Cisco Nexus 5600 Series NX-OS Interfaces Configuration Guide, Release 7.x

First Published: 2014-03-15

Last Modified: 2020-07-24

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

CONTENTS

PREFACE

Preface xiii

- Preface xiii
  - Audience xiii
  - Document Conventions xiii
  - Related Documentation xiv
  - Documentation Feedback xv
  - Communications, Services, and Additional Information xv

New and Changed Information xvii

CHAPTER 1

Configuring Layer 2 Interfaces 1

- Information About Ethernet Interfaces 1
  - Interface Command 1
  - Information About Unified Ports 2
    - Guidelines and Limitations for Unified Ports 2
  - Unidirectional Link Detection Parameter 3
    - Default UDLD Configuration 3
    - UDLD Aggressive and Nonaggressive Modes 4
  - Interface Speed 4
- Information About Ethernet Interfaces 4
  - Cisco Discovery Protocol 4
    - Default CDP Configuration 5
  - Error-Disabled State 5
  - About Port Profiles 5
    - Guidelines and Limitations for Port Profiles 6
  - Debounce Timer Parameters 7

- MTU Configuration 7
- Information About Default Interfaces 7
- Information About Access and Trunk Interfaces 8
  - Understanding Access and Trunk Interfaces 8
  - Understanding IEEE 802.1Q Encapsulation 9
  - Understanding Access VLANs 9
  - Understanding the Native VLAN ID for Trunk Ports 10
  - Understanding Allowed VLANs 10
  - Understanding Native 802.1Q VLANs 10
- Configuring Access and Trunk Interfaces 11
  - Configuring a LAN Interface as an Ethernet Access Port 11
  - Configuring Access Host Ports 12
  - Configuring Trunk Ports 12
  - Configuring the Native VLAN for 802.1Q Trunking Ports 13
  - Configuring the Allowed VLANs for Trunking Ports 14
  - Configuring Native 802.1Q VLANs 15
- Verifying the Interface Configuration 15
- Configuring Ethernet Interfaces 16
  - Configuring Unified Ports 16
  - Configuring the UDLD Mode 18
  - Disabling Link Negotiation 20
  - Configuring the CDP Characteristics 21
  - Enabling or Disabling CDP 22
  - Enabling the Error-Disabled Detection 22
  - Enabling the Error-Disabled Recovery 23
  - Configuring the Error-Disabled Recovery Interval 24
  - Configuring a Default Interface 24
  - Configuring Default Interface Mode 26
  - Configuring the Description Parameter 26
  - Disabling and Restarting Ethernet Interfaces 27
- Configuring Slow Drain Device Detection and Congestion Avoidance 27
  - Fibre Channel Slow Drain Device Detection and Congestion Avoidance- An Overview 27
  - Configuring a Stuck Frame Timeout Value 28
  - Configuring a No-Credit Timeout Value 28

Displaying Credit Loss Counters 29
Displaying Credit Loss Events 30
Displaying Timeout Drops 30
Displaying the Average Credit Not Available Status 30
Port Monitoring 30
Enabling Port Monitor 31
Configuring a Port Monitor Policy 31
Activating a Port Monitor Policy 32
Displaying Port Monitor Policies 32
FCoE Slow Drain Device Detection and Congestion Avoidance 32
Configuring a Pause Frame Timeout Value 34
Displaying Interface Information 35
Default Physical Ethernet Settings 38
CHAPTER 2 Configuring Layer 3 Interfaces 39
Information About Layer 3 Interfaces 39
Routed Interfaces 39
Subinterfaces 40
VLAN Interfaces 40
Loopback Interfaces 41
IP Addressing Scheme with Private VLANs 41
Licensing Requirements for Layer 3 Interfaces 42
Guidelines and Limitations for Layer 3 Interfaces 42
Default Settings for Layer 3 Interfaces 43
Configuring Layer 3 Interfaces 43
Configuring a Routed Interface 43
Configuring a Subinterface 44
Configuring the Bandwidth on an Interface 45
Configuring a VLAN Interface 45
Mapping Secondary VLANs to the Layer 3 VLAN Interface of a Primary VLAN 46
Configuring a Loopback Interface 48
Assigning an Interface to a VRF 48
Verifying the Layer 3 Interfaces Configuration 49
Monitoring Layer 3 Interfaces 50

Configuration Examples for Layer 3 Interfaces 51
Related Documents for Layer 3 Interfaces 52
MIBs for Layer 3 Interfaces 52
Standards for Layer 3 Interfaces 52

CHAPTER 3 Configuring Bidirectional Forwarding Detection 53
Information About BFD 53
Asynchronous Mode 53
Detection of Failures 54
BFD Echo Function 55
Security 55
Virtualization Support 55
Licensing Requirements for BFD 55
Prerequisites 56
Guidelines and Limitations 56
BFD Default Settings 57
Configuring BFD 58
BFD Configuration Hierarchy 58
Task Flow for Configuring BFD 58
Enabling the BFD Feature 58
Configuring Global BFD Parameters 59
Configuring BFD on an Interface 60
Configuring BFD on a Port Channel 61
Configuring BFD Echo Function 63
Optimizing BFD on Subinterfaces 64
Configuring BFD Support for Routing Protocols 65
Configuring BFD on BGP 65
Configuring BFD on EIGRP 66
Configuring BFD on OSPF 67
Configuring BFD on HSRP 68
Configuring BFD on VRRP 69
Configuring BFD on Static Routes 70
Configuring BFD on PIM 71
Disabling BFD on an Interface 72

Configuring FabricPath BFD on All IS-IS Interfaces 72
Configuring FabricPath BFD on a Specific Interface 73
Verifying the BFD Configuration 74
Monitoring BFD 74
Configuration Examples for BFD 75
Related Documents 75
Related Standards 75
CHAPTER 4 Configuring Port Channels 77
Information About Port Channels 77
Understanding Port Channels 77
Compatibility Requirements 78
Load Balancing Using Port Channels 80
Understanding LACP 81
LACP Overview 81
LACP ID Parameters 82
Channel Modes 83
LACP Marker Responders 83
LACP-Enabled and Static Port Channel Differences 84
LACP Port-Channel MinLinks and MaxBundle 84
Configuring Port Channels 85
Default Settings 85
LACP Port-Channel Min Links 85
Creating a Port Channel 86
Adding a Port to a Port Channel 86
Configuring Load Balancing Using Port Channels 87
Enabling LACP 89
Configuring the Channel Mode for a Port 89
Configuring LACP Port-Channel Minimum Links 90
Configuring the LACP Port-Channel MaxBundle 91
Configuring the LACP Fast Timer Rate 92
Configuring the LACP Short-timeout 93
Procedure to revert when BFD Feature and LACP Short-timeout are configured Together 94
Configuring the LACP System Priority and System ID 94

Configuring the LACP Port Priority 95

Disabling LACP Graceful Convergence 96

Reenabling LACP Graceful Convergence 97

Verifying Port Channel Configuration 98

Verifying the Load-Balancing Outgoing Port ID 99

Feature History for Configuring Port Channels 100

CHAPTER 5 **Configuring Enhanced Virtual Port Channels** 101

Information About Enhanced vPCs 101

Enhanced Virtual Port Channels Overview 101

Supported Platforms and Topologies 102

Enhanced vPC Scalability 102

Enhanced vPC Failure Response 103

Licensing Requirements for Enhanced vPC 103

Configuring Enhanced vPCs 104

Overview of Configuration Steps for Enhanced vPC 104

Verifying Enhanced vPCs 104

Verifying the Enhanced vPC Configuration 104

Verifying the Consistency of Port Channel Numbers 105

Verifying Common Port Channel Members 106

Verifying Interface Level Consistency for Enhanced vPCs 108

Enhanced vPC Example Configuration 108

CHAPTER 6 **Configuring Virtual Port Channels** 113

Information About vPCs 113

vPC Overview 113

Terminology 114

vPC Terminology 114

Supported vPC Topologies 115

Cisco Nexus Device vPC Topology 115

vPC Domain 115

Peer-Keepalive Link and Messages 115

Compatibility Parameters for vPC Peer Links 116

Configuration Parameters That Must Be Identical 116

Configuration Parameters That Should Be Identical 117

Graceful Type-1 Check 118

Per-VLAN Consistency Check 118

vPC Auto-Recovery 118

vPC Peer Links 118

vPC Peer Link Overview 118

vPC Number 119

Layer 3 over vPC 120

Layer 3 over vPC: Supported Designs 120

Layer 3 over vPC: Unsupported Designs 122

vPC Interactions with Other Features 125

Configuring vPC Peer Links and Links to the Core 125

vPC and LACP 127

vPC Peer Links and STP 127

vPC and ARP 128

CFSoE 128

vPC Peer Switch 129

Guidelines and Limitations for vPCs 129

Configuring vPCs 130

Enabling vPCs 130

Disabling vPCs 131

Creating a vPC Domain 131

Configuring a vPC Keepalive Link and Messages 132

Creating a vPC Peer Link 134

Checking the Configuration Compatibility 135

Enabling vPC Auto-Recovery 136

Configuring the Restore Time Delay 137

Excluding VLAN Interfaces from Shutting Down a vPC Peer Link Fails 138

Configuring the VRF Name 138

Binding a VRF Instance to a vPC 139

Moving Other Port Channels into a vPC 140

Manually Configuring a vPC Domain MAC Address 141

Manually Configuring the System Priority 141

Manually Configuring a vPC Peer Switch Role 142

Configuring the vPC Peer Switch 143
Configuring a Pure vPC Peer Switch Topology 143
Configuring a Hybrid vPC Peer Switch Topology 145
Configuring Layer 3 over vPC 146
Isolating and Restoring a Switch from the vPC Complex 148
Configuring vPC Shutdown 148
Restoring a vPC Shutdown Switch 148
Verifying the vPC Configuration 149
Viewing the Graceful Type-1 Check Status 150
Viewing a Global Type-1 Inconsistency 150
Viewing an Interface-Specific Type-1 Inconsistency 151
Viewing a Per-VLAN Consistency Status 153
Viewing Dual Active Detection Status 155
vPC Default Settings 155

CHAPTER 7 Configuring Linecard Expansion Modules 157
Configuring Breakout 157
Information About Linecard Expansion Modules 157
Configuring Breakout in a Port 157
Removing the Breakout Configuration 159
Verifying a Breakout Configuration 160

CHAPTER 8 Configuring Q-in-Q VLAN Tunnels 161
Information About Q-in-Q VLAN Tunnels 161
Q-in-Q Tunneling 161
Native VLAN Hazard 163
Information About Layer 2 Protocol Tunneling 164
Licensing Requirements for Q-in-Q Tunnels 166
Guidelines and Limitations for Q-in-Q VLAN Tunnels 166
Configuring Q-in-Q Tunnels and Layer 2 Protocol Tunneling 168
Creating an 802.1Q Tunnel Port 168
Changing the EtherType for Q-in-Q 169
Enabling the Layer 2 Protocol Tunnel 170
Configuring Global CoS for Layer 2 Protocol Tunnel Ports 171

Configuring the Rate Limit for Layer 2 Protocol Tunnel Ports 172

Configuring Thresholds for Layer 2 Protocol Tunnel Ports 172

Verifying the Q-in-Q Configuration 173

Configuration Examples for Q-in-Q and Layer 2 Protocol Tunneling 174

Preface

The preface contains the following sections:

- Preface, on page xiii

Preface

This preface describes the audience, organization, and conventions of the Book Title. It also provides information on how to obtain related documentation.

