vPC configuration synchronization, perform one of the following tasks:

Command	Purpose
show running-config vpc-config-sync	Displays whether config-sync is available or not.
show vpc config-sync cli syntax	Displays the list of commands that are able to be configuration synchronized.
show vpc config-sync database	Displays the configuration synchronization database.
show vpc config-sync merge status	Displays the merge status of the switch and of each vPC interface.
show vpc config-sync status	Displays the status of the last 10 operations of the vPC configuration synchronization process. • Displays merge status (success/failure). • Displays the last action done by the vPC configuration synchronization process and the result of that action.

Checking 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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **show vpc consistency-parameters** {**global** | **interface port-channel** *channel-number*}	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
```

Note Messages regarding the vPC interface configuration compatibility are also logged to the syslog.

Moving Other Port Channels into a vPC

Note 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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you are using a Layer 2 port channel.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Selects the port channel that you want to use as the vPC peer link for this device, and enters interface configuration mode.
Step 3	switch(config-if)# **vpc** *number*	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	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 5	(Optional) switch# **show vpc brief**	Displays brief information about each vPC domain.

	Command or Action	Purpose
Step 6	(Optional) switch# **copy running-config startup-config**	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
```

Enabling Certain vPC Commands Automatically

From Cisco NX-OS Release 6.2(2), you can automatically and simultaneously enable the following commands using the **mode auto** command: **peer-gateway**, **auto-recovery**, **fabricpath multicast load-balance**, **ip arp synchronize**, and **ipv6 nd synchronize**.

Note From Cisco NX-OS Release 6.2(2) and later releases, auto recovery is enabled by default. If you want to disable auto recovery in Release 6.2(2) and later releases, you must use the **no auto-recovery** command to explicitly disable auto recovery.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature vpc**	Enables vPCs on the device.
Step 3	switch(config)# **vpc domain** *domain-id*	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.
Step 4	switch(config-vpc-domain)# [**no**] **mode auto**	Enables the following commands simultaneously: **peer-gateway**, **auto-recovery**, **fabricpath multicast load-balance**, **ip arp synchronize**, and **ipv6 nd synchronize**.

	Command or Action	Purpose
		Use the **no** form of this command to disable the feature.
Step 5	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 6	switch(config)# **exit**	Exits global configuration mode.
Step 7	(Optional) switch# **show running-config vpc**	Displays information about the vPC, including the commands that are enabled.
Step 8	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to simultaneously enable the following commands: **peer-gateway**, **auto-recovery**, **fabricpath multicast load-balance**, **ip arp synchronize**, and **ipv6 nd synchronize**.

```
switch# configure terminal
switch# feature vpc
switch(config)# vpc domain 1
switch(config-vpc-domain)# mode auto

The following commands are executed:
peer-gateway ;
auto-recovery ;
ip arp synchronize ;
ipv6 nd synchronize ;
fabricpath multicast load-balance ;

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# show running-config vpc

!Command: show running-config vpc
!Time: Thu Feb 18 12:31:42 2013

version 6.2(2)
feature vpc

vpc domain 1
peer-gateway
auto-recovery
fabricpath multicast load-balance
ip arp synchronize
ipv6 nd synchronize
```

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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **system-mac** *mac-address*	Enters the MAC address that you want for the specified vPC domain in the following format: aaaa.bbbb.cccc.
Step 4	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 5	(Optional) switch# **show vpc role**	Displays the vPC system MAC address.
Step 6	(Optional) switch# **copy running-config startup-config**	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
```

Manually Configuring 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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **system-priority** *priority*	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	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 5	(Optional) switch# **show vpc role**	Displays the vPC system MAC address.
Step 6	(Optional) switch# **copy running-config startup-config**	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
```

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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **role priority** *priority*	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	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 5	(Optional) switch# **show vpc role**	Displays the vPC system priority.
Step 6	(Optional) switch# **copy running-config startup-config**	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
```

Configuring the Tracking Feature on a Single-Module vPC

From Cisco NX-OS Release 4.2, 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

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- 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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **track** *track-object-id*	Adds the previously configured track-list object with its associated interfaces to the vPC domain. See the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for information about configuring object tracking and track lists.
Step 4	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 5	(Optional) switch# **show vpc brief**	Displays information about the tracked objects.
Step 6	(Optional) switch# **copy running-config startup-config**	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
```

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 7000 Series device to restore vPC services when its peer fails to come on line.

Configuring Reload Restore

Note From Cisco NX-OS Release 5.2(1), 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.

From Cisco NX-OS Release 5.0(2), you can configure the Cisco Nexus 7000 Series device to restore vPC services when its peer fails to come online by using the reload restore command.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **reload restore** [**delay** *time-out*]	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	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 5	(Optional) switch# **show running-config vpc**	Displays information about the vPC, specifically the reload status.

	Command or Action	Purpose
Step 6	(Optional) switch# **show vpc consistency-parameters interface port-channel** *number*	Displays information about the vPC consistency parameters for the specified interface.
Step 7	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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

From Cisco NX-OS Release 5.2(1), you can configure the Cisco Nexus 7000 Series device to restore vPC services when its peer fails to come online by using the **auto-recovery** command.

Note From Cisco NX-OS Release 6.2(2) and later releases, auto recovery is enabled by default. If you want to disable auto recovery in Release 6.2(2) or a later release, you must use the **no auto-recovery** command to explicitly disable auto recovery.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **auto-recovery** [**reload-delay** *time*]	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	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 5	(Optional) switch# **show running-config vpc**	Displays information about the vPC, specifically the reload status.
Step 6	(Optional) switch# **show vpc consistency-parameters interface port-channel** *number*	Displays information about the vPC consistency parameters for the specified interface.
Step 7	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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
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. From Cisco NX-OS Release 5.2(1), 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 peer link or peer-keepalive failure. The orphan ports are restored when the vPC is restored.

Note From Cisco NX-OS Release 6.2 and earlier, configure the vPC orphan-port command on all the member ports and bundle them into the port channel. For later releases, configure the command directly on the port-channel interfaces.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **show vpc orphan-ports**	Displays a list of the orphan ports.
Step 3	switch(config)# **interface port-channel** *channel-number*	Selects the port channel that you want to use as the vPC peer link for this device, and enters interface configuration mode.
Step 4	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	switch(config-if)# **exit**	Exits interface configuration mode.
Step 6	(Optional) switch# **copy running-config startup-config**	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
```

Configuring the vPC Peer Switch

You can configure the Cisco Nexus 7000 Series device to make a pair of vPC devices appear as a single STP root in the Layer 2 topology. This section includes the following topics:

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.

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.

Before you begin

- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	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.

	Command or Action	Purpose
Step 4	switch(config-vpc-domain)# **spanning-tree vlan** *vlan-range* **priority** *value*	Configures the bridge priority of the VLAN. Valid values are multiples of 4096. The default value is 32768.
Step 5	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 6	(Optional) switch# **show spanning-tree summary**	Displays a summary of the spanning tree port states including the vPC peer switch.
Step 7	(Optional) switch# **copy running-config startup-config**	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
```

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 (for more information, see the Cisco Nexus 7000 Series NX-OS Layer 2 Switching Command Reference) 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.

Note 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.

Before you begin

- Ensure that you are in the correct VDC (if you are not in the correct VDC, use the **switchto vdc** command).
- Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree pseudo-information**	Configures the spanning tree pseudo information.
Step 3	switch(config-pseudo)# **vlan** *vlan-range* **designated priority** *value*	Configures the designated bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440.
Step 4	switch(config-pseudo)# **vlan** *vlan-range* **root priority** *value*	Configures the root bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440.
Step 5	switch(config)# **vpc domain** *domain-id*	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.
Step 6	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	switch(config-vpc-domain)# **exit**	Exits vpc-domain configuration mode.
Step 8	(Optional) switch# **show spanning-tree summary**	Displays a summary of the spanning tree port states including the vPC peer switch.
Step 9	(Optional) switch# **copy running-config startup-config**	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
```

Enabling Distribution for vPC

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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.
Step 3	switch(config-vpc-domain)# **config-sync**	Enables the vPC config-sync on the switch and registers with the CFS for physical-ethernet (CFSoE). **Note** Repeat the configuration of the **config-sync** command on the other vPC peer as well.
Step 4	switch(config-vpc-domain)# **exit**	Exits vPC-domain configuration mode.
Step 5	switch(config-vpc-domain)# **vpc config-sync re-emerge** [**sync** { **export**|**import** }]	(Optional) Triggers the merging of configuration with the peer switch if the current merge has failed. **Note** You can use the **sync export** option to apply the local switch configuration to the peer switch. You can use the **sync import** option to apply the remote switch configuration to the local switch.
Step 6	switch(config-vpc-domain)# **vpc config-sync re-emerge interface port-channel** *channel-name* [**sync** { **export**|**import** }]	(Optional) Triggers the merging of **interface port-channel** configuration with the peer switch if the current merge has failed. **Note** You can use the **sync export** option to apply the local **interface port-channel channel-number** command configuration with the peer switch. You can use the **sync import** option to apply the remote **interface port-channel channel-number** command configuration to the local switch.

	Command or Action	Purpose
Step 7	switch(config-vpc-domain)# **vpc config-sync re-emerge interface** *type slot/port* [**sync** { **export**|**import** }]	(Optional) Triggers the merging of **interface ethernet** with the peer switch if the current merge has failed. **Note** You can use the **sync export** option to apply the local **interface ethernet slot/port** command configuration with the peer switch. You can use the **sync import** option to apply the remote **interface ethernet slot/port** command configuration to the local switch.
Step 8	switch(config-vpc-domain)# **exit**	Exits vPC domain configuration mode.
Step 9	switch(config)# **exit**	Exits global configuration mode.
Step 10	switch(config)# **show vpc config-sync merge status**	Displays the status of the configuration merge with the peer switch.

Example

```
switch# configure terminal
switch(config)# vpc domain 1
switch(config-vpc-domain)# config-sync
switch(config-vpc-domain)# vpc config-sync re-merge sync export
switch(config)# vpc config-sync re-merge interface port-channel 1 sync export
switch(config)# vpc config-sync re-merge interface ethernet 1/1 sync export  import
switch(config)# exit
switch(config)# show vpc config-sync merge status
```

Configuring FCoE Over a Physical Port vPC

Configure Physical Port vPC Interfaces

Perform the following task to configure a physical port vPC interface in the Ethernet VDC. Repeat this task to configure the peer VDC.

Before you begin

- Ensure that you have enabled the vPC feature.
- Ensure that you have configured the per link port channel and port channel members.
- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port-list*	Specifies an Ethernet interface and enters interface configuration mode. The range is from 1 to 253 for the slot and from 1 to 128 for the port.
Step 3	switch(config-if)# **switchport**	Configures the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **switchport mode trunk**	Specifies the trunking VLAN interface in Layer 2. A trunk port can carry traffic in one or more VLANs (based on the trunk allowed VLAN list configuration) on the same physical link.
Step 5	switch(config-if)# **switchport trunk allowed vlan** *vlan-list*	Configures a list of allowed VLANs on the trunking interface.
Step 6	switch(config-if)# **spanning-tree port type network**	Configures the interface that connects to a Layer 2 switch as a network spanning tree port.
Step 7	switch(config-if)# **vpc** *number*	Moves port channels into a vPC and enters interface vPC configuration mode. The range of the number argument is from 1 to 4096.
Step 8	switch(config-if-vpc)# **lapc mode active**	Enables LAPC on the peer link member interfaces on which you configured the **channel group mode active** command.
Step 9	switch(config-if-vpc)# **no shutdown**	Brings the port administratively up.

Example

These examples show how to configure a physical port vPC in an Ethernet VDC:

```
switch-eth(config)# feature vpc

switch-eth(config)# interface port-channel 1
switch-eth(config-if)# switchport
switch-eth(config-if)# switchport mode trunk
switch-eth(config-if)# switchport trunk allowed vlan 10-20
switch-eth(config-if)# spanning-tree port type network
switch-eth(config-if)# vpc peer-link

switch-eth(config)# interface Ethernet3/21
switch-eth(config-if)# switchport
switch-eth(config-if)# switchport mode trunk
```

```
switch-eth(config-if)# switchport trunk allowed vlan 10-20
switch-eth(config-if)# channel group 1 mode active
switch-eth(config-if)# no shutdown

switch-eth(config)# interface Ethernet3/1
switch-eth(config-if)# switchport
switch-eth(config-if)# switchport mode trunk
switch-eth(config-if)# switchport trunk allowed vlan 10-20
switch-eth(config-if)# vpc 10
switch-eth(config-if-vpc)# lacp mode active
switch-eth(config-if-vpc)# no shutdown
```

These examples show how to configure a physical port vPC in the peer VDC:

```
switch-eth(config)# feature vpc

switch-eth(config)# interface port-channel 1
switch-eth(config-if)# switchport
switch-eth(config-if)# switchport mode trunk
switch-eth(config-if)# switchport trunk allowed vlan 10-20
switch-eth(config-if)# spanning-tree port type network
switch-eth(config-if)# vpc peer-link

switch-eth(config)# interface Ethernet4/21
switch-eth(config-if)# switchport
switch-eth(config-if)# switchport mode trunk
switch-eth(config-if)# switchport trunk allowed vlan 10-20
switch-eth(config-if)# channel group 1 mode active
switch-eth(config-if)# no shutdown

switch-eth(config)# interface Ethernet4/1
switch-eth(config-if)# switchport
switch-eth(config-if)# switchport mode trunk
switch-eth(config-if)# switchport trunk allowed vlan 10-20
switch-eth(config-if)# vpc 10
switch-eth(config-if-vpc)# lacp mode active
switch-eth(config-if-vpc)# no shutdown
```

Configuring Hitless vPC Role Change

Before you begin

- Enable the vPC feature
- Ensure vPC peer link is up
- Verify the role priority of devices

Procedure

Step 1 Enable hitless vPC role change feature.

switch# **vpc role preempt**

Step 2 (Optional) Verify hitless vPC role change feature.

switch# **show vpc role**

Configuring Hitless vPC Role Change

This example on how to configure hitless vPC role change:

```
! The following is an output from the show vpc role command before the
vPC hitless feature is configured !

switch# show vpc role

vPC 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# vpc role preempt

! The following is an output from the show vpc role command after the
vPC hitless feature is configured !