This chapter includes the following topics:

Audience

This publication is for experienced network administrators who configure and maintain Cisco NX-OS on Cisco Nexus 5000 Series Platform switches.

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.
- The Guidelines and Limitations section contains general guidelines and limitations that are applicable to all the features, and the feature-specific guidelines and limitations that are applicable only to the corresponding feature.

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.

Convention	Description
[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.
{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

Documentation for Cisco Nexus 5000 Series Switches is available at:

- Configuration Guides

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

- Command Reference Guides

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

- Release Notes

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

- Install and Upgrade Guides

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

- Licensing Guide

  http://www.cisco.com/c/en/us/support/switches/nexus-5000-series-switches/products-licensing-information-listing.html

Documentation for Cisco Nexus 5000 Series Switches and Cisco Nexus 2000 Series Fabric Extenders is available at:

http://www.cisco.com/c/en/us/support/switches/nexus-2000-series-fabric-extenders/products-installation-and-configuration-guides-list.html

Documentation Feedback

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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

New and Changed Information

The table below summarizes the new and changed features for this document and shows the releases in which each feature is supported. 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.

Note The new and changed information topic is added in the configuration guide from the Cisco Nexus Release 7.3(0)N1(1) onwards.

Cisco NX-OS Release Number	Platform Supported	New/Enhanced Features	Chapter/Topic Where Documented
7.3(0)N1(1)	Cisco Nexus 5600 and 6000 Series Switches	LACP Short-timeout	Configuring Port Channels
7.3(0)N1(1)	Cisco Nexus 5600 and 6000 Series Switches	Support for Layer3 over vPC	Configuring Virtual Port Channels

CHAPTER 1

Configuring Layer 2 Interfaces

This chapter contains the following sections:

- Information About Ethernet Interfaces, on page 1
- Information About Ethernet Interfaces, on page 4
- Information About Default Interfaces, on page 7
- Information About Access and Trunk Interfaces, on page 8
- Configuring Access and Trunk Interfaces, on page 11
- Verifying the Interface Configuration, on page 15
- Configuring Ethernet Interfaces, on page 16
- Configuring Slow Drain Device Detection and Congestion Avoidance, on page 27
- FCoE Slow Drain Device Detection and Congestion Avoidance, on page 32
- Displaying Interface Information, on page 35
- Default Physical Ethernet Settings , on page 38

Information About Ethernet Interfaces

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

The Ethernet interfaces are enabled by default.

Interface Command

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

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

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

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

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

Note After you perform an upgrade from Cisco NX-OS 6.0(2)A7(2) to Cisco NX-OS 6.0(2)A8(10) and later, you may see the display format of transceiver type for DACs changed to decimal format. However, there wil be no change in the functionality of the device.

Information About Unified Ports

Cisco Nexus unified ports allow you to configure a physical port on a Cisco Nexus device switch as a 1/10-Gigabit Ethernet, Fibre Channel over Ethernet (FCoE), or 2-, 4-, 8-Gigabit native Fibre Channel port.

Currently, most networks have two types of switches for different types of networks. For example, LAN switches carry Ethernet traffic up to Catalyst or Nexus switches carry FC traffic from servers to MDS switches. With unified port technology, you can deploy a unified platform, unified device, and unified wire approach. Unified ports allow you to move from an existing segregated platform approach where you choose LAN and SAN port options to transition to a single, unified fabric that is transparent and consistent with existing practices and management software. A unified fabric includes the following:

- Unified platform—Uses the same hardware platform and the same software code level and certifies it once for your LAN and SAN environments.
- Unified device—Runs LAN and SAN services on the same platform switch. The unified device allows you to connect your Ethernet and Fibre Channel cables to the same device.
- Unified wire—Converges LAN and SAN networks on a single converged network adapter (CNA) and connects them to your server.

A unified fabric allows you to manage Ethernet and FCoE features independently with existing Cisco tools.

Guidelines and Limitations for Unified Ports

Note

- All ports of same type (Fibre Channel or Ethernet) should be contiguous on the module.
- On a Cisco Nexus 5672UP switch, the Fibre Channel port range can be among 33-48, but must end at Port 48.
- On a Cisco Nexus 5672UP-16G switch, the Fibre Channel port range can be among 2/1-2/24 or 2/13-2/24.
- On a Cisco Nexus 56128P switch, only the expansion modules in slot 2 and 3 support native FC type. On each module, the Fibre Channel port range can be among 1-24, but must start from Port 1.
- On a Cisco Nexus 5696Q switch, only M20UP expansion modules support native FC type. All 20 ports can be configured as native Fibre Channel ports, but the port range must either start with 1 or end at 20.

Unidirectional Link Detection Parameter

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

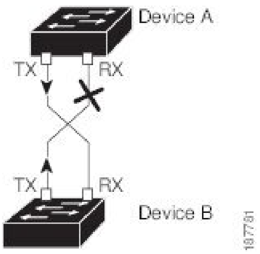
UDLD is a Layer 2 protocol that works with the Layer 1 protocols to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting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

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

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

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

Figure 1: Unidirectional Link



Default UDLD Configuration

The following table shows the default UDLD configuration.

Table 1: UDLD Default Configuration

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

Feature	Default Value
UDLD per-port enable state for twisted-pair (copper) media	Enabled

UDLD Aggressive and Nonaggressive Modes

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

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

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

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

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

Interface Speed

The Cisco Nexus 6004 switch has default port in 40 Gigabit Ethernet mode. The port speed can be changed in group of 12 Quad Small Form-factor Pluggable (QSFP) ports. You need to reset the group after the port mode is changed. The hardware support is provided for port speed of every 3 QSFP interfaces.

Information About Ethernet Interfaces

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

The Ethernet interfaces are enabled by default.

Cisco Discovery Protocol

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

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

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

The switch supports both CDP Version 1 and Version 2.

Default CDP Configuration

The following table shows the default CDP configuration.

Table 2: Default CDP Configuration

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

Error-Disabled State

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

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

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

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

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

About Port Profiles

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

- Ethernet
- VLAN network interface

- Port channel

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

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

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

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

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

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

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

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

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

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

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

Guidelines and Limitations for Port Profiles

Port profiles have the following configuration guidelines and limitations:

- Each port profile must have a unique name across interface types and the network.
- Commands that you enter under the interface mode take precedence over the port profile’s commands if there is a conflict. However, the port profile retains that command in the port profile.

- The port profile’s commands take precedence over the default commands on the interface, unless the default command explicitly overrides the port profile command.
- After you inherit a port profile onto an interface or range of interfaces, you can override individual configuration values by entering the new value at the interface configuration level. If you remove the individual configuration values at the interface configuration level, the interface uses the values in the port profile again.
- There are no default configurations associated with a port profile.
- A subset of commands are available under the port profile configuration mode, depending on which interface type that you specify.
- You cannot use port profiles with Session Manager.

Debounce Timer Parameters

MTU Configuration

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

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

Information About Default Interfaces

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

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

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

Information About Access and Trunk Interfaces

Understanding Access and Trunk Interfaces

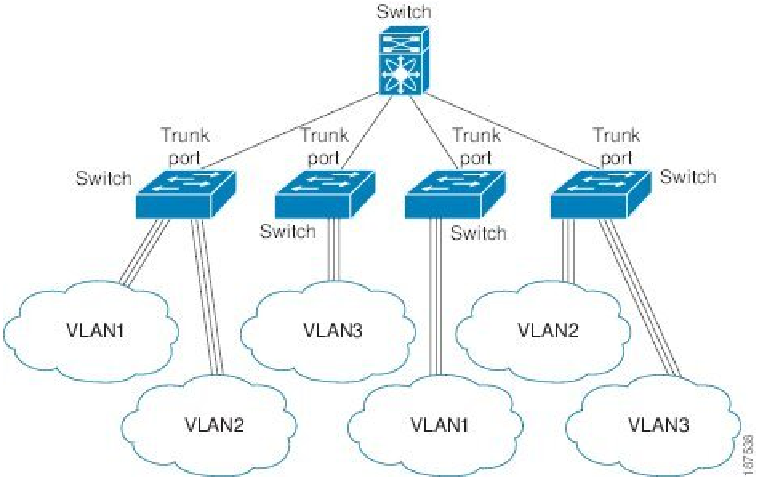
Ethernet interfaces can be configured either as access ports or a trunk ports, as follows:

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

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

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

Figure 2: Devices in a Trunking Environment



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

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

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

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

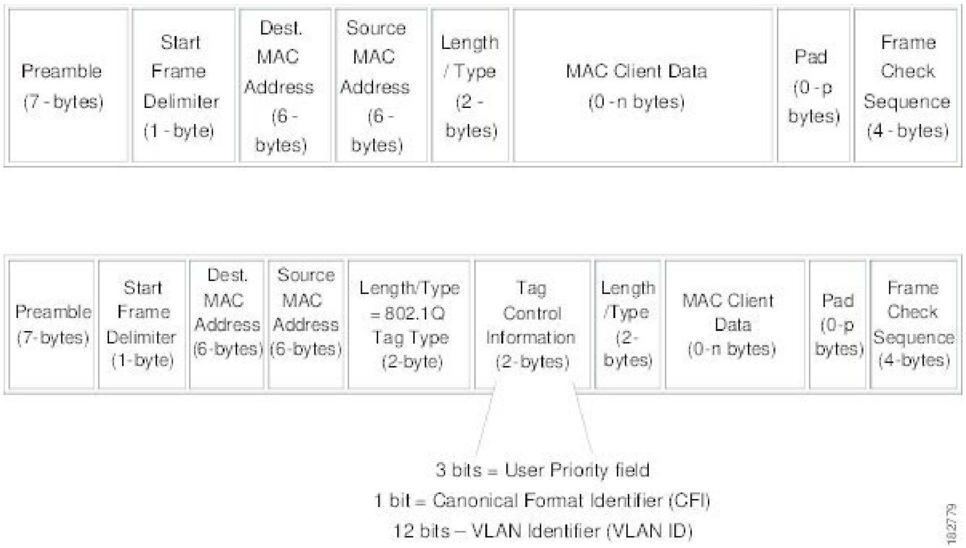
Note An Ethernet interface can function as either an access port or a trunk port; it cannot function as both port types simultaneously.

Understanding IEEE 802.1Q Encapsulation

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

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

Figure 3: Header Without and With 802.1Q Tag Included


Preamble (7 - bytes)
Start Frame Delimiter (1 - byte)
Dest. MAC Address (6 - bytes)
Source MAC Address (6 - bytes)
Length / Type (2 - bytes)
MAC Client Data (0 - n bytes)
Pad (0 - p bytes)
Frame Check Sequence (4 - bytes)
Preamble (7-bytes)
Start Frame Delimiter (1-byte)
Dest. MAC Address (6-bytes)
Source MAC Address (6-bytes)
Length/Type = 802.1Q Tag Type (2-byte)
Tag Control Information (2-bytes)
Length /Type (2-bytes)
MAC Client Data (0-n bytes)
Pad (0-p bytes)
Frame Check Sequence (4-bytes)
3 bits = User Priority field
1 bit = Canonical Format Identifier (CFI)
12 bits – VLAN Identifier (VLAN ID)
182779

Understanding Access VLANs

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

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

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

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

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

Understanding the Native VLAN ID for Trunk Ports

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

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

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

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

Understanding Allowed VLANs

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

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

Understanding Native 802.1Q VLANs

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

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

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

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

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

This feature is supported on all the directly connected Ethernet and Port Channel interfaces.

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

Configuring Access and Trunk Interfaces

Configuring a LAN Interface as an Ethernet Access Port

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

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

Procedure

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

Example

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

```
switch# configure terminal
switch(config)# interface ethernet 1/10
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
```

Configuring Access Host Ports

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

Before you begin

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

Procedure

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

Example

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

```
switch# configure terminal

switch(config)# interface ethernet 1/10

switch(config-if)# switchport host
```

Configuring Trunk Ports

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

Note Cisco NX-OS supports only 802.1Q encapsulation.

Procedure

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

Example

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

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

Configuring the Native VLAN for 802.1Q Trunking Ports

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

Procedure

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

Example

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

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

Configuring the Allowed VLANs for Trunking Ports

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

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

Procedure

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

Example

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

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

Configuring Native 802.1Q VLANs

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

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

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

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

Procedure

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

Example

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

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

Verifying the Interface Configuration

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

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

Configuring Ethernet Interfaces

The section includes the following topics:

Configuring Unified Ports

Before you begin

Confirm that you have a supported Cisco Nexus switch. Unified Ports are available on the following Cisco Nexus switches:

- Cisco Nexus 5672UP
- Cisco Nexus 5672UP-16G
- Cisco Nexus 56128P with N56-M24UP2Q LEMs
- Cisco Nexus 5696Q with N5696-M20UP LEMs

Note For information about the N5672UP-16G platform details, see the *Cisco Nexus 5600 Series Hardware Installation Guide*.

If you're configuring a unified port as Fibre Channel or FCoE, confirm that you have enabled the **feature fcoe** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config) # **slot** *slot number*	Identifies the slot on the switch.
Step 3	switch(config-slot) # **port** *port number* **type** {**ethernet** | **fc**}	Configures a unified port as a native Fibre Channel port and an Ethernet port. • **type**—Specifies the type of port to configure on a slot in a chassis. • **ethernet**—Specifies an Ethernet port.

	Command or Action	Purpose
		• **fc**—Specifies a Fibre Channel (FC) port. **Note** • Changing unified ports on an expansion module (GEM) requires that you power cycle the GEM card. You do not have to reboot the entire switch for changes to take effect. • When you configure unified ports as Fibre Channel, the existing configuration for Fibre Channel interfaces and VSAN memberships are unaffected. **Note** When configuring an FC port on N5672-16G, the fabric mode should be in the 40-G mode to support 16-G. When the ports are changed from Ethernet to FC, the fabric mode changes to 40-G on the next reload. When the ports are changed to FC for the first time, the following message is displayed: "Port type is changed. Fabric mode is also changed. Please copy configuration and reload the switch." Use **show fabric-mode** to verify the current fabric mode configuration. The FC ports can be configured only on Module 2 of Nexus 5672UP-16G. The FC port range must be in multiples of 12, either 1-24 or 13-24. Reload of the module is sufficient, when you increase or decrease the range of FC ports.
Step 4	switch(config-slot) # **copy running-config startup-config**	Copies the running configuration to the startup configuration.
Step 5	switch(config-slot) # **reload**	Reboots the switch.

	Command or Action	Purpose
Step 6	switch(config) # **slot** *slot number*	Identifies the slot on the switch.
Step 7	switch(config-slot) # **no port** *port number* **type fc**	Removes the unified port. **Note** When all the FC ports are removed, the fabric mode changes to the 10-G mode. When all the ports are changed to Ethernet, the following message is displayed: "Port type is changed. Fabric mode is also changed. Please copy configuration and reload the switch."

Example

This example shows how to configure a unified port on a Cisco N5696-M20UP expansion module:

```
switch# configure terminal
switch(config)# slot 2
switch(config-slot)# port 1-20 type fc
switch(config-slot)# copy running-config startup-config
switch(config-slot)# poweroff module 2
switch(config-slot)# no poweroff module 2
```

This example shows how to convert ports 1-24 or 13-24 to FC ports in N5672UP-16G:

Note Individual ports cannot be converted to FC ports. In N5672UP-16G, only Slot 2 has UP ports.

```
switch# configure terminal
switch(config)# slot 2
switch(config-slot)# port 1-24 type fc
Port type is changed. Fabric mode is also changed .. Please copy configuration and reload
the switch
switch(config-slot)#
```

Or

```
switch# configure terminal
switch(config)# slot 2
switch(config-slot)# port 13-24 type fc
Port type is changed. Please power-off and no power-off the module
switch(config-slot)#
```

Configuring the UDLD Mode

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

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

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

Procedure

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

Example

This example shows how to enable UDLD for the switch:

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

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

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

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

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

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

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

This example shows how to disable UDLD for the switch:

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

Disabling Link Negotiation

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

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

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

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

Procedure

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

Example

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

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

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

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

Configuring the CDP Characteristics

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# [**no**] **cdp advertise** {**v1** | **v2** }	Configures the version to use to send CDP advertisements. Version-2 is the default state. Use the **no** form of the command to return to its default setting.
Step 3	(Optional) switch(config)# [**no**] **cdp format device-id** {**mac-address** | **serial-number** | **system-name**}	Configures the format of the CDP device ID. The default is the system name, which can be expressed as a fully qualified domain name. Use the **no** form of the command to return to its default setting.
Step 4	(Optional) switch(config)# [**no**] **cdp holdtime** *seconds*	Specifies the amount of time a receiving device should hold the information sent by your device before discarding it. The range is 10 to 255 seconds; the default is 180 seconds. Use the **no** form of the command to return to its default setting.
Step 5	(Optional) switch(config)# [**no**] **cdp timer** *seconds*	Sets the transmission frequency of CDP updates in seconds. The range is 5 to 254; the default is 60 seconds. Use the **no** form of the command to return to its default setting.

Example

This example shows how to configure CDP characteristics:

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

Enabling or Disabling CDP

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

Procedure

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

Example

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

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

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

Enabling the Error-Disabled Detection

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

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

Procedure

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

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

Example

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

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

Enabling the Error-Disabled Recovery

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

Procedure

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

Example

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

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

Configuring the Error-Disabled Recovery Interval

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **errdisable recovery interval** *interval*	Specifies the interval for the interface to recover from the err-disabled state. The range is from 30 to 65535 seconds. The default is 300 seconds.
Step 3	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

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

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

Configuring a Default Interface

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **default interface** *int-if* [**checkpoint** *name*]	Deletes the configuration of the interface and restores the default configuration. The value of *int-if* can be one of the following: • ethernet • loopback • mgmt • port-channel • vlan Use the **checkpoint** keyword to store a copy of the running configuration of the interface before clearing the configuration.
Step 3	**exit**	Exits the configuration mode.
Step 4	(Optional) **show interface**	Displays the interface status and information.

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)# show running-config interface e1/10
!Command: show running-config interface Ethernet1/10
!Time: Tue Jul 2 10:23:50 2013

version 6.0(2)N2(1)

interface Ethernet1/10
switchport mode trunk
channel-group 1

default interface ethernet 3/1 checkpoint chk1
.......Done
switch(config)# show running-config interface e1/10
!Command: show running-config interface Ethernet1/10
!Time: Tue Jul 2 10:24:41 2013

version 6.0(2)N2(1)