switch# 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
```

Upgrading Line Card Modules for vPC

To upgrade to a new line card module for a virtual port channel (vPC), use one of the following methods:

- Upgrade line card modules using the ISSU method.
- Upgrade line card modules using the reload method.

Upgrading a Line Card Module Using the ISSU Method

In this task, the primary switch is Switch A, and the secondary switch is Switch B.

Note

- Traffic outage might occur on orphan ports when a vPC peer is isolated.
- Multicast receivers behind the vPC might experience traffic outages.
- Ensure that there are alternate paths from core routes to each vPC peer.
- Ensure that the new line card module has the same slot ID and number as the old line card module.

Before you begin

Before you upgrade a line card module, refer to the Cisco Nexus 7000 Series NX-OS Release Notes document, to see the supported Cisco NX-OS release version for a line card module.

Procedure

Step 1 Perform an ISSU upgrade to a supported Cisco NX-OS release version for a new line card module on both the switches. Perform this task one at a time on both the switches. For information on supported release version for a line card module type, see the Cisco Nexus 7000 Series NX-OS Release Notes document. For information on how to perform an ISSU upgrade, see the Cisco Nexus 7000 Series NX-OS Software Upgrade and Downgrade Guide.

Step 2 On both the switches, move the peer-keepalive link out of the existing module, and use the management interface for the peer-keepalive link.

Example:

```
switch# configure terminal
switch(config)# vpc domain <domain-id>
switch(config-vpc-domain)# peer-keepalive destination <peer-switch management-ip>
```

Step 3 Enable the hidden commands on both the switches, one at a time.

Example:

```
switch# configure terminal
switch(config)# vpc domain <domain-id>
switch(config-vpc-domain)# bypass module-check
```

Step 4 Copy the running configuration to the startup configuration on both the switches.

Example:

```
switch# copy running-config startup-config vdc-all
```

Step 5 On the secondary switch (Switch B), shut down the vPC legs. Perform this action in batches and wait until all the traffic is converged. All traffic is now on the primary switch (Switch A).

Example:

```
switch(config)# interface port-channel <channel-number>
switch(config-if)# shutdown
```

Step 6 On the secondary switch (Switch B), shut down all the ports going to core devices. Perform this action in batches and wait until all the traffic is converged.

Step 7 On the secondary switch (Switch B), shut down the vPC peer link.

Step 8 On the secondary switch (Switch B), save the running configuration to a file on bootflash.

Example:

```
switch# copy running-config bootflash:run-cfg-SwitchB.txt vdc-all
```

Step 9 On the secondary switch (Switch B), edit the saved configuration file to change the Virtual Device Context (VDC) type from an existing module to a new module.

For more information on Cisco NX-OS release support for a module type, see the Cisco Nexus 7000 Series NX-OS Release Notes document.

This example shows that the VDC type has changed from an existing module (F2 or F2e) to a new module (F3):

```
Edit { vdc <xyx>
      limit-resource module-type "f3" }
```

Step 10 On the secondary switch (Switch B), replace the old line card with the new line card module.

Step 11 On the secondary switch (Switch B), reconnect the vPC leg ports to the new module. Ensure that all the ports have the same number as the old line card module.

Step 12 On the secondary switch (Switch B), reconfigure the respective ports on the new module using the saved configuration file on bootflash. Ensure that vPC leg ports are in shut state.

Example:

```
switch# copy bootflash:run-cfg-SwitchB.txt running-config
```

Step 13 On the secondary switch, copy the running configuration to the startup configuration on the admin VDC.

Example:

```
switch# copy running-config startup-config vdc-all
```

Step 14 On the secondary switch (Switch B), bring up the vPC peer link. Ensure that the vPC peer link speed is the same on both the switches.

Ensure that vPC is up and Switch A is the primary switch and Switch B is the secondary switch.

Step 15 On the secondary switch (Switch B), bring up the vPC leg ports. Perform this task in batches and wait for all the traffic to converge.

Step 16 On the secondary switch (Switch B), bring up all the ports going to the core device. Perform this task in batches and wait for all the traffic to converge.

Step 17 On the secondary switch (Switch B), clear all the dynamic MAC entries from the MAC address table.

Example:

```
switch# clear mac address-table dynamic
switch# test l2fm dump smac
```

Migration to the new module on the secondary switch is completed.

Step 18 On the primary switch (Switch A), shut down the vPC legs. Perform this action in batches and wait until all the traffic is converged.

Example:

```
switch(config)# interface port-channel <channel-number>
switch(config-if)# shutdown
```

All the traffic is now on the secondary switch (Switch B).

Step 19 On the secondary switch (Switch B), change the vPC role priority to match the primary switch.

Example:

```
switch(config)# vpc-domain <domain-id>
switch(config-vpc-domain)# role priority <priority-id>
```

Step 20 On the primary switch (Switch A), shut down all the ports going to the core devices. Perform this action in batches and wait until all the traffic is converged. All traffic is now on the secondary switch (Switch B).

Step 21 On the primary switch (Switch A), reconfigure the vPC peer-keepalive link by configuring a dummy IP address.

Example:

```
switch# configure terminal
switch(config-if)# vpc domain <domain-id>
switch(config-if)# peer-keepalive destination <dummy-ip>
```

Step 22 On the primary switch (Switch A), shut down the vPC peer link.

vPC role change takes place without any disruption because of the sticky bit feature on the Switch B.

Step 23 On Switch A, save the running configuration to a file on bootflash.

Example:

```
switch# copy running-config bootflash:run-cfg-SwitchA.txt vdc-all
```

Step 24 Edit the saved configuration file to change the VDC type from the existing module to the new module.

For information on Cisco NX-OS release support for a module type, see the Cisco Nexus 7000 Series NX-OS Release Notes document.

Example:

This example shows that the VDC type is changed from F2 to F3 module.

```
Edit { vdc <xyx>
      limit-resource module-type “f3” }
```

Step 25 On the primary switch (Switch A), replace the old line card with the new line card module.

Step 26 On the primary switch (Switch A), reconnect the vPC leg ports to the new module. Ensure that all the ports have the same number as the old line card module.

Step 27 On the primary switch (Switch A), reconfigure the respective ports on the new module using the saved configuration file on bootflash.

Example:

```
switch# copy bootflash:run-cfg-SwitchA.txt running-config
```

Note Ensure that all the vPC leg ports are in shut state.

Step 28 On the primary switch (Switch A), copy the running configuration to the startup configuration on the Admin virtual device context (VDC).

Example:

```
switch# copy running-config startup-config vdc-all
```

Step 29 On the primary switch (Switch A), bring up the vPC peer-keepalive link by configuring the peer-keepalive destination address back to the management IP of Switch B.

Example:

```
switch# configure terminal
switch(config-if)# vpc domain <domain-id>
```

```
switch(config-if)# peer-keepalive destination <management-ip peer-device
```

Step 30 On the primary switch (Switch A), bring up the vPC peer link.

Note Ensure that the vPC peer-link speed configuration is same on both the switches.

All the traffic is on the secondary switch (Switch B).

Step 31 On the primary switch (Switch A), bring up the vPC leg ports. Perform this task in batches and wait for all the traffic to converge.

All the traffic is load balanced on both the switches.

Step 32 On the primary switch (Switch A), bring up all the ports going to the core device. Perform this task in batches and wait for all the traffic to converge.

Step 33 Disable the hidden commands on both the switches. Perform this step one at a time on both the switches.

Example:

```
switch# configure terminal
switch(config)# vpc-domain <domain-id>
switch(config-vpc-domain)# no bypass module-check
```

Step 34 On both the switches, reconfigure the peer-keepalive link on the new card modules.

Step 35 Copy the running configuration to the startup configuration on the Admin VDC on both the switches.

Example:

```
switch# copy running-config startup-config vdc-all
```

Step 36 On the primary switch (Switch A), clear all the dynamic MAC entries from the MAC address table.

Example:

```
switch# clear mac address-table dynamic
switch# test l2fm dump smac
```

Step 37 On the secondary switch (Switch B), run the **test l2fm dump smac** command to view any errors.

Example:

```
switch# test l2fm dump smac
```

Migration to the new module on the primary switch is completed.

Migration from existing line card module to a new module is completed on both the switches.

Upgrading Line Card Modules Using the Reload Method

To upgrade from an existing line card module to a new line card module for vPC using the reload method, perform the following tasks:

1. Install Cisco NX-OS image on vPC peers
2. Install line card module using the reload method

Before you plan to upgrade a line card module, refer the Cisco Nexus 7000 Series NX-OS Release Notes document, to see the supported Cisco NX-OS release version for a line card module.

Installing a Cisco Image on vPC Peers

Perform this task on all the vPC peer devices. Switch A is the primary switch, and Switch B is the secondary switch in this task.

Note Traffic outage might occur on orphan ports when a vPC peer is isolated. Multicast receivers behind the vPC might experience traffic outages (30 to 40 seconds).

Before you begin

- Before you upgrade a line card module, refer to the Cisco Nexus 7000 Series NX-OS Release Notes document to see the supported Cisco NX-OS release version for a line card module.
- Ensure that the feature vPC is enabled on both the primary switch and the secondary switch.
- Ensure that there are alternate paths from core routes to each of the vPC peers.

Procedure

Step 1 Set equal vPC role priority on both the vPC peer devices.

Example:

```
switch# configure terminal
switch(config)# vpc domain <domain-id>
switch(config-vpc-domain)# role priority <priority-id>
```

Step 2 Set the **auto-recovery reload-delay** value, in seconds, to maximum delay time on both the switches.

Example:

```
switch(config-vpc-domain)# auto-recovery reload-delay 84600
```

Step 3 Change the system boot parameters to boot the system from the Cisco NX-OS release verison that is supported on the new module on both the switches.

Example:

This example shows that the Cisco NX-OS 6.2(16) image is used for the Cisco Nexus F3 module:

```
switch(config)# no boot kickstart
switch(config)# no boot system
switch(config)# boot kickstart bootflash://n7000-s2-kickstart.6.2.16.bin
switch(config)# boot system bootflash://n7000-s2-dk9.6.2.16.bin
```

For information on the supported release version for a module type, see the Cisco Nexus 7000 Series NX-OS Release Notes document.

Step 4 On the secondary switch (Switch B), shut down the vPC legs. Perform this action in batches and wait until all the traffic is converged.

Example:

```
switch(config)# interface port-channel <channel-number>
switch(config-if)# shutdown
```

All the traffic is now on the primary switch (Switch A).

Step 5 On the secondary switch (Switch B), copy the running configuration to the start up configuration for an Admin VDC.

Example:

```
switch# copy running-config startup-config vdc-all
```

Step 6 On the secondary switch (Switch B), reboot the system with the new Cisco NX-OS image. Wait for the system to boot up and for the Layer 3 links to come up.

Example:

```
switch# reload
```

Step 7 On the secondary switch (Switch B), bring the vPC legs up again. Perform this action in batches and wait until all the traffic is converged.

Example:

```
switch(config)# interface port-channel <channel-number>
switch(config-if)# no shutdown
```

Step 8 On the primary switch (Switch A), shut down the vPC legs. Perform this action in batches and wait until all the traffic is converged.

Example:

```
switch(config)# interface port-channel <channel-number>
switch(config-if)# shutdown
```

Step 9 On the primary switch (Switch A), copy the running configuration to the start up configuration for an Admin VDC.

Example:

```
switch# copy running-config startup-config vdc-all
```

Step 10 On the primary switch (Switch A), reboot the system with the new Cisco NX-OS image. Wait for the system to boot up and for the Layer 3 links to come up.

Example:

```
switch# reload
```

Step 11 On the primary switch (Switch A), bring the vPC legs up again. Perform this action in batches and wait until all the traffic is converged.

Example:

```
switch(config)# interface port-channel <channel-number>
switch(config-if)# no shutdown
```

Traffic is load balanced between the primary switch (Switch A) and the secondary switch (Switch B).

Switch B takes on the role of the operational primary, and Switch A takes on the role of the operational secondary.

Installing a Line Card Module on a vPC Peer Using the Reload Method

Before you begin

- Ensure that you have installed a compatible Cisco NX-OS release version on the vPC peers. For more information, on how to install a Cisco NX-OS release version using the reload method, see Installing a Cisco Image on vPC Peers, on page 286. For more information on the compatible Cisco NX-OS release version for a line card module type, refer to the Cisco Nexus 7000 Series NX-OS Release Notes document.
- Ensure that the new line card module has the same slot ID and number as the old line card module.

Note In this task, Switch A is the operational secondary, and Switch B is the operational primary switch.

Procedure

Step 1 Set equal vPC role priority on both the switches.

Example:

```
switch(config)# vpc-domain <domain-id>
switch(config-vpc-domain)# role priority <priority-id>
```

Step 2 Set the **auto-recovery reload-delay** value , in seconds, to maximum delay time on both the switches.

Example:

```
switch(config-vpc-domain)# auto-recovery reload-delay 86400
```

Step 3 Enable the hidden commands on both the switches, one at a time.

Example:

```
switch# configure terminal
switch(config)# vpc domain <domain-id>
switch(config-vpc-domain)# bypass module-check
```

Step 4 Copy the running configuration to the startup configuration on the Admin VDC on both the switches.

Example:

```
switch# copy running-config startup-config vdc-all
```

Step 5 On the operational secondary (Switch A) switch, shut down the vPC legs. Perform this action in batches and wait until all the traffic is converged.

Example:

```
switch(config)# interface port-channel <channel-number>
switch(config-if)# shutdown
```

All the traffic is on Switch B.

Step 6 Save the running configuration to a file on bootflash and transfer the configuration file outside the switch (Switch A).

Example:

```
switch# copy running-config bootflash:run-cfg-SwitchA.txt vdc-all
switch# copy bootflash:run-cfg-SwitchA.txt tftp://server/run-cfg-SwitchA.txt vrf management
```

Step 7 On the operational secondary switch, edit the saved configuration file to change the VDC type from an existing module to a new module. Copy the configuration file back to the switch (Switch A).

Example:

This example show that the VDC type is changed from F2 to F3 module:

```
Edit { vdc <xyx>
      limit-resource module-type "f3" }

switch# copy tftp://server/ run-cfg-SwitchA.txt bootflash:run-cfg-SwitchA.txt vrf management
```

For information on the Cisco NX-OS release support for a module type, see the Cisco Nexus 7000 Series NX-OS Release Notes document.

Step 8 Power off the operational secondary switch (Switch A) and physically replace the existing module with the new module on the switch.

Step 9 Power on the switch (Switch A) and wait for the system to go online.

Ensure that the Admin VDC is active. On the Admin VDC, reconfigure the new module ports using the saved configuration file. Ensure that all the ports have the same number as the old line card module.

Ensure that all the vPC leg ports are in shut state, and the vPC peer link and the Layer 3 links are up.

Example:

```
switch# copy bootflash:run-cfg-SwitchA.txt running-config
```

Step 10 Bring up the vPC legs on the operational secondary (Switch A). Perform this task in batches and wait for all the traffic to converge.

Example:

```
switch# interface port-channel <channel-number>
Switch# no shutdown
```

Step 11 On the operational primary (Switch B) switch, shut down the vPC legs. Perform this action in batches and wait until all the traffic is converged.

Example:

```
switch(config)# interface port-channel <channel-number>
switch(config-if)# shutdown
```

All the traffic is on Switch A.

Step 12 Save the running configuration to a file on bootflash and transfer the configuration file outside the switch (Switch B).

Example:

```
switch# copy running-config bootflash:run-cfg-SwitchB.txt vdc-all
switch# copy bootflash:run-cfg-SwitchA.txt tftp://server/run-cfg-SwitchB.txt vrf management
```

Step 13 On the operational primary switch (Switch B), edit the saved configuration file to change the VDC type from an existing module to a new module. Copy the configuration file back to the switch (Switch B).

Example:

This example shows that the VDC type is changed from F2 to F3 module:

```
Edit { vdc <xyx>
      limit-resource module-type "f3" }
```

```
switch# copy tftp://server/ run-cfg-SwitchB.txt bootflash:run-cfg-SwitchB.txt vrf management
```

For information on the Cisco NX-OS release support for a module type, see the Cisco Nexus 7000 Series NX-OS Release Notes document.

Step 14 Power off the operational primary switch (Switch B) and physically replace the existing module with the new module on the switch.

Step 15 Power on the switch (Switch B) and wait for the system to go online.

Note Ensure that the Admin VDC is active. On the Admin VDC, reconfigure the new module ports using the saved configuration file. Ensure that all the ports have the same number as the old line card module.

Ensure that all the vPC leg ports are in shut state, and the vPC peer link and the Layer 3 links are up.

Example:

```
switch# copy bootflash:run-cfg-SwitchB.txt running-config
```

Step 16 Bring up the vPC legs on the operational primary (Switch B). Perform this task in batches and wait for all the traffic to converge.

Switch A resumes the role of a primary switch and Switch B takes on the role of a secondary switch. Traffic is load balanced between both the switches.

Example:

```
switch# interface port-channel <channel-number>
Switch# no shutdown
```

Step 17 Disable the hidden commands on both the switches. Perform this step one at a time on both the switches.

Example:

```
switch# configure terminal
switch(config)# vpc-domain <domain-id>
switch(config-vpc-domain)# no bypass module-check
```

Step 18 Copy the running configuration to the startup configuration on the Admin VDC on both the switches.

Example:

```
switch# copy running-config startup-config vdc-all
```

Migration from existing line card module to a new module is completed on both the switches.

Verifying the vPC Configuration

Use the information in the following table to verify the vPC configuration:

Table 46: Verifying the vPC Configuration

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.
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.

For detailed information about the fields in the output from these commands, see the Cisco Nexus 7000 Series NX-OS Interfaces Command Reference.

Verifying Physical Port vPC on F2, F3, and FEX

Use the information in the following table to verify the physical port vPC on F2, F3, and FEX:

Table 47: Verifying Physical Port vPC on F2, F3, and FEX

Command	Purpose
show vpc brief	Displays brief information about the vPCs.
show lacp port-vpc summary	Displays the LACP status for the physical port VPC, such as the vPC ID, physical port, and the LACP port state details.
show lacp counters	Displays the LACP counters for port-channels and physical port vPC interfaces.
show lacp counters interface *name number*	Displays the LACP counters on a physical interface or port-channel interface depending on the interface name.
show lacp neighbor	Displays LACP neighbor information for the port.

Command	Purpose
show lacp neighbor interface *name number*	Displays the neighbors of ports that are configured on a physical interface.

This example shows how to verify brief information about the vPCs:

```
switch# show vpc brief

vPC status
----------------------------------------------------------------------------------
id   Port          Status     Consistency   Reason           Active    vlans
----------------------------------------------------------------------------------
1    Ethernet1/1    up        success       - - - -          200-250,  900-1000
```

This example shows how to verify LACP status for the physcial port VPC, such as the vPC ID, physical port, and the LACP port state details:

```
switch# show lacp port-vpc summary

Flags:         D - Down                    P - up
               s - Suspended              H - Hot-standby (LACP only)

VPC-Id                  Member Port
1                       Ethernet 1/1(P)
2                       Ethernet 1/2(H)
3                       Ethernet 1/3(s)
```

This example shows how to verify LACP counters for port-channel and physical port vPC interfaces:

```
switch# show lacp counters

                    LACPDUs         Marker      Marker Response    LACPDUs
Port              Sent   Recv     Sent   Recv     Sent   Recv      Pkts Err
--------------------------------------------------------------------------------
Ethernet2/1
Ethernet2/1        1677   1804     0      0        0      0         0

port-channel2
Ethernet2/2        1677   1808     0      0        0      0         0
```

This example shows how to verify the LACP counters on a physical interface:

```
switch# show lacp counters  interface ethernet 1/1

LACPDUs         Marker      Marker Response    LACPDUs
Port              Sent   Recv     Sent   Recv     Sent   Recv      Pkts Err
--------------------------------------------------------------------------------
Ethernet1/1
Ethernet1/1        17466  17464    0      0        0      0         0
```

This example shows how to verify the neighbors of ports that are configured both as a vPC and as a port-channel member:

```
switch# show lacp neighbor
```

```
Flags:  S - Device is sending Slow LACPDUs F - Device is sending Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Eth1/1 neighbors
Partner's information
           Partner               Partner                     Partner
Port       System ID             Port Number     Age         Flags
Eth1/1     32768,2-0-0-0-0-66    0x2402          41595       SA

           LACP Partner          Partner                     Partner
           Port Priority         Oper Key                    Port State
           32768                 0x91                        0x3d
```

This example shows how to verify the neighbors of ports that are configured on the physical interface:

```
switch# show lacp neighbor interface ethernet 1/1

Flags:  S - Device is sending Slow LACPDUs F - Device is sending Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Eth1/1 neighbor
Partner's information
           Partner               Partner                     Partner
Port       System ID             Port Number     Age         Flags
Eth1/1     32768,0-26-98-14-e-c1 0x207           13          SA

           LACP Partner          Partner                     Partner
           Port Priority         Oper Key                    Port State
           32768                 0x0                         0x3d
```

Monitoring vPCs

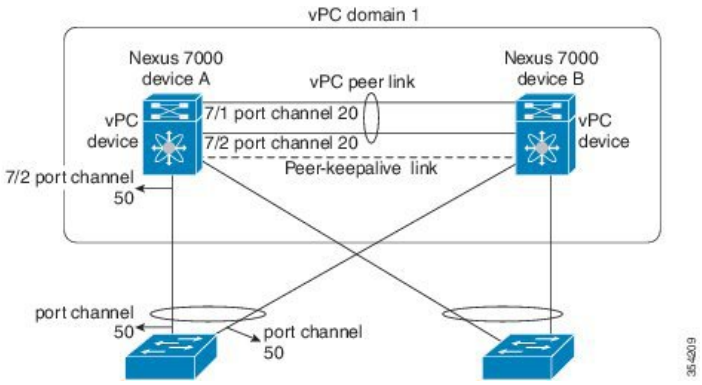
Use the **show vpc statistics** command to display vPC statistics.

Note This command displays the vPC statistics only for the vPC peer device that you are working on.

Configuration Examples for vPCs

This example shows how to configure vPC on device A as shown in the figure below:

Figure 28: 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 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)# rate-mode dedicated
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)#
```

3. (Optional) Configure the second, redundant interface that you want to be a 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)# rate-mode dedicated
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 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 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

Table 48: Related Documents

Related Topic
Cisco Nexus 7000 Series NX-OS Interfaces Command Reference
Cisco Nexus 7000 Series NX-OS System Management Configuration Guide
Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide
Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 7000 Series Switches, Release 6.x
Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide
Cisco NX-OS Licensing Guide
VLANs, MAC address tables, private VLANs, and the Spanning Tree Protocol. Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide
Cisco Nexus 7000 Series NX-OS FabricPath Command Reference
Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide
Cisco Nexus 7000 Series NX-OS Release Notes

Standards

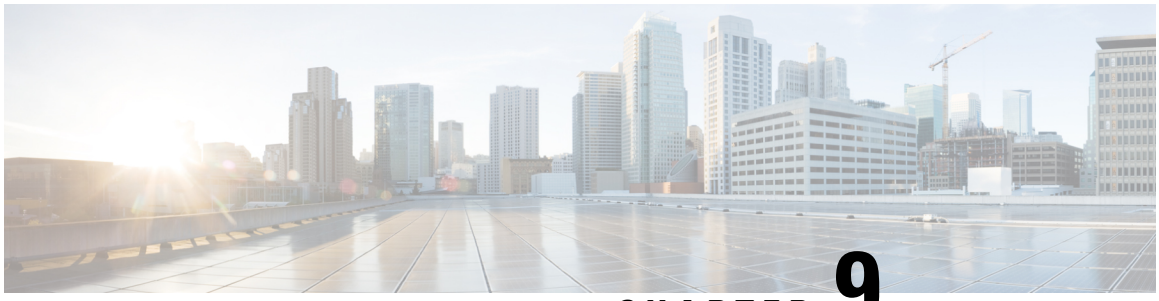
Table 49: Standards

Standards	Title
IEEE 802.3ad	—

MIBs

Table 50: MIBs

MIBs	MIBs Link
• IEEE8023-LAG-CAPABILITY • CISCO-LAG-MIB	To locate and download MIBs, go to: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.s



CHAPTER 9

Configuring Interfaces in Breakout Mode

This chapter describes how to configure interfaces in breakout mode.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Feature History for Breakout

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 51: Feature History for Breakout

Feature Name	Release	Feature Information
QSA support for M/F4 modules	8.4(2)	Added support to bring up the link with CVR-QSFP-SFP10G adapter in a M3/F4 100G module.
Breakout	8.3(1)	Added support for the Breakout feature that enables splitting of 40 Gigabit ethernet ports on the Cisco Nexus 7700 M3-Series 12-port 100-Gigabit Ethernet I/O module.

Feature Name	Release	Feature Information
Breakout	6.2(6) 6.2(8)	Added support for the Breakout feature on the Cisco Nexus 7000 Series Switches. Added support for the Breakout feature on the Cisco Nexus 7700 Switches.

Information About Breakout

The Cisco Nexus 7000 Series Switches and the Cisco Nexus 7700 Series Switches support the Breakout feature. Breakout enables a 40 Gigabit Ethernet port to be split into four independent and logical 10 Gigabit Ethernet ports. Breakout is supported on an active Twinax (1 to 10 m) cable or a multimode fiber cable (SR4 optic cable with an MTP connector or an MPO connector). For the list of transceivers that are supported on Cisco Nexus 7000 Series switches, refer the Cisco Nexus 7000 Series NX-OS Release Notes.

Note When the Breakout feature is configured, the configuration for the corresponding interface is removed.

The Breakout feature that enables a 40 Gigabit Ethernet port to be split into four independent and logical 10 Gigabit Ethernet ports is supported in the following modules:

- Cisco Nexus 7000 F3-Series 12-Port 40 Gigabit Ethernet Module
- Cisco Nexus 7000 M2-Series 6-Port 40 Gigabit Ethernet Module
- Cisco Nexus 7700 F3-Series 24-Port 40 Gigabit Ethernet Module

Starting from Cisco NX-OS Release 8.3(1), the Breakout feature that enables a 40 Gigabit Ethernet port to be split into four independent and logical 10 Gigabit Ethernet ports is also supported in the Cisco Nexus 7700 M3-Series 12-port 100-Gigabit Ethernet I/O module.

Guidelines and Limitations for Breakout

- In a Cisco Nexus 7700 F3-Series 24-Port 40 Gigabit Ethernet Module, you can break out any 19 ports of the available 24 ports.
- In a Cisco Nexus 7700 M3-Series 24-port 40-Gigabit Ethernet Module, you can break out any 23 ports of the available 24 ports.
- Before swapping a line card type, remove the breakout configurations configured for current line card ports. Use the **no interface breakout module** *slot* **port** *port-range* command to remove the breakout configurations.
- To bring up the link with CVR-QSFP-SFP10G adapter in a M3/F4 100G module, you need to perform the 10x4g breakout configuration on the interface. In older release versions with CVR-QSFP-SFP10G, the adapter link remains down. After ISSU to Cisco NX-OS Release 8.4(2) the link will still remain down. Perform OIR to bring up the link.
- QSFP-100G-DR-S and QSFP-100G-FR-S optics does not support breakout.

Configuring Breakout in a Port

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface breakout module** *slot* **port** *port-range* **map** {**10g-4x** | **25g-4x**}	Configures the Breakout feature for a port. The **10g-4x** keyword enables a 40 Gigabit Ethernet port to be split into four independent and logical 10 Gigabit Ethernet ports. The **25g-4x** keyword enables a 100 Gigabit Ethernet port to be split into four independent and logical 25 Gigabit Ethernet ports. • *slot*—Slot number of port depending on the chassis model. • *port-range*—Single port or range of ports on which breakout is configured.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the changes by copying the running configuration to the startup configuration.

Example

This example shows how to break out a 40 Gigabit Ethernet port into four 10 Gigabit Ethernet ports:

```
switch# configure terminal
switch(configure)# interface breakout module 1 port 12 map 10g-4x
switch(configure)# copy running-config startup-config
```

This example shows how to break out a 100 Gigabit Ethernet port into four 25 Gigabit Ethernet ports:

```
switch# configure terminal
switch(configure)# interface breakout module 1 port 8 map 25g-4x
switch(configure)# copy running-config startup-config
```

Removing the Breakout Configuration

Before you begin

Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no interface breakout module** *slot* **port** *port-range* **map** {**10g-4x** | **25g-4x**}	Removes the breakout configuration for a port module and returns the interface to the 40 Gigabit Ethernet mode of operation. • *slot*—Slot number of module depending on the chassis model. **Note** Enter the same *slot* module value that you used for the corresponding port while configuring the Breakout feature. • *port-range*—Single port or range of ports. **Note** Enter the same *port-range* value that you used for the corresponding port while configuring the Breakout feature.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the changes by copying the running configuration to the startup configuration.

Example

This example shows how to remove the breakout configuration in a port and return to the 40 Gigabit Ethernet mode of operation:

```
switch# configure terminal
switch(configure)# no interface breakout module 1 port 12 map 10g-4x
switch(configure)# copy running-config startup-config
```

This example shows how to remove the breakout configuration in a port and return to the 100 Gigabit Ethernet mode of operation:

```
switch# configure terminal
switch(configure)# no interface breakout module 1 port 8 map 25g-4x
switch(configure)# copy running-config startup-config
```

Verifying a Breakout Configuration

Use the following commands to verify a breakout configuration. You can use the commands in any order.

Procedure

Step 1 **show interface eth1/1 capabilities**

Displays information about the interface configuration.

Step 2 **show interface brief**

Displays a brief summary of the interface configuration.

Example

This example shows how to verify a breakout configuration for an interface:

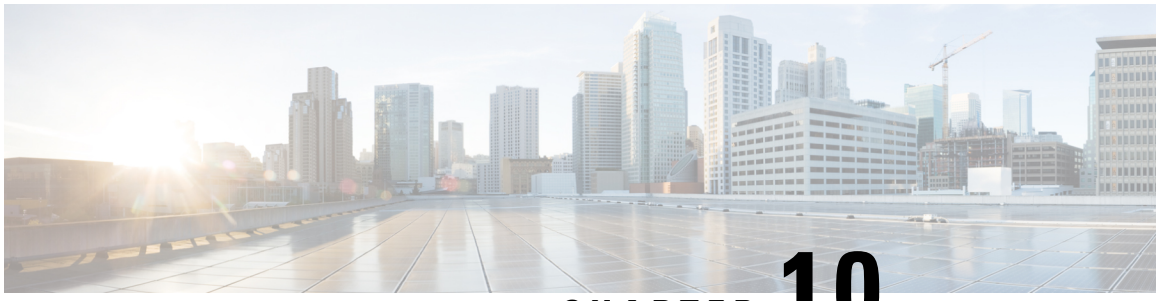
```
switch# show interface ethernet 1/1 capabilities | i Breakout

Breakout capable:      yes
```

This example shows how to display the summary of an interface configuration:

```
switch# show interface brief | grep 1/1

Eth1/1/1      --      eth  routed down    SFP not inserted           auto(D) --
Eth1/1/2      --      eth  routed down    SFP not inserted           auto(D) --
Eth1/1/3      --      eth  routed down    SFP not inserted           auto(D) --
Eth1/1/4      --      eth  routed down    SFP not inserted           auto(D) --
```



CHAPTER 10

Configuring IP Tunnels

This chapter describes how to configure IP tunnels.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Feature History for Configuring IP Tunnels

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Release	Feature Information
GRE tunnels	7.3(0)DX(1)	Support for M3 Series modules was added.
GRE tunnels	6.2(10)	Support for F3 Series modules was added.
Support for tunnel and its transport in different VRFs	6.1(1)	This feature was introduced.

Feature Name	Release	Feature Information
IP tunnels in VDC and VRF other than default	4.2(1)	This feature was introduced.
IP tunnels	4.0(1)	This feature was introduced.

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 see configure it. From Cisco NX-OS Release 4.2, the system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. See the Cisco Nexus 7000 Series NX-OS System Management Configuration Guide for information about rollbacks and checkpoints.

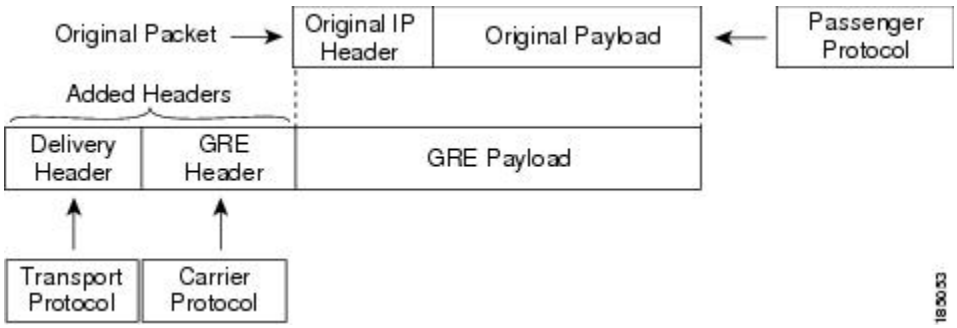
GRE Tunnels

Note From Cisco NX-OS Release 5.1(1), the software supports multicasting over GRE tunnels.

You can use generic routing encapsulation (GRE) as the carrier protocol for a variety of passenger protocols.

The figure below 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 29: GRE PDU



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. Cisco NX-OS software disables ICMP unreachable messages by default. ICMP unreachable messages can be enabled in the Cisco NX-OS software using the **ip unreachables** interface command.

Virtualization Support

From Cisco NX-OS Release 4.2, you can configure tunnels in a nondefault VDC and a nondefault VRF. A tunnel configured in one VDC is isolated from a tunnel with the same number configured in another VDC. For example, Tunnel 0 in VDC 1 is independent of tunnel 0 in VDC 2.

Before Cisco NX-OS Release 6.1(1), a tunnel interface and tunnel transport should be in the same VRF. See the Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide for information about VDCs and see the Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide for information about VRFs.

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 for IP Tunnels

IP tunnels have the following configuration guidelines and limitations:

- 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.
- Tunnels are supported only on the M Series cards on Cisco Nexus 7000 Series platforms.
- Cisco NX-OS does not support the Web Cache Control Protocol (WCCP) on tunnel interfaces.
- Tunnel features are supported only on M series and F3 series modules on Cisco Nexus 7000 Series and Cisco Nexus 7700 Series platforms.
- Cisco NX-OS does not support GRE tunnel keepalives.
- When the tunnelled (encapsulated) traffic is forwarded to the same interface from where the traffic was originally received (unencapsulated), make ensure that the IP redirects are disabled using the **no ip redirects** command.
- IPv6 as a carrier or a passenger/transport protocol is not supported in GRE Tunnels.

Default Settings for IP Tunnels

Table 52: Default Settings for IP Tunnels

Parameter	Default
Path MTU discovery age timer	10 minutes
Path MTU discovery minimum MTU	64
Tunnel feature	Disabled

Configuring IP Tunnels

Enabling Tunneling

Before you begin

You must enable the tunneling feature before you can configure any IP tunnels.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature tunnel**	Allows the creation of a new tunnel interface. To disable the tunnel interface feature, use the **no** form of this command.
Step 3	(Optional) switch(config)# **show feature**	Displays information about the features enabled on the device.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Creating a Tunnel Interface

Before you begin

- From Cisco NX-OS Release 6.1 and later releases, you can configure the tunnel source and the tunnel destination in different VRFs. Ensure that you have enabled the tunneling feature.
- You can create a tunnel interface and then configure this logical interface for your IP tunnel.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)#**interface tunnel** *number*	Creates a new tunnel interface.
Step 3	switch(config-if)# **tunnel source** {*ip-address* | *interface-name*}	Configures the source address for this IP tunnel.
Step 4	switch(config-if)# **tunnel destination** {*ip-address* | *host-name*}	Configures the destination address for this IP tunnel.

	Command or Action	Purpose
Step 5	switch(config-if)# **tunnel use-vrf** *vrf-name*	Uses the configured VRF to look up the tunnel IP destination address.
Step 6	(Optional) switch(config-if)# **show interfaces tunnel** *number*	Displays the tunnel interface statistics.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Use the **no interface tunnel** command to remove the tunnel interface and all associated configuration.

Table 53: Removing the Tunnel Interface and its Associated Configuration

Command	Purpose
no interface tunnel *number*	Deletes the tunnel interface and the associated configuration.

You can configure the following optional parameters to tune the tunnel in interface configuration mode:

Table 54: Configuring Optional Parameters

Command	Purpose
description *string*	Configures a description for the tunnel.
mtu *value*	Sets the MTU of IP packets sent on an interface.
tunnel ttl *value*	Sets the tunnel time-to-live value. The range is from 1 to 255.

Example

This example shows how to create a tunnel interface:

```
switch# configure terminal
switch(config)# interface tunnel 1
switch(config-if)# tunnel source ethernet 1/2
switch(config-if)# tunnel destination 192.0.2.1
switch(config-if)# copy running-config startup-config
```

Configuring a GRE Tunnel

You can set a tunnel interface to GRE tunnel mode.

Before you begin

Ensure that you have enabled the tunneling feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface tunnel** *number*	Creates a new tunnel interface.
Step 3	switch(config-if)# **tunnel mode gre ip**	Sets this tunnel mode to GRE.
Step 4	(Optional) switch(config-if)# **show interfaces tunnel** *number*	Displays the tunnel interface statistics.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Enabling Path MTU Discovery

Use the **tunnel path-mtu discovery** command to enable path MTU discovery on a tunnel.

Command	Purpose
tunnel path-mtu-discovery [**age-timer** *min*] [**min-mtu** *bytes*]	Enables Path MTU Discovery (PMTUD) on a tunnel interface. The parameters are as follows: • *mins*—Number of minutes. The range is from 10 to 30. The default is 10. • *mtu-bytes*—Minimum MTU recognized. The range is from 92 to 65535. The default is 92.

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.
- Before you configure this feature for the entire system, ensure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Assign the IP address for a tunnel interface after you have configured the interface for a VRF.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface tunnel** *number*	Enters interface configuration mode.

	Command or Action	Purpose
Step 3	switch(config-if)# **vrf member** *vrf-name*	Adds this interface to a VRF.
Step 4	switch(config-vrf)# **ip address** *ip-prefix/length*	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 5	(Optional) switch(config-vrf)# **show vrf** [*vrf-name*] **interface** *interface-type number*	Displays VRF information.
Step 6	(Optional) switch(config-vrf)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

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
```

Configuration Examples for IP Tunneling

These examples show 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 ethernet1/2
  ip address 192.0.2.55/8
```

Router B:

```
feature tunnel
interface tunnel 0
  ip address 209.165.20.1/8
  tunnel source ethernet2/1
  tunnel destination 192.0.2.55
  tunnel mode gre ip
interface ethernet 2/1
ip address 192.0.2.2/8
```

Verifying the IP Tunnel Configuration

Use one of the following commands to verify IP tunnel configuration information:

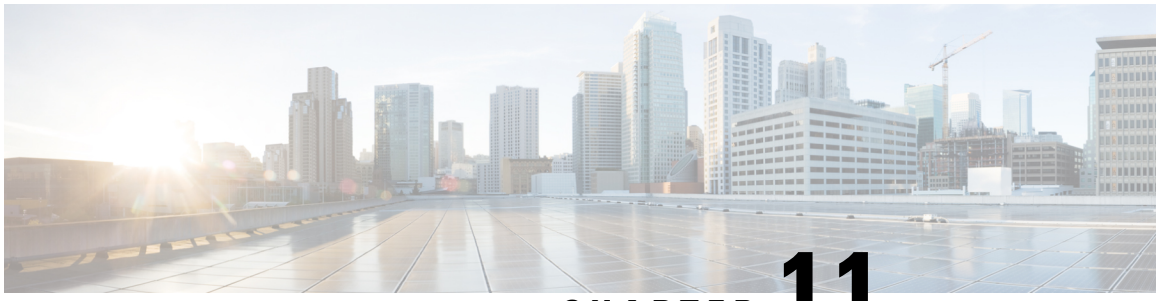
Table 55: Verifying the IP Tunnel Configuration

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 brief | include Tunnel	Displays the operational status, IP address, encapsulation type, and MTU of the tunnel interface.
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.

Related Documents

Table 56: Related Documents

Related Topic
Cisco Nexus 7000 Series NX-OS Interfaces Command Reference
Cisco Nexus 7000 Series NX-OS System Management Configuration Guide
Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide
Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 7000 Series Switches, Release 6.x
Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide
Cisco NX-OS Licensing Guide
VLANs, MAC address tables, private VLANs, and the Spanning Tree Protocol. Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide
Cisco Nexus 7000 Series NX-OS FabricPath Command Reference
Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide
Cisco Nexus 7000 Series NX-OS Release Notes



CHAPTER 11

Configuring Q-in-Q VLAN Tunnels

This chapter describes how to configure Q-in-Q VLAN tunnels.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Feature History for Q-in-Q Tunnels and Layer 2 Protocol Tunneling

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 57: Feature History for Q-in-Q Tunnels and Layer 2 Protocol tunneling

Feature Name	Release	Feature Information
Display policy errors on interfaces and VLANs	6.2(2)	Added the **show interface status error policy** command.
Q-in-Q VLAN Tunnels	5.0(2)	This feature was introduced.
L2 Protocol tunneling	5.0(2)	This feature was introduced.

Information About Q-in-Q Tunnels

This chapter describes how to configure IEEE 802.1Q-in-Q (Q-in-Q) VLAN tunnels and Layer 2 protocol tunneling on Cisco NX-OS devices.

A Q-in-Q VLAN tunnel enables a service provider to segregate the traffic of different customers in their infrastructure, while still giving the customer a full range of VLANs for their internal use by adding a second 802.1Q tag to an already tagged frame.

Q-in-Q Tunneling

Business customers of service providers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN ranges required by different customers in the same service-provider network might overlap, and the traffic of customers through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the VLAN limit of 4096 of the 802.1Q specification.

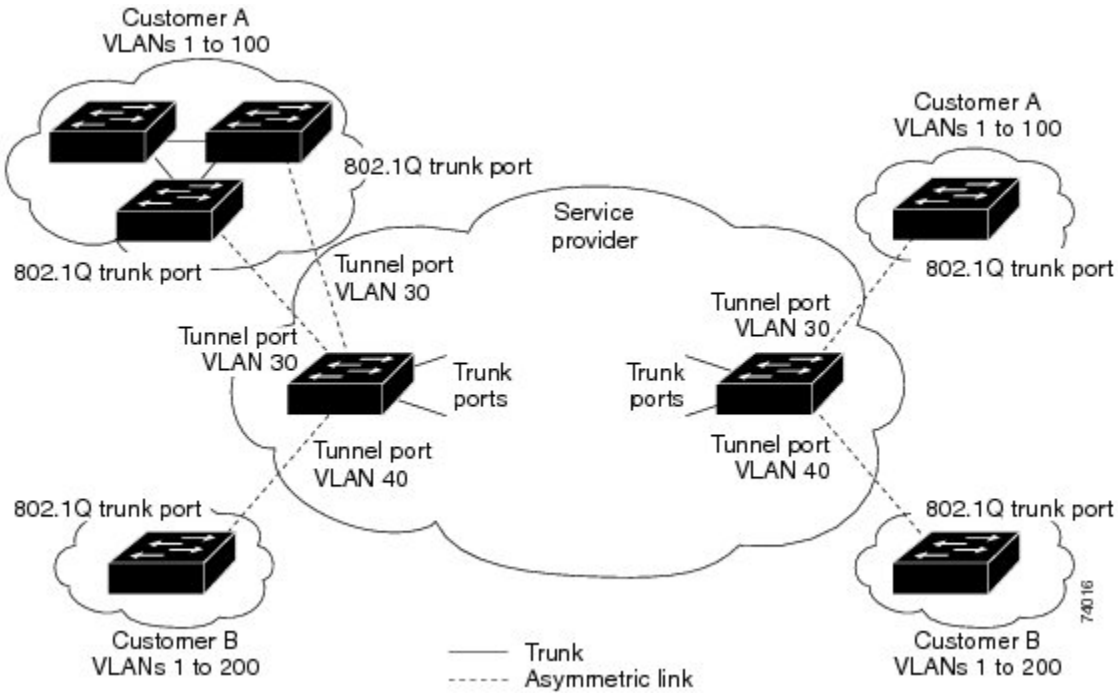
Note Q-in-Q is supported on port channels and virtual port channels (vPCs). To configure a port channel as an asymmetrical link, all ports in the port channel must have the same tunneling configuration.