interface Ethernet1/10

switch(config)#
```

Configuring Default Interface Mode

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **system default switchport** [**shutdown**]	Sets default interface mode. **Note** When the **system default switchport shutdown** command is issued, any switchports (including FEX HIFs) that are not configured with **no shutdown** command are shut down. To avoid the shutdown, configure the switchports with **no shutdown** command.

Example

This example shows how to set the default interface mode:

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

Configuring the Description Parameter

You can provide textual interface descriptions for the Ethernet ports.

Procedure

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

Example

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

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

Disabling and Restarting Ethernet Interfaces

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **shutdown**	Disables the interface.
Step 4	switch(config-if)# **no shutdown**	Restarts the interface.

Example

This example shows how to disable an Ethernet port:

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

This example shows how to restart an Ethernet interface:

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

Configuring Slow Drain Device Detection and Congestion Avoidance

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

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

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

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

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

Configuring a Stuck Frame Timeout Value

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

Procedure

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

Example

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

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

Configuring a No-Credit Timeout Value

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

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

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

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

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

Procedure

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

Example

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

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

Displaying Credit Loss Counters

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

Procedure

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

Displaying Credit Loss Events

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

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

Displaying Timeout Drops

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

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

Displaying the Average Credit Not Available Status

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

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

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

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

Port Monitoring

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

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

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

Enabling Port Monitor

Procedure

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

Configuring a Port Monitor Policy

Procedure

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

Example

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

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

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

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

Activating a Port Monitor Policy

Procedure

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

Example

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

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

Displaying Port Monitor Policies

Use the following command to display port monitor policies:

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

Example

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

FCoE Slow Drain Device Detection and Congestion Avoidance

The data traffic between end devices in Fibre Channel over Ethernet (FCoE) uses link level, per-hop Priority Flow Control (PFC). This allows the FCoE class on a link to be paused independently in each direction, while other classes continue to transmit and receive on the link. When end devices transmit PFC pause frames to the switch port they prevent the switch port from being able to transmit FCoE frames to the end device. Although some of this occurs normally, if it occurs in large amounts it can cause congestion in the fabric. End devices doing this are called a slow devices, or slow drain devices. When this occurs it can cause frames to queue at the switch which results in the switch transmitting its own PFC pause frames back towards the source of the incoming frames. If the switch port where the frames are being received (the source of the incoming frames) is connected to an end device, then this end device will temporarily be paused. It will not be able to transmit any frames into the switch for any destination (not just for the slow device). If switch port where the frames are being received on is an Inter-Switch-Link (ISL) then all inbound traffic across that ISL will be paused. This will affect all devices transiting that ISL.

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

- Congestion timeout, on page 33
- Pause timeout, on page 33

Congestion timeout

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

Pause timeout

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

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

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

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

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

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

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

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

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

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

Configuring a Pause Frame Timeout Value

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

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

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

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch# **system default interface pause timeout** *milliseconds* **mode** {**core** | **edge**}	Configures a new pause frame timeout value in milliseconds and the port mode for the device.
Step 3	switch# **system default interface pause mode** {**core** | **edge**}	Configures the default pause frame timeout value in milliseconds and the port mode for the device.
Step 4	switch# **no system default interface pause timeout** *milliseconds* **mode** {**core** | **edge**}	Disables the pause frame timeout for the device.
Step 5	switch# **no system default interface pause mode** {**core** | **edge**}	Disables the default pause frame timeout for the device.

	Command or Action	Purpose
Step 6	(Optional) switch# **show logging onboard flow-control pause-event**	Displays the total number of the pause events per module per interface.
Step 7	(Optional) switch# **show logging onboard flow-control timeout-drop**	Displays the timeout drops per module per interface with the time-stamp information.

Example

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

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

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

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

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

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

Displaying Interface Information

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

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

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

This example shows how to display the physical Ethernet interface:

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

This example shows how to display the physical Ethernet capabilities:

```
switch# show interface ethernet 1/1 capabilities
Ethernet1/1
  Model:                 734510033
  Type:                  10Gbase-(unknown)
  Speed:                 1000,10000
  Duplex:                full
  Trunk encap. type:     802.1Q
  Channel:               yes
  Broadcast suppression: percentage(0-100)
  Flowcontrol:           rx-(off/on),tx-(off/on)
  Rate mode:             none
  QOS scheduling:        rx-(6q1t),tx-(1p6q0t)
  CoS rewrite:           no
  ToS rewrite:           no
  SPAN:                  yes
  UDLD:                  yes
  Link Debounce:         yes
  Link Debounce Time:    yes
  MDIX:                  no
  FEX Fabric:            yes
```

This example shows how to display the physical Ethernet transceiver:

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

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

```
switch# show interface brief

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

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

```
switch# show interface debounce

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

```
Eth1/3        enable              100
...
```

This example shows how to display the CDP neighbors:

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

Default Physical Ethernet Settings

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

Parameter	Default Setting
Debounce	Enable, 100 milliseconds
Duplex	Auto (full-duplex)
Encapsulation	ARPA
MTU[1]	1500 bytes
Port Mode	Access
Speed	Auto (10000)

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

CHAPTER 2

Configuring Layer 3 Interfaces

This chapter contains the following sections:

- Information About Layer 3 Interfaces, on page 39
- Licensing Requirements for Layer 3 Interfaces, on page 42
- Guidelines and Limitations for Layer 3 Interfaces, on page 42
- Default Settings for Layer 3 Interfaces, on page 43
- Configuring Layer 3 Interfaces, on page 43
- Verifying the Layer 3 Interfaces Configuration, on page 49
- Monitoring Layer 3 Interfaces, on page 50
- Configuration Examples for Layer 3 Interfaces, on page 51
- Related Documents for Layer 3 Interfaces, on page 52
- MIBs for Layer 3 Interfaces, on page 52
- Standards for Layer 3 Interfaces, on page 52

Information About Layer 3 Interfaces

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

Routed Interfaces

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

All Ethernet ports are switched 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.

You can also create a Layer 3 port channel from routed interfaces.

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.

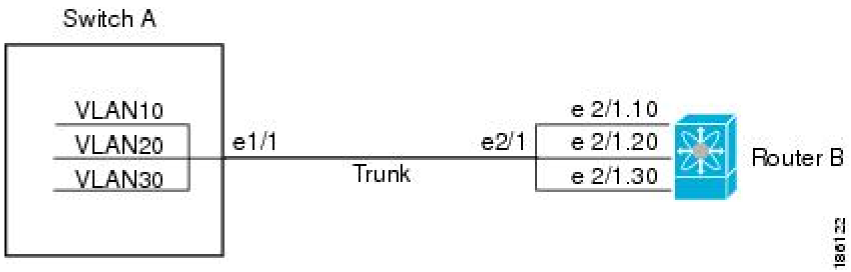
You create a subinterface with a name that consists of the parent interface name (for example, Ethernet 2/1) followed by a period and then by a number that is unique for that subinterface. For example, you could create a subinterface for Ethernet interface 2/1 named Ethernet 2/1.1 where .1 indicates 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 VLAN that is supported by the parent interface. In this scenario, the parent interface connects to a Layer 2 trunking port on another device. You configure a subinterface and associate the subinterface to a VLAN ID using 802.1Q trunking.

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

Figure 4: Subinterfaces for VLANs



VLAN Interfaces

A VLAN interface or a 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.

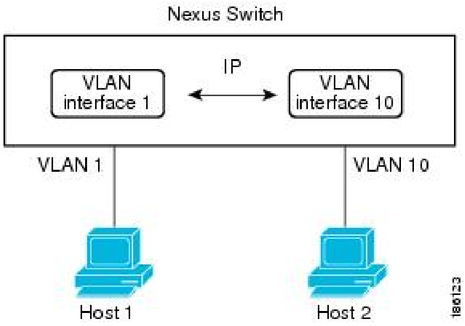
You must enable the VLAN network interface feature before you can configure it. The system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. For information about rollbacks and checkpoints, see the System Management Configuration Guide for your device.

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 on IP addresses and IP routing, see the Unicast Routing Configuration Guide for your device.

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

Figure 5: Connecting Two VLANs with VLAN Interfaces



Loopback Interfaces

A loopback interface is a virtual interface with a single endpoint that is always up. Any packet that is transmitted over a loopback interface is immediately received by this interface. Loopback interfaces emulate a physical interface.

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.

IP Addressing Scheme with Private VLANs

When you assign a separate VLAN to each customer, an inefficient IP addressing scheme is created as follows:

- Assigning a block of addresses to a customer VLAN can result in unused IP addresses.
- If the number of devices in the VLAN increases, the number of assigned addresses might not be large enough to accommodate them.

These problems are reduced by using private VLANs, where all members in the private VLAN share a common address space, which is allocated to the primary VLAN. Hosts are connected to secondary VLANs, and the DHCP server assigns them IP addresses from the block of addresses allocated to the primary VLAN. Subsequent IP addresses can be assigned to customer devices in different secondary VLANs, but in the same primary VLAN. When new devices are added, the DHCP server assigns them the next available address from a large pool of subnet addresses.

Licensing Requirements for Layer 3 Interfaces

After installing a Layer 3 license, the following guidelines and limitations apply to the device:

- In Service Software Upgrades (ISSUs) are not supported.
- Temporary Layer 3 feature licenses are not supported. (The Layer 3 Base Services Package license has a grace period of 0.)
- Management Switch Virtual Interfaces (SVIs) are supported without a Layer 3 Base Services Package license, and ISSU can be performed with Management SVIs configured.
- All SVIs (whether management keyword is configured or not) are operationally up when no Layer 3 Base Services Package license is installed. After the Layer 3 Base Services Packages feature license is installed, routed SVIs are brought operationally down and then brought back up again. This reload happens because the routed SVIs behave like management SVIs before a Layer 3 Base Services Packages feature license is installed, and the interface state saved in the hardware needs to be cleared followed by programming of the SVI routes in the Forwarding Information Base (FIB).
- After clearing a Layer 3 license, you must copy the running-configuration to the startup-configuration and reload the device. Then, you can perform a non-disruptive ISSU.
- After clearing a Layer 3 license, you must copy the running-configuration to the startup-configuration and reload the device. Then, you can perform a non-disruptive ISSU.
- Although HSRP and VRRP do not need to be removed before clearing a Layer 3 license, we recommend that you clear their configurations as well.
- Although VRRP and HSRP can be configured without a Layer 3 license, they will not work without a Layer 3 license. If they are configured, non-disruptive ISSU is not supported.

Guidelines and Limitations for Layer 3 Interfaces

Layer 3 interfaces have the following configuration guidelines and limitations:

- 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.
- Beginning with Cisco Nexus release 7.2(1)N1(1), Cisco Nexus 5600 Series and Cisco Nexus 6000 Series Switches support 1019 Layer 3 physical interfaces. In earlier release versions, only 59 Layer 3 physical interfaces with sub interfaces are supported.
- FHRP is supported only for VPC VLANs, and not supported for non-VPC VLAN with VPC topologies. Refer to Table 1 in Supported Topologies for Routing over Virtual Port Channel on Nexus Platforms.

Default Settings for Layer 3 Interfaces

The default setting for the Layer 3 Admin state is Shut.

Configuring Layer 3 Interfaces

Configuring a Routed Interface

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	switch(conifg-if)# **no switchport**	Configures the interface as a Layer 3 interface and deletes any configuration specific to Layer 2 on this interface. **Note** To convert a Layer 3 interface back into a Layer 2 interface, use the **switchport** command.
Step 4	switch(config-if)# [**ip**|**ipv6**]*ip-address/length*	Configures an IP address for this interface.
Step 5	(Optional) switch(config-if)# **medium** {**broadcast** | **p2p**}	Configures the interface medium as either point to point or broadcast. **Note** The default setting is broadcast, and this setting does not appear in any of the **show** commands. However, if you do change the setting to **p2p**, you will see this setting when you enter the **show running-config** command.
Step 6	(Optional) switch(config-if)# **show interfaces**	Displays the Layer 3 interface statistics.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure an IPv4-routed Layer 3 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
```

Configuring a Subinterface

Before you begin

- Configure the parent interface as a routed interface.
- Create the port-channel interface if you want to create a subinterface on that port channel.