Using the 802.1Q tunneling feature, service providers can use a single VLAN to support customers who have multiple VLANs. Customer VLAN IDs are preserved and the traffic from different customers is segregated within the service-provider infrastructure even when they appear to be on the same VLAN. The 802.1Q tunneling expands the VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets. A port configured to support 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN that is dedicated to tunneling. Each customer requires a separate VLAN, but that VLAN supports all of the customer's VLANs.

Customer traffic that is tagged in the normal way with appropriate VLAN IDs come from an 802.1Q trunk port on the customer device and into a tunnel port on the service-provider edge switch. The link between the customer device and the edge switch is an asymmetric link because one end is configured as an 802.1Q trunk port and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID that is unique to each customer. See the figure below.

Note Selective Q-in-Q tunneling is not supported. All frames that enter the tunnel port are subject to Q-in-Q tagging.

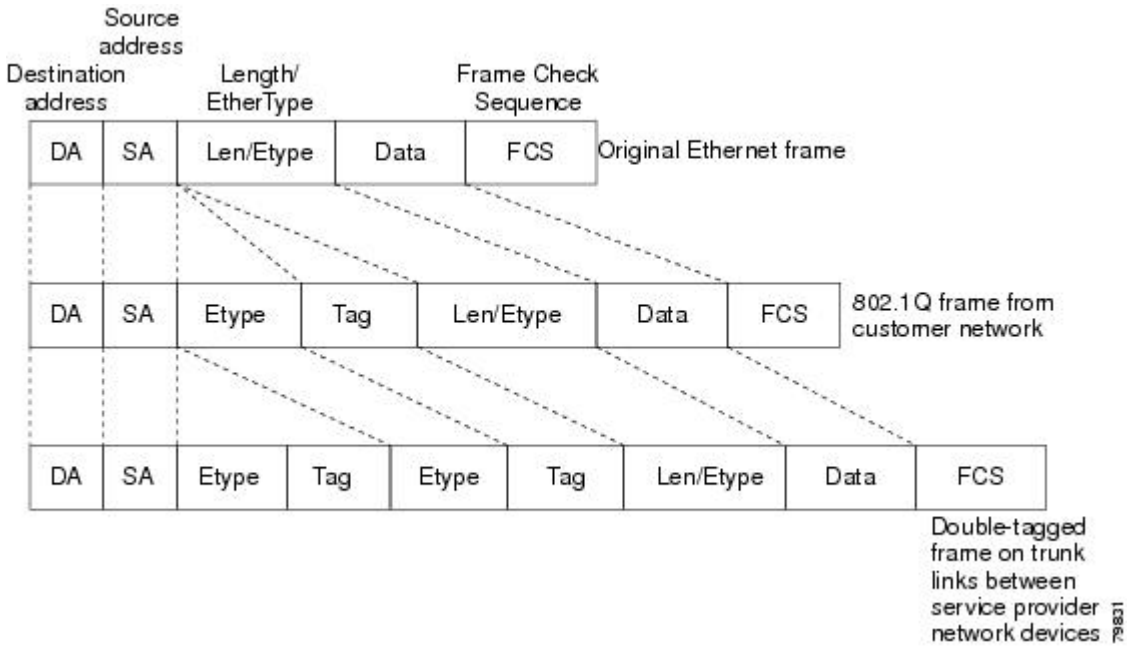
Figure 30: 802.1Q-in-Q Tunnel Ports



Packets that enter the tunnel port on the service-provider edge switch, which are already 802.1Q-tagged with the appropriate VLAN IDs, are encapsulated with another layer of an 802.1Q tag that contains a VLAN ID that is unique to the customer. The original 802.1Q tag from the customer is preserved in the encapsulated packet. Therefore, packets that enter the service-provider infrastructure are double-tagged.

The outer tag contains the customer's access VLAN ID (as assigned by the service provider), and the inner VLAN ID is the VLAN of the incoming traffic (as assigned by the customer). This double tagging is called tag stacking, Double-Q, or Q-in-Q as shown in the figure below.

Figure 31: Untagged, 802.1Q-Tagged, and Double-Tagged Ethernet Frames



By using this method, the VLAN ID space of the outer tag is independent of the VLAN ID space of the inner tag. A single outer VLAN ID can represent the entire VLAN ID space for an individual customer. This technique allows the customer's Layer 2 network to extend across the service provider network, potentially creating a virtual LAN infrastructure over multiple sites.

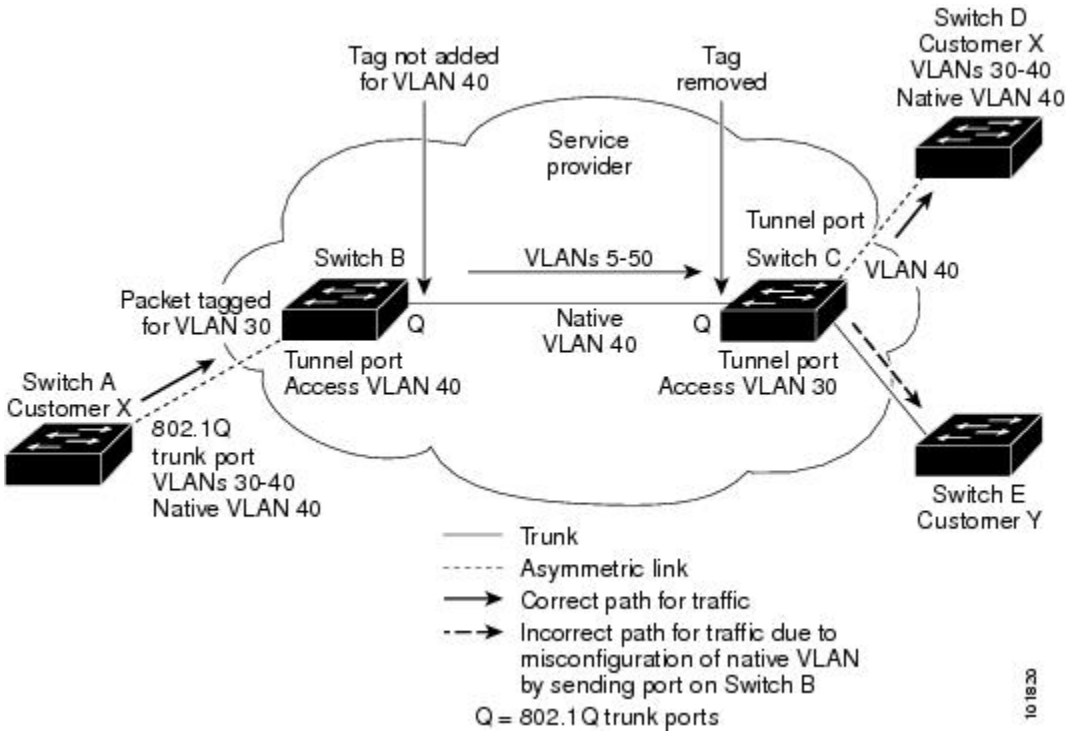
Note Hierarchical tagging, or multi-level dot1q tagging Q-in-Q, is not supported.

Native VLAN Hazard

When configuring 802.1Q tunneling on an edge switch, you must use 802.1Q trunk ports for sending out packets into the service-provider network. However, packets that go through the core of the service-provider network might be carried through 802.1Q trunks, ISL trunks, or nontrunking links. When 802.1Q trunks are used in these core switches, the native VLANs of the 802.1Q trunks must not match any native VLAN of the dot1q-tunnel port on the same switch because traffic on the native VLAN is not tagged on the 802.1Q transmitting trunk port.

In the figure below, VLAN 40 is configured as the native VLAN for the 802.1Q trunk port from Customer X at the ingress edge switch in the service-provider network (Switch B). Switch A of Customer X sends a tagged packet on VLAN 30 to the ingress tunnel port of Switch B in the service-provider network that belongs to access VLAN 40. Because the access VLAN of the tunnel port (VLAN 40) is the same as the native VLAN of the edge-switch trunk port (VLAN 40), the 802.1Q tag is not added to tagged packets that are received from the tunnel port. The packet carries only the VLAN 30 tag through the service-provider network to the trunk port of the egress-edge switch (Switch C) and is misdirected through the egress switch tunnel port to Customer Y.

Figure 32: Native VLAN Hazard



These are a couple ways to solve the native VLAN problem:

- Configure the edge switch so that all packets going out an 802.1Q trunk, including the native VLAN, are tagged by using the vlan dot1q tag native command. If the switch is configured to tag native VLAN packets on all 802.1Q trunks, the switch accepts untagged packets but sends only tagged packets.

Note The **vlan dot1q tag native** command is a global command that affects the tagging behavior on all trunk ports.

- Ensure that the native VLAN ID on the edge switch trunk port is not within the customer VLAN range. For example, if the trunk port carries traffic of VLANs 100 to 200, assign the native VLAN a number outside that range.

Information About Layer 2 Protocol Tunneling

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. The Spanning Tree Protocol (STP) must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure. The Cisco Discovery Protocol (CDP) must be able to discover neighboring Cisco devices from local and remote sites, and the VLAN Trunking Protocol (VTP) must provide consistent VLAN configuration throughout all sites in the customer network.

When protocol tunneling is enabled, edge switches on the inbound side of the service-provider infrastructure encapsulate Layer 2 protocol packets with a special MAC address and send them across the service-provider

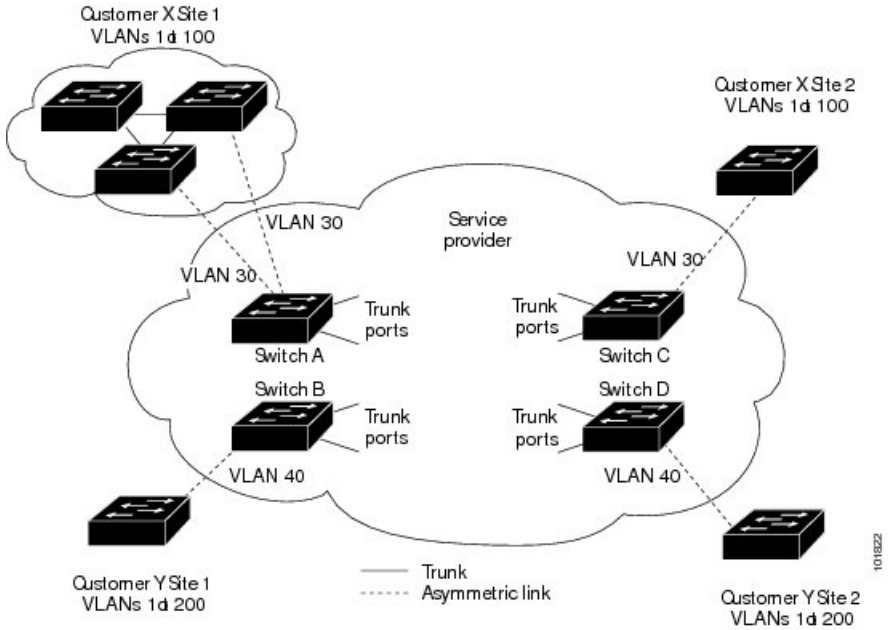
network. Core switches in the network do not process these packets, but forward them as normal packets. Bridge protocol data units (BPDUs) for CDP, STP, or VTP cross the service-provider infrastructure and are delivered to customer switches on the outbound side of the service-provider network. Identical packets are received by all customer ports on the same VLANs.

If protocol tunneling is not enabled on 802.1Q tunneling ports, remote switches at the receiving end of the service-provider network do not receive the BPDUs and cannot properly run STP, CDP, 802.1X, and VTP. When protocol tunneling is enabled, Layer 2 protocols within each customer's network are totally separate from those running within the service-provider network. Customer switches on different sites that send traffic through the service- provider network with 802.1Q tunneling achieve complete knowledge of the customer's VLAN.

Note Layer 2 protocol tunneling works by tunneling BPDUs in the software. A large number of BPDUs that come into the supervisor will cause the CPU load to go up. You might need to make use of hardware rate limiters to reduce the load on the supervisor CPU. See the "Configuring the Rate Limit for Layer 2 Protocol Tunnel Ports" section.

For example, in the figure below, Customer X has four switches in the same VLAN that are connected through the service-provider network. If the network does not tunnel BPDUs, switches on the far ends of the network cannot properly run the STP, CDP, 802.1X, and VTP protocols.

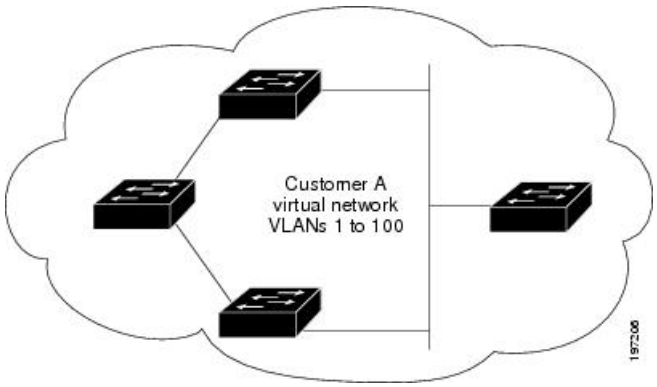
Figure 33: Layer 2 Protocol Tunneling



In the preceding example, STP for a VLAN on a switch in Customer X, Site 1 will build a spanning tree on the switches at that site without considering convergence parameters based on Customer X's switch in Site 2.

The figure below shows the resulting topology on the customer's network when BPDU tunneling is not enabled.

Figure 34: Virtual Network Topology Without BPDU Tunneling



Guidelines and Limitations for Q-in-Q Tunnels

Q-in-Q tunnels and Layer 2 tunneling have the following configuration guidelines and limitations:

- Q-in-Q tunnels are not supported on F1 linecards.
- Switches in the service-provider network must be configured to handle the increase in MTU size due to Q-in-Q tagging.
- MAC address learning for Q-in-Q tagged packets is based on the outer VLAN (Service Provider VLAN) tag. Packet forwarding issues might occur in deployments where a single MAC address is used across multiple inner (customer) VLANs.
- Layer 3 and higher parameters cannot be identified in tunnel traffic (for example, Layer 3 destination and source addresses). Tunneled traffic cannot be routed.
- Cisco Nexus 7000 Series devices can provide only MAC-layer ACL/QoS for tunnel traffic (VLAN IDs and src/dest MAC addresses).
- You should use MAC address-based frame distribution.
- Asymmetrical links do not support the Dynamic Trunking Protocol (DTP) because only one port on the link is a trunk. You must configure the 802.1Q trunk port on an asymmetrical link to trunk unconditionally.
- You cannot configure the 802.1Q tunneling feature on ports that are configured to support private VLANs. Private VLAN are not required in these deployments.
- You must disable IGMP snooping on the tunnel VLANs.
- Control Plane Policing (CoPP) is not supported.
- You should enter the vlan dot1Q tag native command to maintain the tagging on the native VLAN and drop untagged traffic. This command prevents native VLAN misconfigurations.
- You must manually configure the 802.1Q interfaces to be edge ports.
- Dot1x tunneling is not supported.
- You should perform an EPLD upgrade to newer versions in order for EtherType configuration to take effect on some Cisco Nexus devices.

- You cannot configure Layer 2 protocol features to forward STP BPDU or CDP packets across the tunnel.

Configuring Q-in-Q Tunnels and Layer 2 Protocol Tunneling

Creating a 802.1Q Tunnel Port

You create the dot1q-tunnel port using the **switchport mode** command.

Note You must set the 802.1Q tunnel port to an edge port with the **spanning-tree port type edge** command. The VLAN membership of the port is changed using the **switchport access vlan** *vlan-id* command.

You should disable IGMP snooping on the access VLAN allocated for the dot1q-tunnel port to allow multicast packets to traverse the Q-in-Q tunnel.

Before you begin

You must first configure the interface as a switchport.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport**	Sets the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **switchport mode dot1q-tunnel**	Creates a 802.1Q tunnel on the port. The port will go down and reinitialize (port flap) when the interface mode is changed. BPDU filtering is enabled and CDP is disabled on tunnel interfaces.
Step 5	(Optional) switch(config-if)# **no switchport mode dot1q-tunnel**	Disables the 802.1Q tunnel on the port.
Step 6	switch(config-if)# **exit**	Exits configuration mode.
Step 7	(Optional) switch(config)# **show dot1q-tunnel** [**interface** *if-range*]	Displays all ports that are in dot1q-tunnel mode. Optionally, you can specify an interface or range of interfaces to display.
Step 8	(Optional) switch(config)# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce an error during policy programming. This ensures that policies are consistent with hardware policies.

	Command or Action	Purpose
		Use the **detail** command to display the details of the interfaces that produce an error.
Step 9	(Optional) switch(config)# **no shutdown**	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 10	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# exit
switch(config)# exit
switch# show dot1q-tunnel
```

Changing the EtherType for Q-in-Q

You can change the 802.1Q EtherType value to be used for Q-in-Q encapsulation.

Note You must set the EtherType only on the egress trunk interface that carries double tagged frames (the trunk interface that connects the service providers). If you change the EtherType on one side of the trunk, you must set the same value on the other end of the trunk (symmetrical configuration).

Caution The EtherType value you set affect all the tagged packets that go out on the interface (not just Q-in-Q packets).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport**	Sets the interface as a Layer 2 switching port.

	Command or Action	Purpose
Step 4	switch(config-if)# **switchport dot1q ethertype** *value*	Sets the EtherType for the Q-in-Q tunnel on the port.
Step 5	(Optional) switch(config-if)# **no switchport dot1q ethertype**	Resets the EtherType on the port to the default value of 0x8100.
Step 6	switch(config-if)# **exit**	Exits configuration mode.
Step 7	(Optional) switch(config)# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce an error during policy programming. This ensures that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 8	(Optional) switch(config)# **no shutdown**	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	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport dot1q ethertype 0x9100
switch(config-if)# exit
switch(config)# exit
switch# show dot1q-tunnel
```

Enabling the Layer 2 Protocol Tunnel

You can enable protocol tunneling on the 802.1Q tunnel port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.

	Command or Action	Purpose
Step 3	switch(config-if)# **switchport**	Sets the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **switchport mode dot1q-tunnel**	Creates a 802.1Q tunnel on the port. The port will go down and reinitialize (port flap) when the interface mode is changed. BPDU filtering is enabled and CDP is disabled on tunnel interfaces.
Step 5	switch(config-if)# **l2protocol tunnel** [**cdp** | **stp** | **vtp**]	Enables Layer 2 protocol tunneling. Optionally, you can enable CDP, STP, or VTP tunneling.
Step 6	(Optional) switch(config-if)# **no l2protocol tunnel** [**cdp** | **stp** | **vtp**]	Disables protocol tunneling.
Step 7	switch(config-if)# **exit**	Exits configuration mode.
Step 8	(Optional) switch(config)# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce an error during policy programming. This ensures that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 9	(Optional) switch(config)# **no shutdown**	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 10	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable protocol tunneling on an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# l2protocol tunnel stp
switch(config-if)# exit
switch(config)# exit
```

Configuring Global CoS for L2 Protocol Tunnel Ports

You can specify a Class of Service (CoS) value globally so that ingress BPDUs on the tunnel ports are encapsulated with the specified class.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **l2protocol tunnel cos** *value*	Specifies a global CoS value on all Layer 2 protocol tunneling ports. The default cos-value is 5.
Step 3	(Optional) switch(config)# **no l2protocol tunnel cos**	Sets the global CoS value to default.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	(Optional) switch# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce an error during policy programming. This ensures that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 6	(Optional) switch# **no shutdown**	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	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to specify a global CoS value for the purpose of Layer 2 protocol tunneling:

```
switch# configure terminal
switch(config)# l2protocol tunnel cos 6
switch(config)# exit
```

Configuring the Rate Limit for Layer 2 Protocol Tunnel Ports

You can specify the hardware rate limiter configuration for Layer 2 protocol tunneling. The default is set to 500 packets per second. Depending on the load or the number of VLANs to be tunneled for a customer, you might need to adjust this value to prevent STP errors on the customer's network.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware rate-limiter layer-2 l2pt** *packets-per-sec*	Sets the threshold in packets per second above which incoming protocol packets from dot1q-tunnel ports are dropped in hardware. Valid values are from 0 to 30000.
Step 3	(Optional) switch(config)# **no hardware rate-limiter layer-2 l2pt**	Resets the threshold values to the default of 500 packets per second.

Configuring Thresholds for Layer 2 Protocol Tunnel Ports

You can specify the port drop and shutdown value for a Layer 2 protocol tunneling port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport**	Sets the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **switchport mode dot1q-tunnel**	Creates a 802.1Q tunnel on the port.
Step 5	switch(config-if)# **l2protocol tunnel drop-threshold** [**cdp** | **stp** | **vtp**] *packets-per-sec*	Specifies the maximum number of packets that can be processed on an interface before being dropped. Optionally, you can specify CDP, STP, or VTP. Valid values for the packets are from 1 to 4096.
Step 6	(Optional) switch(config-if)# **no l2protocol tunnel drop-threshold** [**cdp** | **stp** | **vtp**]	Resets the threshold values to 0 and disables the drop threshold.
Step 7	switch(config-if)# **l2protocol tunnel shutdown-threshold** [**cdp** | **stp** | **vtp**] *packets-per-sec*	Specifies the maximum number of packets that can be processed on an interface. When the number of packets is exceeded, the port is put in error-disabled state. Optionally, you can specify CDP, STP, or VTP. Valid values for the packets is from 1 to 4096.
Step 8	(Optional) switch(config-if)# **l2protocol tunnel shutdown-threshold** [**cdp** | **stp** | **vtp**]	Resets the threshold values to 0 and disables the shutdown threshold.
Step 9	switch(config-if)# **exit**	Exits configuration mode.

	Command or Action	Purpose
Step 10	(Optional) switch(config)# **show interface status error policy** [**detail**]	Displays the interfaces and VLANs that produce an error during policy programming. This ensures that policies are consistent with hardware policies. Use the **detail** command to display the details of the interfaces that produce an error.
Step 11	(Optional) switch(config)# **no shutdown**	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 12	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Verifying the Q-in-Q Configuration

Use the following commands to verify the Q-in-Q configuration:

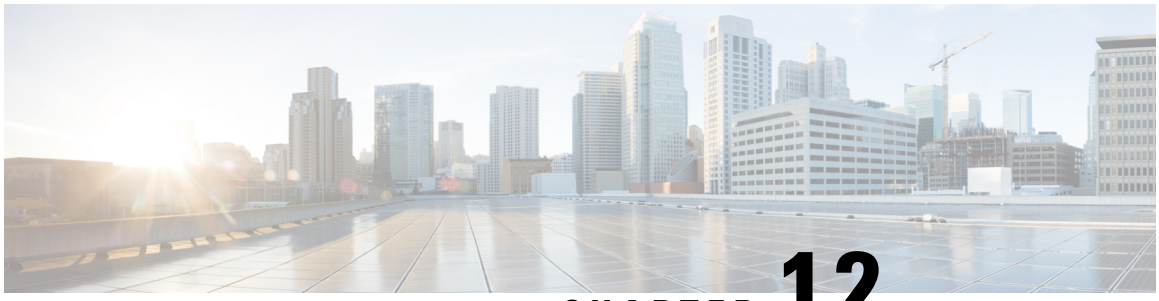
Table 58: Verifying the Q-in-Q Configuration

Command	Purpose
clear l2protocol tunnel counters [**interface** *if-range*]	Clears all the statistics counters. If no interfaces are specified, the Layer 2 protocol tunnel statistics are cleared for all interfaces.
show dot1q-tunnel [**interface** *if-range*]	Displays a range of interfaces or all interfaces that are in dot1q-tunnel mode.
show l2protocol tunnel [**interface** *if-range* | **vlan** *vlan-id*]	Displays Layer 2 protocol tunnel information for a range of interfaces, for all dot1q-tunnel interfaces that are part of a specified VLAN or all interfaces.
show l2protocol tunnel summary	Displays a summary of all ports that have Layer 2 protocol tunnel configurations.
show running-config l2pt	Displays the current Layer 2 protocol tunnel running configuration.
show interface status error policy [**detail**]	Displays errors on interfaces and VLANs that are inconsistent with hardware policies. The **detail** command displays the details of the interfaces and VLANs that receive an error.

Configuration Examples for Q-in-Q and Layer 2 Protocol Tunneling

This example shows a service provider switch that is configured to process Q-in-Q for traffic coming in on Ethernet 7/1. A Layer 2 protocol tunnel is enabled for STP BPDUs. The customer is allocated VLAN 10 (outer VLAN tag).

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vlan 10
switch(config-vlan)# no shutdown
switch(config-vlan)# no ip igmp snooping
switch(config-vlan)# exit
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree port type edge
switch(config-if)# l2protocol tunnel stp
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# exit
switch#
```



CHAPTER 12

Configuring Ethernet OAM

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Feature History for Ethernet OAM

Table 59: Feature History for Ethernet OAM

Feature Name	Release	Feature Information
Ethernet OAM	8.2(3)	Ethernet OAM is enhanced for the following: • Frame error threshold values can be configured on the Ethernet link. This helps to measure the quality of the link. • The **link-oam-dying-gasp** and the **link-oam-discovery-timeout** options are supported under the **errdisable recovery cause** command to recover the Ethernet link OAM.

Feature Name	Release	Feature Information
Ethernet OAM	7.3(0)D1(1)	This feature was introduced.

Information About Ethernet OAM

Ethernet as a Metro Area Network (MAN) or a Wide Area Network (WAN) technology benefits greatly from the implementation of Operations, Administration and Maintenance (OAM) features. Ethernet link OAM features allow Service Providers to monitor the quality of the connections on a MAN or WAN. Service providers can monitor specific events, take actions on events, and if necessary, put specific interfaces into loopback mode for troubleshooting. Ethernet link OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link.

Ethernet link OAM can be configured in the following ways:

- A Link OAM profile can be configured, and this profile can be used to set the parameters for multiple interfaces.
- Link OAM can be configured directly on an interface.

When an interface is also using a link OAM profile, specific parameters that are set in the profile can be overridden by configuring a different value directly on the interface.

An EOAM profile simplifies the process of configuring EOAM features on multiple interfaces. An Ethernet OAM profile, and all of its features, can be referenced by other interfaces, allowing other interfaces to inherit the features of that Ethernet OAM profile.

Individual Ethernet link OAM features can be configured on individual interfaces without being part of a profile. In these cases, the individually configured features always override the features in the profile.

The preferred method of configuring custom EOAM settings is to create an EOAM profile in Ethernet configuration mode and then attach it to an individual interface or to multiple interfaces.

The following standard Ethernet Link OAM features are supported on the Cisco Nexus 7000 Series switch:

- Neighbor Discovery
- Link Monitoring
- Miswiring Detection (Cisco-Proprietary)

Neighbor Discovery

Neighbor discovery enables each end of a link to learn the OAM capabilities of the other end and establish an OAM peer relationship. Each end also can require that the peer have certain capabilities before it will establish a session. You can configure certain actions to be taken if there is a capabilities conflict or if a discovery process times out, using the **capabilities-conflict** or **discovery-timeout** commands in the action configuration submode.

Link Monitoring

Link monitoring enables an OAM peer to monitor faults that cause the quality of a link to deteriorate over time. When link monitoring is enabled, an OAM peer can be configured to take action when the configured thresholds are exceeded.

Miswiring Detection (Cisco-Proprietary)

Miswiring Detection is a Cisco-proprietary feature that uses the 32-bit vendor field in every Information OAMPDU to identify potential miswiring cases.

Prerequisites for Ethernet OAM

- You must be in a user group associated with a task group that includes proper task IDs. The command reference guides include the task IDs required for each command. If you suspect that user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Guidelines and Limitations for Ethernet OAM

Cisco NX-OS Release 8.2(3) has the following Ethernet OAM enhancements:

- Frame error threshold values can be configured on the Ethernet link to measure the quality of the link.
- Error-disabled ports need to be manually shut/no shut to bring the port up.