Procedure

	Command or Action	Purpose
Step 1	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 2	switch(config)# **interface ethernet** *slot/port***.***number*	Enters interface configuration mode. The range for the *slot* is from 1 to 255. The range for the *port* is from 1 to 128.
Step 3	switch(config-if)# [**ip** | **ipv6**] **address** *ip-address/length*	Configures an IP address for this interface.
Step 4	switch(config-if)# **encapsulation dot1Q** *vlan-id*	Configures IEEE 802.1Q VLAN encapsulation on the subinterface. The range for the *vlan-id* is from 2 to 4093.
Step 5	(Optional) switch(config-if)# **show interfaces**	Displays the Layer 3 interface statistics.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a subinterface:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# encapsulation dot1Q 33
switch(config-if)# copy running-config startup-config
```

Configuring the Bandwidth on an Interface

You can configure the bandwidth for a routed interface, port channel, or subinterface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Enters interface configuration mode. The range for the *slot* is from 1 to 255. The range for the *port* is from 1 to 128.
Step 3	switch(conifg-if)# **bandwidth** [*value* | **inherit** [*value*]]	Configures the bandwidth parameter for a routed interface, port channel, or subinterface, as follows: • *value*—Size of the bandwidth in kilobytes. The range is from 1 to 10000000. • **inherit**—Indicates that all subinterfaces of this interface inherit either the bandwidth value (if a value is specified) or the bandwidth of the parent interface (if a value is not specified).
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure Ethernet interface 2/1 with a bandwidth value of 80000:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# bandwidth 80000
switch(config-if)# copy running-config startup-config
```

Configuring a VLAN Interface

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature interface-vlan**	Enables VLAN interface mode.

	Command or Action	Purpose
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.
Step 5	switch(config-if)# **no shutdown**	Brings the interface up administratively.
Step 6	(Optional) switch(config-if)# **show interface vlan** *number*	Displays the VLAN interface statistics. The *number* range is from 1 to 4094.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to 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
```

Mapping Secondary VLANs to the Layer 3 VLAN Interface of a Primary VLAN

To map secondary VLANs to the Layer 3 VLAN interface of a primary VLAN to allow Layer 3 switching of private VLAN ingress traffic, perform this task:

Procedure

	Command or Action	Purpose
Step 1	Router(config)# **interface-vlan** *primary_vlan_ID*	Enters interface configuration mode for the primary VLAN. **Note** Isolated and community VLANs are both called secondary VLANs.
Step 2	Router(config-if)# **private-vlan mapping** {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*}	Maps the secondary VLANs to the Layer 3 VLAN interface of a primary VLAN to allow Layer 3 switching of private VLAN ingress traffic. When you map secondary VLANs to the Layer 3 VLAN interface of a primary VLAN, note the following information:

	Command or Action	Purpose
		• The **private-vlan mapping interface** configuration command only affects private VLAN ingress traffic that is Layer 3-switched. • The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single private VLAN ID or a hyphenated range of private VLAN IDs. • Enter a *secondary_vlan_list* parameter or use the **add** keyword with a *secondary_vlan_list* parameter to map the secondary VLANs to the primary VLAN. • Use the **remove** keyword with a *secondary_vlan_list* parameter to clear the mapping between secondary VLANs and the primary VLAN.
Step 3	Router(config-if)# **no private-vlan mapping**	Clears the mapping between the secondary VLANs and the primary VLAN.
Step 4	Router(config-if)# **end**	Exits configuration mode.
Step 5	Router **show interface private-vlan mapping**	Verifies the configuration.

Example

This example shows how to permit routing of secondary VLAN ingress traffic from private VLANs 303 through 307, 309, and 440 and verify the configuration:

```
Router# configure terminal
 Router(config)# interface vlan 202
 Router(config-if)# private-vlan mapping add 303-307,309,440
 Router(config-if)# end
 Router# show interfaces private-vlan mapping
 Interface Secondary VLAN Type
 --------- -------------- -----------------
 vlan202   303            community
 vlan202   304            community
 vlan202   305            community
 vlan202   306            community
 vlan202   307            community
 vlan202   309            community
 vlan202   440            isolated
 Router#
```

Configuring a Loopback Interface

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface loopback** *instance*	Creates a loopback interface. The *instance* range is from 0 to 1023.
Step 3	switch(config-if)# [**ip** | **ipv6**] **address** *ip-address/length*	Configures an IP address for this interface.
Step 4	(Optional) switch(config-if)# **show interface loopback** *instance*	Displays the loopback interface statistics. The *instance* range is from 0 to 1023.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to 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

Before you begin

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** *interface-typenumber*	Enters interface configuration mode.
Step 3	switch(conifg-if)#**vrf member** *vrf-name*	Adds this interface to a VRF.

	Command or Action	Purpose
Step 4	switch(config-if)# [**ip** | **ipv6**]*ip-address/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-if)# **show vrf** [*vrf-name*] **interface** *interface-type number*	Displays VRF information.
Step 6	(Optional) switch(config-if)# **show interfaces**	Displays the Layer 3 interface statistics.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to 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

Use one of the following commands to verify the 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).

Command	Purpose
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.

Monitoring Layer 3 Interfaces

Use one of the following commands to display statistics about the feature:

Command	Purpose
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 error**	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. You cannot clear these counters.

Command	Purpose
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).
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 *counters* **snmp**	Displays the VLAN interface counters reported by SNMP MIBs. You cannot clear these counters.

Configuration Examples for Layer 3 Interfaces

This example shows how to configure Ethernet subinterfaces:

```
switch# configuration terminal
switch(config)# interface ethernet 2/1.10
switch(config-if)# description Layer 3 for VLAN 10
switch(config-if)# encapsulation dot1q 10
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

This example shows how to configure a VLAN interface:

```
switch# configuration terminal
switch(config)# interface vlan 100
switch(config-if)# no switchport

switch(config-if)# ipv6 address 33:0DB::2/8
switch(config-if)# copy running-config startup-config
```

This example shows how to configure a loopback interface:

```
switch# configuration terminal
switch(config)# interface loopback 3
switch(config-if)# no switchport
switch(config-if)# ip address 192.0.2.2/32
switch(config-if)# copy running-config startup-config
```

Related Documents for Layer 3 Interfaces

Related Topics	Document Title
Command syntax	
IP	“Configuring IP” chapter in the
VLAN	“Configuring VLANs” chapter in the

MIBs for Layer 3 Interfaces

MIB	MIB Link
IF-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml
CISCO-IF-EXTENSION-MIB	
ETHERLIKE-MIB	

Standards for Layer 3 Interfaces

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

CHAPTER 3

Configuring Bidirectional Forwarding Detection

This chapter contains the following sections:

- Information About BFD, on page 53
- Licensing Requirements for BFD, on page 55
- Prerequisites, on page 56
- Guidelines and Limitations, on page 56
- BFD Default Settings, on page 57
- Configuring BFD, on page 58
- Verifying the BFD Configuration, on page 74
- Monitoring BFD, on page 74
- Configuration Examples for BFD, on page 75
- Related Documents, on page 75
- Related Standards, on page 75

Information About BFD

The Bidirectional Forwarding Detection (BFD) provides 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 at variable rates for different protocol hello mechanisms. BFD makes network profiling and planning easier and reconvergence time consistent and predictable.

BFD makes all routing and control plane applications less CPU intensive by offloading Layer 3 hello message adjacency detection to a single process. The BFD process runs uniformly for all applications and detects failures at subsecond intervals.

A BFD global configuration command is available for fabricpath interfaces. The **bfd** keyword in the `config-farbicpath-isis` command mode, enables BFD on all the FabricPath interfaces. The interface specific commands override the global values.

For FabricPath BFD, fabricpath-isis is the client. FabricPath-ISIS is a part of feature-set fabricpath.

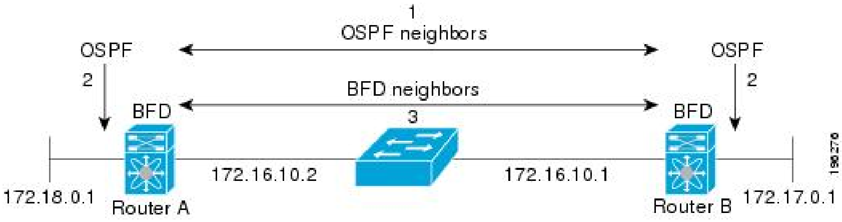
Asynchronous Mode

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). After BFD has been enabled on the interfaces and on the appropriate protocols, Cisco NX-OS creates a BFD session, negotiates BFD session parameters, and begins to send BFD control packets to each BFD neighbor at the negotiated interval. The BFD session parameters include the following:

- Desired minimum transmit interval—The interval at which this device wants to send BFD hello messages.
- Required minimum receive interval—The minimum interval at which this device can accept BFD hello messages from another BFD device.
- Detect multiplier—The number of missing BFD hello messages from another BFD device before this local device detects a fault in the forwarding path.

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

Figure 6: Establishing a BFD Neighbor Relationship



Detection of Failures

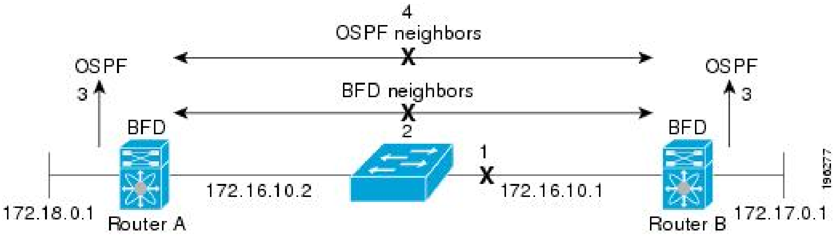
After a BFD session has been established and timer negotiations are complete, BFD neighbors send BFD control packets that act in the same manner as an IGP hello protocol to detect liveliness, except at a more accelerated rate. BFD detects a failure, but the protocol must take action to bypass a failed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

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

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

Note 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



BFD Echo Function

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

The echo function is asymmetric 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.

You can configure SHA-1 authentication of BFD packets.

Virtualization Support

BFD supports virtual routing and forwarding (VRF) instances.