  The error-disabled ports need to be manually shut/no-shut to bring the ports to the up state. This is applicable to all the Ethernet OAM links except for **link-oam-dying-gasp** and for **link-oam-discovery-timeout** if the **errdisable recovery cause** is configured with the **link-oam-dying-gasp** and **link-oam-discovery-timeout** options.

  This recovery mechanism for **link-oam-dying-gasp** and **link-oam-discovery-timeout** links is introduced in Cisco NX-OS Release 8.2(3).

The following modules are supported from Cisco Nexus Release 7.3(0)D1(1):

- M2-Series 10-Gigabit Ethernet Series Module for Cisco Nexus 7000 Series Switches.
- F3-Series 10-Gigabit Ethernet Series Module for Cisco Nexus 7000 Series Switches and Cisco Nexus 7700 Switches.

Ethernet OAM is not supported on the F2 series modules.

The following functional areas of Ethernet OAM are not supported on the Cisco Nexus 7000 Series switches:

- Remote loopback
- Ethernet Fault Detection (EFD)

Configuring Ethernet OAM

Configuring an Ethernet OAM Profile

Procedure

	Command or Action	Purpose

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

Step 2	**feature ethernet-link-oam** **Example:** `switch(config)# feature ethernet-link-oam`	Enables the Ethernet Link OAM feature.

Step 3	**ethernet oam profile** *profile-name*
	Example:
	`switch(config)# ethernet oam profile Profile_1`
	`switch(config-eoam)#`
		Creates a new Ethernet Operations, Administration and Maintenance (OAM) profile and enters Ethernet OAM configuration mode.

Step 4	**hello-interval** *interval-time*
	Example:
	`switch(config-eoam)# hello-interval 100ms`
	`switch(config-eoam-lm)#`
		Configures hello interval limit for the Ethernet OAM profile. The valid value for hello interval is 100 millisecond. The default value is 1 second.

Step 5	**link-monitor**
	Example:
	`switch(config-eoam)# link-monitor`
	`switch(config-eoam-lm)#`
		Enters the Ethernet OAM link monitor configuration mode.

Step 6	(Optional) **symbol-period window** *window* **Example:** `switch(config-eoam-lm)# symbol-period window 60000`	Configures the window size (in milliseconds) for an Ethernet OAM symbol-period error event. The range is 1000 to 60000. The default value is 1000.
Step 7	(Optional) **symbol-period threshold low** *threshold* [**high** *threshold*] **Example:** `switch(config-eoam-lm)# symbol-period threshold low 10000000 high 60000000`	Configures the thresholds (in symbols) that trigger an Ethernet OAM symbol-period error event. The high threshold is optional and is configurable only in conjunction with the low threshold.

	Command or Action	Purpose
		The range is 1 to 60000000. The default low threshold is 1.
Step 8	(Optional) **frame window** *window* **Example:** `switch(config-eoam-lm)# frame window 60`	Configures the frame window size (in milliseconds) of an OAM frame error event. The range is 1000 to 60000. The default value is 1000.
Step 9	(Optional) **frame threshold low** *threshold* **high** *threshold* **Example:** `switch(config-eoam-lm)# frame threshold low 10000000 high 60000000`	Configures the thresholds (in frames) that triggers an Ethernet OAM frame error event. The high threshold is optional and is configurable only in conjunction with the low threshold. The range is 1 to 12000000. The default low threshold is 1.
Step 10	(Optional) **frame-period window** *window* **Example:** `switch(config-eoam-lm)# frame-period window 60000`	Configures the window size (in milliseconds) for an Ethernet OAM frame-period error event. The range is 1000 to 60000. The default value is 1000.
Step 11	(Optional) **frame-period threshold low** *threshold* [**high** *threshold*] **Example:** `switch(config-eoam-lm)# frame-period threshold low 100 high 1000000`	Configures the thresholds (in frames) that trigger an Ethernet OAM frame-period error event. The high threshold is optional and is configurable only in conjunction with the low threshold. The range is 1 to 1000000. The default low threshold is 60000. The IEEE 802.3 standard defines threshold crossing events as number of error frames in a window. To comply with the standards, the low and high threshold for frame-period events is measured in errors per million frames. Hence, the calculation to determine the remote low and high threshold is (configured threshold * frame window in received Bridge Protocol Data Unit (BPDU))/1000000. For example, if the received frame window=300, then high threshold is 20000 * 300 / 1000000 = 6.
Step 12	(Optional) **frame-seconds window** *window* **Example:** `switch(config-eoam-lm)# frame-seconds window 900000`	Configures the window size (in milliseconds) for the OAM frame-seconds error event. The range is 10000 to 900000. The default value is 60000.
Step 13	(Optional) **frame-seconds threshold low** *threshold* [**high** *threshold*]	Configures the thresholds (in seconds) that trigger a frame-seconds error event. The high

	Command or Action	Purpose
	Example: `switch(config-eoam-lm)# frame-seconds threshold 3 threshold 900`	threshold value can be configured only in conjunction with the low threshold value. The range is 1 to 900. The default value is 1.

Step 14	Required: **exit**
	Example:
	`switch(config-eoam-lm)# exit`
	`switch(config-eoam)#`
		Exits to Ethernet OAM mode.

Step 15	Required: **connection timeout** *seconds* **Example:** `switch(config-eoam)# connection timeout 30`	Configures the timeout value (in seconds) for an Ethernet OAM session. The range is 2 to 30. The default value is 5.
Step 16	Required: **mode** {**active** | **passive**} **Example:** `switch(config-eoam)# mode passive`	Configures the Ethernet OAM mode. The default is active.

Step 17	Required: **require-remote**
	Example:
	`switch(config-eoam)# require-remote`
	`switch(config-eoam-require)#`
		Enters the require-remote configuration submode to specify the features that you have to enable before an Ethernet OAM session can become active.

Step 18	Required: **mode** {**active** | **passive**} **Example:** `switch(config-eoam-require)# mode active`	Requires that active mode or passive mode is configured on the remote end before the Ethernet OAM session becomes active.

Step 19	Required: **link-monitoring**
	Example:
	`switch(config-eoam-require)#`
	`link-monitoring`
		Requires that link-monitoring is configured on the remote end before the Ethernet OAM session becomes active.

Step 20	Required: **exit**
	Example:
	`switch(config-eoam-require)# exit`
	`switch(config-eoam)#`
		Exits the require-remote configuration submode.

Step 21	Required: **action**
	Example:
	`switch(config-eoam)# action`
	`switch(config-eoam-action)#`
		Enters the action configuration submode to configure event actions.

Step 22	Required: **capabilities-conflict** {**disable** | **efd** | **error-disable-interface**} **Example:**	Specifies the action that is taken on an interface when a capabilities-conflict event occurs. The default action is to create a syslog entry.

	Command or Action	Purpose
	`switch(config-eoam-action)# capabilities-conflict disable`	**Note** If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 23	Required: **critical-event** {**disable** | **error-disable-interface**} **Example:** `switch(config-eoam-action)# critical-event error-disable-interface`	Specifies the action that is taken on an interface when a critical-event notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 24	Required: **discovery-timeout** {**disable** | **efd** | **error-disable-interface**} **Example:** `switch(config-eoam-action)# discovery-timeout disable`	Specifies the action that is taken on an interface when a connection timeout occurs. The default action is to create a syslog entry. **Note** If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 25	Required: **dying-gasp** {**disable** | **error-disable-interface**} **Example:** `switch(config-eoam-action)# dying-gasp error-disable-interface`	Specifies the action that is taken on an interface when a dying-gasp notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 26	Required: **high-threshold** {**error-disable-interface** | **log**} **Example:** `switch(config-eoam-action)# high-threshold error-disable-interface`	Specifies the action that is taken on an interface when a high threshold is exceeded. The default is to take no action when a high threshold is exceeded. **Note** If you change the default, the **disable** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and take no action at the interface when the event occurs.

	Command or Action	Purpose
Step 27	Required: **remote-loopback disable** **Example:** `switch(config-eoam-action)# remote-loopback disable`	Specifies that no action is taken on an interface when a remote-loopback event occurs. The default action is to create a syslog entry. **Note** If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 28	Required: **session-down** {**disable** | **efd** | **error-disable-interface**} **Example:** `switch(config-eoam-action)# session-down error-disable-interface`	Specifies the action that is taken on an interface when an Ethernet OAM session goes down. **Note** If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 29	Required: **session-up disable** **Example:** `switch(config-eoam-action)# session-up disable`	Specifies that no action is taken on an interface when an Ethernet OAM session is established. The default action is to create a syslog entry. **Note** If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 30	Required: **uni-directional link-fault** {**disable** | **efd** | **error-disable-interface**} **Example:** `switch(config-eoam-action)# uni-directional link-fault disable`	Specifies the action that is taken on an interface when a link-fault notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 31	Required: **wiring-conflict** {**disable** | **efd** | **log**} **Example:** `switch(config-eoam-action)# wiring-conflict disable`	Specifies the action that is taken on an interface when a wiring-conflict event occurs. The default is to put the interface into error-disable state.

	Command or Action	Purpose
		Note If you change the default, the **error-disable-interface** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and put the interface into error-disable state when the event occurs.
Step 32	Required: **end** **Example:** `switch(config-eoam-action)# end`	Ends the configuration session and exits to the EXEC mode.

Attaching an Ethernet OAM Profile to an Interface

Procedure

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

Step 3	**ethernet oam**
	Example:
	`switch(config-if)# ethernet oam`
	`switch(config-if-eoam)#`
		Enables Ethernet OAM and enters Interface Ethernet OAM configuration mode.

Step 4	**profile** *profile-name* **Example:** `switch(config-if-eoam)# profile Profile_1`	Attaches the specified Ethernet OAM profile and all of it's configuration to the interface.
Step 5	Required: **end** **Example:** `switch(config-if-eoam)# end`	Ends the configuration session and exits to the EXEC mode.

Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration

Using an EOAM profile is an efficient way of configuring multiple interfaces with a common EOAM configuration. However, if you want to use a profile but also change the behavior of certain functions for a particular interface, then you can override the profile configuration. To override certain profile settings that are applied to an interface, you can configure that command in interface Ethernet OAM configuration mode to change the behavior for that interface.

In some cases, only certain keyword options are available in interface Ethernet OAM configuration due to the default settings for the command. For example, without any configuration of the **action** configuration submode commands, several forms of the command have a default behavior of creating a syslog entry when a profile is created and applied to an interface. Therefore, the **log** keyword is not available in Ethernet OAM configuration for these commands in the profile because it is the default behavior. However, the **log** keyword is available in Interface Ethernet OAM configuration if the default is changed in the profile configuration so you can retain the action of creating a syslog entry for a particular interface.

To configure Ethernet OAM settings at an interface and override the profile configuration, perform the following steps:

Procedure

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

Step 3	**ethernet oam**
	Example:
	`switch(config-if)# ethernet oam`
	`switch(config-if-eoam)#`
		Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.

Step 4	**hello-interval** *interval-time* **Example:** `switch(config-if-eoam)# hello-interval 1s`	Configures Ethernet OAM hello interval limit for an interface. The possible values for hello interval is 100 millisecond or 1 second. The default value is 1 second.
Step 5	*interface-Ethernet-OAM-command* **Example:** `switch(config-if-eoam)# mode passive`	Configures a setting for an Ethernet OAM configuration command and overrides the setting for the profile configuration, where *interface-Ethernet-OAM-command* is one of the supported commands on the platform in Interface Ethernet OAM configuration mode.

	Command or Action	Purpose
Step 6	Required: **end** **Example:** `switch(config-if-eoam)# end`	Ends the configuration session and exits to the EXEC mode.

Clearing Ethernet OAM Statistics on an Interface

Use the **clear ethernet oam statistics** command to clear the packet counters on all Ethernet OAM interfaces. Use the **clear ethernet oam statistics** *interface* command to clear the packet counters on a specific Ethernet OAM interface.

```
switch# clear ethernet oam statistics interface ethernet 1/3
```

Verifying the Ethernet OAM Configuration

Use the **show ethernet oam configuration** command to display the values for the Ethernet OAM configuration for a specific interface, or for all interfaces.

Note Some of these settings are not supported on certain platforms, but the defaults are still reported. On the Cisco Nexus 7000 Series switches, the following areas are unsupported:

- Remote loopback
- EFD

```
switch# show ethernet oam configuration interface ethernet 1/3
Ethernet3/1:
  Hello interval:                                        1s
  Link monitoring enabled:                                Y
  Remote loopback enabled:                                N
  Mib retrieval enabled:                                  N
  Uni-directional link-fault detection enabled:           N
  Configured mode:                                   Active
  Connection timeout:                                     5
  Symbol period window:                                1000
  Symbol period low threshold:                            1
  Symbol period high threshold:                        None
  Frame window:                                        1000
  Frame low threshold:                                    1
  Frame high threshold:                                None
  Frame period window:                                 1000
  Frame period low threshold:                             1
  Frame period high threshold:                         None
  Frame seconds window:                               60000
  Frame seconds low threshold:                            1
  Frame seconds high threshold:                        None
  High threshold action:                               None
  Link fault action:                                    Log
  Dying gasp action:                                    Log
  Critical event action:                                Log
  Discovery timeout action:                             Log
  Capabilities conflict action:                         Log
```

```
  Wiring conflict action:                          Error-Disable
  Session up action:                                         Log
  Session down action:                                       Log
  Remote loopback action:                                    Log
  Require remote mode:                                    Ignore
  Require remote MIB retrieval:                                N
  Require remote loopback support:                             N
  Require remote link monitoring:                              N
```

Use the **show ethernet oam discovery** command to display the status of the OAM sessions. If no interface is specified, details of all interfaces that have OAM configured will be displayed.

```
switch# show ethernet oam discovery ethernet 1/3
Ethernet3/1
Local client
  Administrative configuration:
    PDU revision:                         2
    Mode:                            Active
    Unidirectional support:               N
    Link monitor support:                 N
    Remote loopback support:              Y
    MIB retrieval support:                Y
    Maximum PDU size:                  1500
    Mis-wiring detection key:        20492C

  Operational status:
    Port status:                Operational
    Loopback status:                   None
    Interface mis-wired:                  N

Remote client
  MAC address:  0030.96fd.6bfa
  Vendor (OUI): 00.00.0C (Cisco)

  Administrative configuration:
    PDU revision:                         5
    Mode:                            Passive
    Unidirectional support:               N
    Link monitor support:                 Y
    Remote loopback support:              Y
    MIB retrieval support:                N
    Maximum PDU size:                  1500
```