Licensing Requirements for BFD

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	BFD requires a LAN Base Services license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Prerequisites

BFD has the following prerequisites:

- You must enable the BFD feature, see Enabling the BFD Feature, on page 58.
- For any client protocols that you want to enable BFD on, enable BFD in that client protocol, see Configuring BFD Support for Routing Protocols, on page 65.
- See other detailed prerequisites that are listed with the configuration tasks.

Guidelines and Limitations

- BFD supports BFD version 1.
- BFD supports IPv4.
- BFD supports single-hop BFD.
- BFD supports the Border Gateway Protocol (BGP).
- BFD for BGP supports single-hop External Border Gateway Protocol (EBGP) and internal Border Gateway Protocol (iBGP) peers.
- BFD supports the Enhanced Interior Gateway Routing Protocol (EIGRP).
- BFD supports the Open Shortest Path First (OSPF) routing protocol.
- BFD supports keyed SHA-1 authentication.
- BFD supports the following Layer 3 interfaces: physical interfaces, port channels, subinterfaces, and VLAN interfaces.
- BFD does not support Anycast HSRP.
- BFD depends on 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.
- Port-channel configuration limitations are as follows:
    - Fabricpath BFD sessions are not supported on a port channel logical interface on any type of a line card.
    - For Layer 3 port channels used by BFD, you must enable Link Aggregation Control Protocol (LACP) on the port channel.

  - For Layer 2 port channels used by SVI sessions, you must enable LACP on the port channel.

- BFD is supported on SVI interfaces that are formed over virtual port channels (vPCs), vPC peer-links and FabricPath links.
- SVI limitations are as follows:
  - An ASIC reset causes traffic disruption for other ports and could possibly cause SVI sessions on other ports to flap.
  - When you change the topology (for example, when you add or delete a link into a VLAN or delete a member from a Layer 2 port channel), the SVI session could be affected. The SVI session might go down first and then come up after the topology discovery is finished.

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

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

BFD Default Settings

The following table lists the default settings for BFD parameters.

Parameter	Default
BFD feature	Disabled
Required minimum receive interval	
Desired minimum transmit interval	
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

BFD Configuration Hierarchy

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

Procedure

Step 1 Enable the BFD feature.

Step 2 Configure global BFD parameters or configure BFD on an interface.

Step 3 Configure BFD support for routing protocols.

Enabling the BFD Feature

Before you begin

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

Ensure that you are in the correct VRF when you are about to change the BFD configuration at the protocol global level (for example, for OSPF or BGP).

Procedure

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

Example

This example shows how to enable BFD:

```
switch# configure terminal
switch(config)# feature bfd
switch(config)# show feature | include bfd
switch(config)# copy running-config startup-config
```

Configuring Global BFD Parameters

Before you begin

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 Configuring BFD on an Interface, on page 60 to override these global session parameters on an interface.

Ensure that you are in the correct VRF.

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **bfd** [**fabricpath**] **interval** *mintx* **min_rx** *msec* **multiplier** *value*	Configures the BFD session parameters for all BFD sessions on the device. You can override these values by configuring the BFD session parameters on an interface. Use the optional **fabricpath** keyword to configure the global parameters for fabricpath BFD sessions. The *mintx* and **min_rx** *msec* range is from 250 to 999 milliseconds, and the default is 250. The **multiplier** *value* range is from 3 to 50. The multiplier default is 3.
Step 3	switch(config)# **bfd** [**fabricpath**] **slow-timer** [*milliseconds*]	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.

	Command or Action	Purpose
Step 4	switch(config)# **bfd echo-interface loopback** *interface number*	Configures the interface used for 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**	Displays the BFD running configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure global BFD parameters:

```
switch# configure terminal
switch(config)# bfd interval 250 min_rx 250 multiplier 3
switch(config)# bfd slow-timer 2000
switch(config)# bfd echo-interface loopback 1 3
switch(config-if)# show running-config bfd
switch(config-if)# copy running-config startup-config
```

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.

Note You can use **bfd** [**fabricpath**] commands on an interface in fabricpath mode. The interface should be configured with **switchport mode fabricpath** to enable fabricpath mode.

Before you begin

Ensure that you are in the correct VRF.

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters 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** [**fabricpath**]	Enables fabricpath BFD at the interface level.
Step 4	switch(config-if)# **bfd** [**fabricpath**] **interval** *mintx* **min_rx** *msec* **multiplier** *value*	Configures the BFD session parameters for all BFD sessions on the interface. These values override the global BFD session parameters. Use the optional **fabricpath** keyword to configure the global parameters for fabricpath BFD sessions. The *mintx* and *msec* range is from 50 to 999 milliseconds and the default is 250. The multiplier range is from 3 to 50. The multiplier default is 3.
Step 5	(Optional) switch(config-if)# **bfd** [**fabricpath**] **authentication keyed-SHA1 key-id** *id* {**hex-key** *hex-key* | **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, which allows the use of multiple active keys. To disable SHA-1 authentication on the interface, use the **no** form of this 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**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure BFD on an interface:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# bfd interval 250 min_rx 250 multiplier 3
switch(config-if)# bfd authentication keyed-SHA1 key-id 1 key cisco123
switch(config-if)# show running-config bfd
switch(config-if)# copy running-config startup-config
```

Configuring BFD on a Port Channel

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

If you use per-link mode 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.

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 Optimizing BFD on Subinterfaces, on page 64 for more information.

Before you begin

Ensure that you are in the correct VRF.

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

Enable the BFD feature. See “Enabling the BFD Feature.”

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters 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	(Optional) switch(config-if)# **bfd** [**fabricpath**] **authentication keyed-SHA1 key-id** *id* {**hex-key** *hex-key* | **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, which allows the use of multiple active keys. To disable SHA-1 authentication on the interface, use the **no** form of this 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**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure BFD on a port channel:

```
switch# configure terminal
switch(config)# interface port-channel 2
switch(config-if)# bfd interval 250 min_rx 250 multiplier 3
switch(config-if)# bfd authentication keyed-SHA1 key-id 1 key cisco123
switch(config-if)# show running-config bfd
switch(config-if)# copy running-config startup-config
```

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 set to zero if the echo function is disabled. The slow timer becomes the required minimum receive interval if the echo function is enabled.

You can configure a fabricpath (L2) BFD session on any port belonging to any VLAN provided the default VLAN (VLAN 1) is configured in fabricpath mode. It is a requirement that the default VLAN 1 is in fabricpath mode for fabricpath BFD sessions to come up.

Note Echo mode is not supported on fabricpath interfaces.

Before you begin

Ensure that you are in the correct VRF.

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Configure the BFD session parameters. See Configuring Global BFD Parameters, on page 59 or Configuring BFD on an Interface, on page 60.

Ensure that ICMP redirect messages are disabled on BFD-enabled interfaces. Use the **no ip redirects** command or the **no ipv6 redirects** command on the interface.

Ensure that the IP packet verification check for identical IP source and destination addresses is disabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **bfd** [**fabricpath**] **slow-timer**[*milliseconds*]	Configures the slow timer used in the echo function. This value determines how fast BFD starts up a new session and is used to slow down the asynchronous sessions when the BFD echo function is enabled. This value overwrites the required minimum receive interval when the echo function is enabled. The range is from 1000 to 30000 milliseconds. The default is 2000.
Step 3	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.
Step 5	(Optional) switch(config-if)# **show running-config bfd**	Displays the BFD running configuration.

	Command or Action	Purpose
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the BFD Echo Function.

```
switch# configure terminal
switch(config)# bfd slow-timer 2000
switch(config)# interface ethernet 2/1
switch(config-if)# bfd echo
switch(config-if)# show running-config bgp
switch(config-if)# copy running-config startup-config
```

Optimizing BFD on Subinterfaces

You can optimize BFD on subinterfaces. BFD creates sessions for all configured subinterfaces. BFD sets the subinterface with the lowest configured VLAN ID 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.

Before you begin

Ensure that you are in the correct VRF.

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Configure the BFD session parameters. See Configuring Global BFD Parameters, on page 59 or Configuring BFD on an Interface, on page 60.

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

Example

This example shows how to optimize BFD on an interface:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# bfd optimize subinterface
switch(config-if)# show running-config bfd
switch(config-if)# copy running-config startup-config
```

Configuring BFD Support for Routing Protocols

- Border Gateway Protocol (BGP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Open Shortest Path First Version 2 (OSPFv2)
- Hot Standby Router Protocol (HSRP)
- Virtual Router Redundancy Protocol (VRRP)
- Static routes
- Protocol-Independent Multicast (PIM)
- FabricPath on Intermediate System to Intermediate System (IS-IS)

Configuring BFD on BGP

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

Before you begin

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Configure the BFD session parameters. See Configuring Global BFD Parameters, on page 59 or Configuring BFD on an Interface, on page 60.

Enable the BGP feature.

Procedure

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

	Command or Action	Purpose
Step 3	switch(config-router)# **neighbor** {*ip-address*} **remote-as** *as-number*	Configures the IPv4 address and AS number for a remote BGP peer. The *ip-address* format is x.x.x.x.
Step 4	switch(config-router-neighbor)# **bfd**	Enables BFD for this BGP peer.
Step 5	switch(config-router-neighbor)# **update-source** *interface-type number*	Brings up BFD for this BGP peer.
Step 6	(Optional) switch(config-router-neighbor)# **show running-config bgp**	Displays the BGP running configuration.
Step 7	(Optional) switch(config-router-neighbor)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure BFD on BGP:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# neighbor 209.165.201.1 remote-as 64497
switch(config-router-neighbor)# bfd
switch(config-router-neighbor)# update-source ethernet 2/1
switch(config-router-neighbor)# show running-config bgp
switch(config-router-neighbor)# copy running-config startup-config
```

Configuring BFD on EIGRP

Before you begin

Ensure that you are in the correct VRF.

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Configure the BFD session parameters. See Configuring Global BFD Parameters, on page 59 or Configuring BFD on an Interface, on page 60.

Enable the EIGRP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters 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

	Command or Action	Purpose
		autonomous-system command to configure the AS number explicitly, or this EIGRP instance remains in the shutdown state.
Step 3	(Optional) switch(config-router)# **bfd**	Enables BFD for this EIGRP router.
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 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*] [**interface** *if*]	Displays the EIGRP running configuration.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure BFD on EIGRP:

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# bfd
switch(config-router)# interface ethernet 2/1
switch(config-if)# ip eigrp Test1 bfd
switch(config-if)# show ip eigrp
switch(config-if)# copy running-config startup-config
```

Configuring BFD on OSPF

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

Before you begin

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Configure the BFD session parameters. See Configuring Global BFD Parameters, on page 59 or Configuring BFD on an Interface, on page 60.