Use the **show ethernet oam statistics** command to display statistics for local and remote OAM sessions. If no interface is specified, statistics of all interfaces that have OAM configured will be displayed.

```
switch# show ethernet oam statistics
Ethernet1/3
Counters
--------
  Information OAMPDU Tx            45
  Information OAMPDU Rx            42
  Unique Event Notification OAMPDU Tx         0
  Unique Event Notification OAMPDU Rx             0
  Duplicate Event Notification OAMPDU Tx          0
  Duplicate Event Notification OAMPDU Rx          0
  Loopback Control OAMPDU Tx                      0
  Loopback Control OAMPDU Rx                      3
  Variable Request OAMPDU Tx                      0
  Variable Request OAMPDU Rx                      0
  Variable Response OAMPDU Tx                     0
```

```
  Variable Response OAMPDU Rx                     0
  Organization Specific OAMPDU Tx                 0
  Organization Specific OAMPDU Rx                 0
  Unsupported OAMPDU Tx            93
  Unsupported OAMPDU Rx             0
  Frames Lost due to OAM                         12

Local event logs
----------------
  Errored Symbol Period records                   0
  Errored Frame records             0
  Errored Frame Period records           0
  Errored Frame Second records           0

Remote event logs
-----------------
  Errored Symbol Period records              0
  Errored Frame records             0
  Errored Frame Period records           0
  Errored Frame Second records                    0
```

Use the **show ethernet oam event-log** command to display the most recent event logs for interfaces on which OAM is configured.

```
switch# show ethernet oam event-log
Wed Jan 23 06:16:46.684 PST
Local Action Taken:
    N/A    - No action needed         EFD    - Interface brought down using EFD
    None   - No action taken          Err.D  - Interface error-disabled
    Logged - System logged

Ethernet3/1
================================================================================
Time                       Type           Loc'n  Action Threshold Breaching Value
-------------------------- -------------- ------ ------ --------- ---------------
Wed Jan 23 06:13:25 PST   Symbol period  Local  N/A            1               4
Wed Jan 23 06:13:33 PST   Frame          Local  N/A            1               6
Wed Jan 23 06:13:37 PST   Frame period   Local  None           9              12
Wed Jan 23 06:13:45 PST   Frame seconds  Local  N/A            1              10
Wed Jan 23 06:13:57 PST   Dying gasp     Remote Logged       N/A             N/A


Ethernet3/1
================================================================================
Time                       Type           Loc'n  Action Threshold Breaching Value
-------------------------- -------------- ------ ------ --------- ---------------
Wed Jan 23 06:26:14 PST   Dying gasp     Remote Logged       N/A             N/A
Wed Jan 23 06:33:25 PST   Symbol period  Local  N/A            1               4
Wed Jan 23 06:43:33 PST   Frame period   Remote N/A            9              12
Wed Jan 23 06:53:37 PST   Critical event Remote Logged       N/A             N/A
Wed Jan 23 07:13:45 PST   Link fault     Remote EFD          N/A             N/A
Wed Jan 23 07:18:23 PST   Dying gasp     Local  Logged       N/A             N/A
```

Use the **show ethernet oam event-log interface detail** command to display detailed event logs for specific interfaces on which OAM is configured.

```
switch# show ethernet oam event-log interface detail
Wed Jan 23 06:21:16.392 PST
(Scaled): For remote threshold events "Local High Threshold" is scaled for
          comparison with "Breaching Value".
          This is to account for different local and remote window sizes.
```

```
Ethernet3/1
============================================================================
Event at Wed Jan 23 2013 06:26:14.62 PST:
    Type:                                                  Dying gasp
    Location:                                                  Remote
    Local Action Taken:                                 System logged
    Local Event Running Total:                                     1
Event at Wed Jan 23 2013 06:33:25.62 PST:
    Type:                             Threshold Event - Symbol period
    Location:                                                   Local
    Local Action Taken:                               No action needed
    Local Event Running Total:                                     1
    Local Window Size:                                          1000
    Local Threshold:                                               1
    Local High Threshold:                             Not configured
    Breaching Value:                                               4
    Local Error Running Total:                                     8
Event at Wed Jan 23 2013 06:43:37.73 PST:
    Type:                              Threshold Event - Frame period
    Location:                                                  Remote
    Local Action Taken:                               No action needed
    Remote Event Running Total:                                    1
    Remote Window Size:                                         1000
    Remote Threshold:                                              9
    Local High Threshold (Scaled):                    Not configured
    Breaching Value:                                              12
    Remote Error Running Total:                                   24
Event at Wed Jan 23 2013 06:53:57.12 PST:
    Type:                                              Critical event
    Location:                                                  Remote
    Local Action Taken:                                 System logged
    Local Event Running Total:                                     1
Event at Wed Jan 23 2013 07:13:57.12 PST:
    Type:                                                  Link fault
    Location:                                                  Remote
    Local Action Taken:             Interface brought down using EFD
    Local Event Running Total:                                     1
Event at Wed Jan 23 2013 07:18:57.12 PST:
    Type:                                                  Dying gasp
    Location:                                                   Local
    Local Action Taken:                                 System logged
    Local Event Running Total:                                     1
```

Use the **show ethernet oam summary** to display a summary of all the active OAM sessions.

```
switch# show ethernet oam summary
Link OAM System Summary
=======================

Profiles                                6
Interfaces                             10
  Interface states:
    Port down                           1
    Passive wait                        1
    Active send                         1
    [Evaluating                         0]
    [Local accept                       0]
    [Local reject                       0]
    Remote reject                       1
    Operational                         6
      Loopback mode                     1
```

```
  Miswired connections                    1
Events                                   13
  Local                                   4
    Symbol error                          0
    Frame                                 2
    Frame period                          1
    Frame seconds                         1
  Remote                                  9
    Symbol error                          3
    Frame                                 4
    Frame period                          1
    Frame seconds                         1
```

Use the **show ethernet oam summary detail** command to display a summary of all the active OAM sessions and details about the 10 most recent events across all interfaces.

```
switch# show ethernet oam summary detail
Link OAM System Summary
=======================

Profiles                                  6
Interfaces                               10
  Interface states:
    Port down                             1
    Passive wait                          1
    Active send                           1
    [Evaluating                           0]
    [Local accept                         0]
    [Local reject                         0]
    Remote reject                         1
    Operational                           6
      Loopback mode                       1
  Miswired connections                    1
Events                                   13
  Local                                   4
    Symbol error                          0
    Frame                                 2
    Frame period                          1
    Frame seconds                         1
  Remote                                  9
    Symbol error                          3
    Frame                                 4
    Frame period                          1
    Frame seconds                         1
```

Configuration Examples for Ethernet OAM

Configuration Example for Configuring an Ethernet OAM Profile Globally

```
switch# configure terminal
switch(config)# feature ethernet-link-oam
switch(config)# ethernet oam profile Profile_1
switch(config-eoam)# hello-interval 100ms
switch(config-eoam)# link-monitor
switch(config-eoam-lm)# symbol-period window 60000
switch(config-eoam-lm)# symbol-period threshold low 10000000 high 60000000
switch(config-eoam-lm)# frame window 60
```

```
switch(config-eoam-lm)# frame threshold low 10000000 high 60000000
switch(config-eoam-lm)# frame-period window 60000
switch(config-eoam-lm)# frame-period threshold low 100 high 1000000
switch(config-eoam-lm)# frame-seconds window 900000
switch(config-eoam-lm)# frame-seconds threshold 3 threshold 900
switch(config-eoam-lm)# exit
switch(config-eoam)# connection timeout 30
switch(config-eoam)# mode passive
switch(config-eoam)# require-remote
switch(config-eoam-require)# mode active
switch(config-eoam-require)# link-monitoring
switch(config-eoam-require)# exit
switch(config-eoam)# action
switch(config-eoam-action)# capabilities-conflict disable
switch(config-eoam-action)# critical-event error-disable-interface
switch(config-eoam-action)# discovery-timeout disable
switch(config-eoam-action)# dying-gasp error-disable-interface
switch(config-eoam-action)# high-threshold error-disable-interface
switch(config-eoam-action)# remote-loopback disable
switch(config-eoam-action)# session-down error-disable-interface
switch(config-eoam-action)# session-up disable
switch(config-eoam-action)# uni-directional link-fault disable
switch(config-eoam-action)# wiring-conflict disable
```

Configuration Example for Attaching an Ethernet OAM Profile to a Specific Interface

```
switch# configure terminal
switch(config)# interface Ethernet 3/2
switch(config-if)# ethernet oam
switch(config-if-eoam)# profile Profile_1
```

Configuration Example for Configuring Ethernet OAM Features on a Specific Interface

```
switch# configure terminal
switch(config)# interface ethernet 3/2
switch(config-if)# ethernet oam
switch(config-if-eoam)# link-monitor
switch(config-if-eoam)# hello-interval 1s
switch(config—if-eoam-lm)# symbol-period window 60000
switch(config-if-eoam-lm)# symbol-period threshold low 10000000 high 60000000
switch(config-if-eoam-lm)# frame window 60
switch(config-if-eoam-lm)# frame threshold low 10000000 high 60000000
switch(config-if-eoam-lm)# frame-period window 60000
switch(config-if-eoam-lm)# frame-period threshold low 100 high 1000000
switch(config-if-eoam-lm)# frame-seconds window 900000
switch(config-if-eoam-lm)# frame-seconds threshold 3 threshold 900
switch(config-if-eoam-lm)# exit
switch(config-if-eoam)# connection timeout 30
switch(config-if-eoam)# mode passive
switch(config-if-eoam)# require-remote
switch(config-if-eoam-require)# mode active
switch(config-if-eoam-require)# link-monitoring
switch(config-if-eoam-require)# exit
switch(config-if-eoam)# action
switch(config-if-eoam-action)# capabilities-conflict disable
switch(config-if-eoam-action)# critical-event error-disable-interface
```

```
switch(config-if-eoam-action)# discovery-timeout disable
switch(config-if-eoam-action)# dying-gasp error-disable-interface
switch(config-if-eoam-action)# high-threshold error-disable-interface
switch(config-if-eoam-action)# remote-loopback disable
switch(config-if-eoam-action)# session-down error-disable-interface
switch(config-if-eoam-action)# session-up disable
switch(config-if-eoam-action)# uni-directional link-fault disable
switch(config-if-eoam-action)# wiring-conflict disable
```

Configuration Example for Configuration of Ethernet OAM Features in a Profile Followed by an Override of that Configuration on an Interface

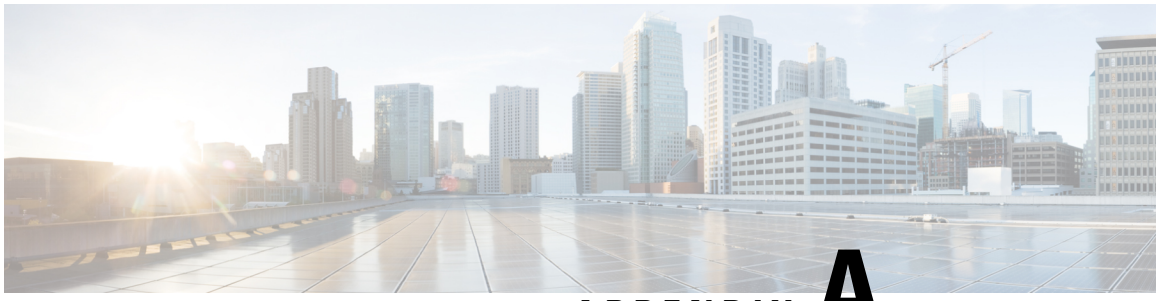
```
switch# configure terminal
switch(config)# ethernet oam profile Profile_1
switch(config-eoam)# mode passive
switch(config-eoam)# action
switch(config-eoam-action)# capabilities-conflict disable
switch(config-eoam-action)# critical-event error-disable-interface
switch(config-eoam-action)# discovery-timeout disable
switch(config-eoam-action)# dying-gasp error-disable-interface
switch(config-eoam-action)# remote-loopback disable
switch(config-eoam-action)# session-down error-disable-interface
switch(config-eoam-action)# session-up disable
switch(config-eoam-action)# uni-directional link-fault disable
switch(config-eoam-action)# wiring-conflict disable
```

```
switch# configure terminal
switch(config)# interface Ethernet 3/2
switch(config-if)# ethernet oam
switch(config-if-eoam)# profile Profile_1
switch(config-if-eoam)# mode active
switch(config-if-eoam)# action
switch(config-if-eoam-action)# capabilities-conflict disable
switch(config-if-eoam-action)# critical-event error-disable-interface
switch(config-if-eoam-action)# discovery-timeout disable
switch(config-if-eoam-action)# dying-gasp error-disable-interface
switch(config-if-eoam-action)# remote-loopback disable
switch(config-if-eoam-action)# session-down error-disable-interface
switch(config-if-eoam-action)# session-up disable
switch(config-if-eoam-action)# uni-directional link-fault disable
switch(config-if-eoam-action)# wiring-conflict disable
```

Related Documents

Table 60: Related Documents

Related Topic
Cisco NX-OS Licensing Guide
Cisco Nexus 7000 Series NX-OS Release Notes



APPENDIX A

IETF RFCs Supported by Cisco NX-OS Interfaces

Information about the Internet suite of protocols is contained in documents called Requests for Comments (RFCs).

Table 61: IETF RFCs Supported by Cisco NX-OS Interfaces

RFC	Title
RFC 1981	Path MTU Discovery for IP version 6
RFC 2373	IP Version 6 Addressing Architecture
RFC 2374	An Aggregatable Global Unicast Address Format
RFC 2460	Internet Protocol, Version 6 (IPv6) Specification
RFC 2461	Neighbor Discovery for IP Version 6 (IPv6)
RFC 2462	IPv6 Stateless Address Autoconfiguration
RFC 2463	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
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	Title
RFC 3596	DNS Extensions to Support IP version 6
RFC 4193	Unique Local IPv6 Unicast Addresses

• IETF RFCs supported by Cisco NX-OS Interfaces, on page 348

IETF RFCs supported by Cisco NX-OS Interfaces

RFCs	Title
RFC 1981	Path MTU Discovery for IP version 6
RFC 2373	IP Version 6 Addressing Architecture
RFC 2374	An Aggregatable Global Unicast Address Format
RFC 2460	Internet Protocol, Version 6 (IPv6) Specification
RFC 2461	Neighbor Discovery for IP Version 6 (IPv6)
RFC 2462	IPv6 Stateless Address Autoconfiguration
RFC 2463	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
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