Enable the OSPF feature.

Procedure

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

	Command or Action	Purpose
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 this OSPFv2 instance. Each OSPFv2 interface must also be enabled or disabled.
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*] [**interface** *if*]	Displays the OSPF running configuration.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure BFD on OSPF:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# bfd
switch(config-router)# interface ethernet 2/1
switch(config-if)# ip ospf bfd
switch(config-if)# show ip ospf
switch(config-if)# copy running-config startup-config
```

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 timer expiry and takes over as the active HSRP router. The **show hsrp detail** command shows this event as BFD@Act-down or BFD@Sby-down.

Before you begin

Ensure that you are in the correct VRF.

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Configure the BFD session parameters. See Configuring Global BFD Parameters, on page 59 or Configuring BFD on an Interface, on page 60.

Enable the HSRP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **hsrp bfd all-interfaces**	Enables or disables BFD on all HSRP interfaces. The default is disabled.
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**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure BFD on HSRP:

```
switch# configure terminal
switch(config)# hsrp bfd all-interfaces
switch(config)# interface ethernet 2/1
switch(config-if)# hsrp bfd
switch(config-if)# show running-config hsrp
switch(config-if)# copy running-config startup-config
```

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 timer expiry and takes over as the active VRRP router.

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

Before you begin

Ensure that you are in the correct VRF.

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Configure the BFD session parameters. See Configuring Global BFD Parameters, on page 59 or Configuring BFD on an Interface, on page 60.

Enable the VRRP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters 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)# **vrrp bfd** *address*	Enables or disables BFD on an VRRP interface. The default is disabled.
Step 5	(Optional) switch(config-if-vrrp)# **show running-config vrrp**	Displays the VRRP running configuration.
Step 6	(Optional) switch(config-if-vrrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure BFD on VRRP:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# vrrp 1
switch(config-if-vrrp)# vrrp bfd 10.0.0.10
switch(config-if-vrrp)# show running-config vrrp
switch(config-if-vrrp)# copy running-config startup-config
```

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

Ensure that you are in the correct VRF.

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Configure the BFD session parameters. See Configuring Global BFD Parameters, on page 59 or Configuring BFD on an Interface, on page 60.

Enable the HSRP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters 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** {*network-address* | *network-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	Saves the configuration persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure BFD on static routes:

```
switch# configure terminal
switch(config)# vrf context Red
switch(config-vrf)# ip route 1.1.1.0/24 ethernet 2/1 192.0.2.1
switch(config-vrf)# ip route static bfd ethernet 2/1 192.0.2.4
switch(config-vrf)# show ip route static vrf Red
switch(config-vrf)# copy running-config startup-config
```

Configuring BFD on PIM

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

Before you begin

Ensure that you are in the correct VRF.

Enable the BFD feature. See Enabling the BFD Feature, on page 58.

Configure the BFD session parameters. See Configuring Global BFD Parameters, on page 59 or Configuring BFD on an Interface, on page 60.

Enable the PIM feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip pim bfd**	Enables BFD for PIM.
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)# **ip pim bfd-instance** [**disable**]	Enables or disables BFD on a PIM interface. The default is disabled.

	Command or Action	Purpose
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**	Saves the configuration persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure BFD on PIM:

```
switch# configure terminal
switch(config)# ip pim bfd
switch(config)# interface ethernet 2/1
switch(config-if)# ip pim bfd-instance
switch(config-if)# show running-config pim
switch(config-if)# copy running-config startup-config
```

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:

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **ip eigrp** *instance-tag* **bfd disable** **Example:** `switch(config)# ip eigrp Test1 bfd disable`	Disables BFD on an EIGRP interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 2	switch(config-if)# **ip ospf bfd disable** **Example:** `switch(config-if)# ip ospf bfd disable`	Disables BFD on an OSPFv2 interface.

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 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
```

Verifying the BFD Configuration

Use the following commands to verify BFD:

Command	Purpose
switch# **show running-config bfd**	Displays information about BFD for a supported application, such as BGP or OSPFv2.
switch# **show startup-config bfd**	Displays information about the BFD configuration that will be applied on startup.

Monitoring BFD

Use the following commands to monitor BFD:

Command	Purpose
switch# **show bfd neighbors** [**application** *name*] [**details**]	Displays information about BFD for a supported application, such as BGP or OSPFv2.
switch# **show bfd neighbors** [**interface** *int-if*] [**details**]	Displays information about BFD on a specified interface.
switch# **show bfd neighbors** [**dest-ip** *ip-address*] [**src-ip** *ip-address*] [**details**]	Displays information about BFD on the specified session on an interface.
switch# **show bfd neighbors** [**vrf** *vrf-name*] [**details**]	Displays information about BFD for a VRF.
switch# **show bfd neighbors** [**fabricpath**] [**dest-ip** *ip-address*] [**src-ip** *ip-address*] [**details**]	Displays information about BFD on the specified session on an interface. Use the optional **fabricpath** keyword to display information about the specific fabricpath neighbor.
switch# **show bfd neighbors** [**dest-sys-id***dest-sys-id-value*]	Displays information about a specific FabricPath neighbor.

Configuration Examples for BFD

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

```
switch# configure terminal
switch(config)# feature bfd
switch(config)# feature ospf
switch(config)# router ospf Test1
switch(config-router)# interface ethernet 2/1
switch(config-if)# ip ospf bfd
switch(config-if)# no shutdown
```

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

```
switch# configure terminal
switch(config)# feature bfd
switch(config)# feature eigrp
switch(config)# bfd interval 250 min_rx 250 multiplier 4
switch(config)# router eigrp Test2
switch(config-router)# bfd
```

Related Documents

Related Topic	Document Title
BFD commands	

Related Standards

These industry standards cover Bidirectional Forwarding Detection.

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

CHAPTER 4

Configuring Port Channels

This chapter contains the following sections:

- Information About Port Channels, on page 77
- Configuring Port Channels, on page 85
- Verifying Port Channel Configuration, on page 98
- Verifying the Load-Balancing Outgoing Port ID , on page 99
- Feature History for Configuring Port Channels, on page 100

Information About Port Channels

A port channel bundles individual interfaces into a group to provide increased bandwidth and redundancy. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational as long as at least one physical interface within the port channel is operational.

You create a port channel by bundling compatible interfaces. You can configure and run either static port channels or port channels running the Link Aggregation Control Protocol (LACP).

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, Cisco NX-OS applies those parameters to each interface in the port channel.

You can use static port channels, with no associated protocol, for a simplified configuration. For more efficient use of the port channel, 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.

Related Topics

LACP Overview, on page 81

Understanding Port Channels

Using port channels, Cisco NX-OS provides wider bandwidth, redundancy, and load balancing across the channels.

You can collect ports into a static port channel or you can enable the Link Aggregation Control Protocol (LACP). Configuring port channels with LACP requires slightly different steps than configuring static port channels. For information on port channel configuration limits, see the *Verified Scalability* document for your platform. For more information about load balancing, see Load Balancing Using Port Channels, on page 80.

Note Cisco NX-OS does not support Port Aggregation Protocol (PAgP) for port channels.

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

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 operate in full-duplex mode. When you are running static port channels without LACP, the individual links are all in the on channel mode; you cannot change this mode without enabling LACP.

Note You cannot change the mode from ON to Active or from ON to Passive.

You can create a port channel 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, Cisco NX-OS creates a matching port channel automatically if the port channel does not already exist. You can also create the port channel first. In this instance, Cisco NX-OS creates an empty channel group with the same channel number as the port channel and takes the default configuration.

Note A 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.

Compatibility Requirements

When you add an interface to a port channel group, Cisco NX-OS checks certain interface attributes to ensure that the interface is compatible with the channel group. Cisco NX-OS 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:

- Port mode
- Access VLAN
- Trunk native VLAN
- Allowed VLAN list
- Speed
- 802.3x flow control setting
- MTU
- Broadcast/Unicast/Multicast Storm Control setting
- Priority-Flow-Control
- Untagged CoS

Use the **show port-channel compatibility-parameters** command to see the full list of compatibility checks that Cisco NX-OS uses.

You can only add interfaces configured with the channel mode set to on to static port channels. You can also 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.

When the interface joins a port channel, the following individual parameters are replaced with the values on the port channel:

- Bandwidth
- MAC address
- Spanning Tree Protocol

The following interface parameters remain unaffected when the interface joins a port channel:

- Description
- CDP
- LACP port priority
- Debounce

After you enable forcing a port to be added to a channel group by entering the **channel-group force** command, the following two conditions occur:

- When an interface joins a port channel, the following parameters are removed and they are operationally replaced with the values on the port channel; however, this change will not be reflected in the running configuration for the interface:
    - QoS
    - Bandwidth
    - Delay
    - STP
    - Service policy
    - ACLs
- When an interface joins or leaves a port channel, the following parameters remain unaffected:
    - Beacon
    - Description
    - CDP
    - LACP port priority
    - Debounce
    - UDLD
    - Shutdown

- SNMP traps

Load Balancing Using Port Channels

Cisco NX-OS load balances traffic across all operational interfaces in a port channel by reducing part of the binary pattern formed from 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.

For a HIF port-channel, same hashing parameters are used by the FEX as a Cisco Nexus 5000 or Cisco Nexus 6000 series parent. If the number of interfaces is a power of 2, then there might be some difference between the outgoing port on the switch and on the FEX. Use the **show port-channel load-balance forwarding-path interface** command to get the display of the outgoing port. This is applicable to N2K-C2232PP-10GE, N2K-C2232TM-10GE, N2K-C2232TM-E-10GE, N2K-C2348UPQ-10GE, N2K-C2248TP-1GE, N2K-C2224TP-1GE, N2K-C2248TP-E-1GE, N2K-C2248PQ-10GE, N2K-C2348TQ, N2K-C2332TQ-10GE, N2K-C2348TQ-E.

The basic configuration uses the following criteria to select the link:

- For a Layer 2 frame, it uses the source and destination MAC addresses.
- For a Layer 3 frame, it uses the source and destination MAC addresses and the source and destination IP addresses.
- For a Layer 4 frame, it uses the source and destination MAC addresses and the source and destination IP addresses.

Note You have the option to include the source and destination port number for the Layer 4 frame.

You can configure the switch to use one of the following methods (see the following table for more details) 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
- Destination TCP/UDP port number
- Source TCP/UDP port number
- Source and destination TCP/UDP port number

Table 3: Port Channel Load-Balancing Criteria

Configuration	Layer 2 Criteria	Layer 3 Criteria	Layer 4 Criteria
Destination MAC	Destination MAC	Destination MAC	Destination MAC
Source MAC	Source MAC	Source MAC	Source MAC
Source and destination MAC	Source and destination MAC	Source and destination MAC	Source and destination MAC
Destination IP	Destination MAC	Destination MAC, destination IP	Destination MAC, destination IP
Source IP	Source MAC	Source MAC, source IP	Source MAC, source IP
Source and destination IP	Source and destination MAC	Source and destination MAC, source and destination IP	Source and destination MAC, source and destination IP
Destination TCP/UDP port	Destination MAC	Destination MAC, destination IP	Destination MAC, destination IP, destination port
Source TCP/UDP port	Source MAC	Source MAC, source IP	Source MAC, source IP, source port
Source and destination TCP/UDP port	Source and destination MAC	Source and destination MAC, source and destination IP	Source and destination MAC, source and destination IP, source and destination port

Use the option that provides the balance criteria with the greatest variety in your configuration. For example, if the traffic on a port channel is going only to a single MAC address and you use the destination MAC address as the basis of port-channel load balancing, the port channel always chooses the same link in that port channel; using source addresses or IP addresses might result in better load balancing.

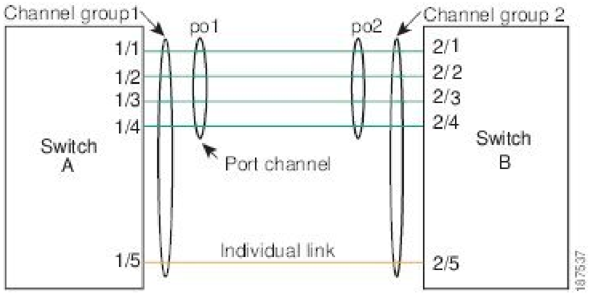
Understanding LACP

LACP Overview

Note You must enable the LACP feature before you can configure and use LACP functions.

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

Figure 8: Individual Links Combined into a Port Channel



With LACP, just like with static port channels, you can bundle up to 16 interfaces in a channel group.

Note When you delete the port channel, Cisco NX-OS automatically deletes the associated channel group. All member interfaces revert to their previous configuration.

You cannot disable LACP while any LACP configurations are present.

LACP ID Parameters

LACP uses the following parameters:

- LACP system priority—Each system that runs LACP has an LACP system priority value. You can accept the default value of 32768 for this parameter, or you can configure a value between 1 and 65535. LACP uses the system priority with the MAC address to form the system ID and also uses the system priority during negotiation with other devices. A higher system priority value means a lower priority.

Note The LACP system ID is the combination of the LACP system priority value and the MAC address.

- LACP port priority—Each port 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, the duplex capability, and the point-to-point or shared medium state

    - Configuration restrictions that you establish

Channel Modes

Individual interfaces in port channels are configured with channel modes. When you run static port channels, with no 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.

Note You must enable LACP globally before you can configure an interface in either the active or passive channel mode.

The following table describes the channel modes.

Table 4: Channel Modes for Individual Links in a Port Channel

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 is, 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 returns 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.

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 as long as the modes are compatible as in the following 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.

LACP Marker Responders

Using port channels, data traffic may be dynamically redistributed due to either a link failure or load balancing. LACP uses the Marker Protocol to ensure that frames are not duplicated or reordered because of this redistribution. Cisco NX-OS supports only Marker Responders.

LACP-Enabled and Static Port Channel Differences

The following table provides a brief summary of major differences between port channels with LACP enabled and static port channels. For information about the maximum configuration limits, see the *Verified Scalability* document for your device.

Table 5: Port Channels with LACP Enabled and Static Port Channels

Configurations	Port Channels with LACP Enabled	Static Port Channels
Protocol applied	Enable globally.	Not applicable.
Channel mode of links	Can be either: • Active • Passive	Can only be On.

LACP Port-Channel MinLinks and MaxBundle

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

The introduction of the minlinks and maxbundle feature further refines LACP port-channel operation and provides increased bandwidth in one manageable interface.

The LACP port-channel MinLink 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 minlink 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.

The LACP minlink and maxbundle configurations are not applicable to FEX HIF port-channels.

The minlink/maxbundle configuration can be applied to a non-LACP HIF PO, but the feature is not operational. Configuring a minlink/maxbundle configuration on a LACP HIF PO is not allowed and is rejected.

Configuring Port Channels

Default Settings

Table 6: Default Port-Channel Parameters

Parameters	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	32678
Minimum links for LACP	1
Minimum links for FEX fabric port channel	1
Maxbundle	16

LACP Port-Channel Min Links

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

The introduction of the minimum links 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.

Note The minimum links 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.

Creating a Port Channel

You can create a port channel before creating a channel group. Cisco NX-OS automatically creates the associated channel group.

Note If you want LACP-based port channels, you need to 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. The range is from 1 to 4096. Cisco NX-OS automatically creates the channel group if it does not already exist.
Step 3	switch(config)# **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
```

Adding a Port to a Port Channel

You can add a port to a new channel group or to a channel group that already contains ports. Cisco NX-OS creates the port channel associated with this channel group if the port channel does not already exist.

Note If you want LACP-based port channels, you need to enable LACP.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Specifies the interface that you want to add to a channel group and enters the interface configuration mode.

	Command or Action	Purpose
Step 3	(Optional) switch(config-if)# **switchport mode trunk**	Configures the interface as a trunk port.
Step 4	(Optional) switch(config-if)# **switchport trunk** {**allowed vlan** *vlan-id* | **native vlan** *vlan-id*}	Configures necessary parameters for a trunk port.
Step 5	switch(config-if)# **channel-group** *channel-number*	Configures the port in a channel group and sets the mode. The channel-number range is from 1 to 4096. Cisco NX-OS creates the port channel associated with this channel group if the port channel does not already exist. This is called implicit port channel creation.
Step 6	(Optional) switch(config-if)# **no channel-group**	Removes the port from the channel group. The port reverts to its original configuration.

Example

This example shows how to add an Ethernet interface 1/4 to channel group 1:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# switchport mode trunk
switch(config-if)# channel-group 1
```

Configuring Load Balancing Using Port Channels

You can configure the load-balancing algorithm for port channels that applies to the entire device.

Note If you want LACP-based port channels, you need to enable LACP.

Note For load-balancing FC traffic across SAN PO members in Nexus 5672UP-16G switch, the **port-channel load-balance ethernet** command is not needed. The load-balancing happens by default.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-channel load-balance ethernet** {[**destination-ip** | **destination-mac** | **destination-port** | **source-dest-ip** | **source-dest-mac** | **source-dest-port** | **source-ip** | **source-mac** | **source-port**] | **crc-poly**}	Specifies the load-balancing algorithm for the device. The range depends on the device. The default is **source-dest-ip**.

	Command or Action	Purpose
		Note • source-dest-ip-only • source-dest-port-only • source-dest-ip • source-dest-port • source-dest-ip-gre The Cisco Nexus 5500 Platform switches support 8 hash polynomials that can be used for compression on the hash-parameters. Depending on variations in the hash parameters for egress traffic flows from a port channel, different polynomials could provide different load distribution results. The default hash polynomial is CRC8a. The variable can be configured as follows: • CRC8a • CRC8b • CRC8c • CRC8d • CRC8e • CRC8f • CRC8g
Step 3	(Optional) switch(config)# **no port-channel load-balance ethernet**	Restores the default load-balancing algorithm of source-dest-ip.
Step 4	(Optional) switch# **show port-channel load-balance**	Displays the port-channel load-balancing algorithm.

Example

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

```
switch# configure terminal
switch (config)# port-channel load-balance ethernet source-ip
```

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 facilitates grouping the links into an port channel. The port channel is then added to the spanning tree as a single bridge port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature lacp**	Enables LACP on the switch.
Step 3	(Optional) switch(config)# **show feature**	Displays enabled features.

Example

This example shows how to enable LACP:

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

Configuring the Channel Mode for a Port

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 protocol, all interfaces on both sides of the link remain in the on channel mode.

Before you begin

Ensure that you have enabled the LACP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **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.

	Command or Action	Purpose
		force—Specifies that the LAN port be forcefully added to the channel group. **mode**—Specifies the port channel mode of the interface. **active**—Specifies that when you enable LACP, this command enables LACP on the specified interface. The interface is in an active negotiating state in which the port initiates negotiations with other ports by sending LACP packets. **on**—(Default mode) Specifies that all port channels that are not running LACP remain in this mode. **passive**—Enables LACP only if an LACP device is detected. The interface is in a passive negotiation state in which the port responds to LACP packets that it receives but does not initiate LACP negotiation. When you run port channels with no associated protocol, the channel mode is always on.
Step 4	switch(config-if)# **no channel-group** *number* **mode**	Returns the port mode to on for the specified interface.

Example

This example shows how to set the LACP-enabled interface to 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
```

This example shows how to forcefully add an interface to the channel group 5:

```
switch(config)# interface ethernet 1/1
switch(config-if)# channel-group 5 force
switch(config-if)#
```

Configuring LACP Port-Channel Minimum Links

You can configure the LACP minimum links feature. Although minimum links work only in LACP, you can enter the CLI commands for this feature 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** *number*	Specifies the 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 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. **Note** When you upgrade from an earlier Cisco NX-OS release to Cisco NX-OS release 7.2(1)N1(1) or a later release, then an additional configuration line, **no lacp suspend-individual**, is displayed in the **show** command output of the **show running-config interface port-channel** *number* command.

Example

The following example shows how to configure LACP port-channel minimum links.

```
switch# configure terminal
switch(config)# interface port-channel 3
switch(config-if)# lacp min-links 3
switch(config-if)# show running-config interface port-channel 3
interface port-channel 3
lacp min-links 3
```

Configuring the LACP Port-Channel MaxBundle

You can configure the LACP minlinks feature. Although minlinks and maxbundles work only in LACP, you can enter the CLI commands for these features for non-LACP port channels, but these commands are nonoperational.

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)# **lacp max-bundle** *number*	Specifies the port-channel interface to configure max-bundle, and enters the interface configuration mode. The default value for the port-channel max-bundle is 16. The allowed range is from 1 to 16. **Note** Even if the default value is 16, the number of active members in a port channel is the minimum of the pc_max_links_config and pc_max_active_members that is allowed in the port channel. The min-links and max-bundle features work well when both ends of the port-channel are configured with the same value. Mismatched min/max-bundle values might result in port flap, traffic drops, and unnecessary port suspension.
Step 4	switch(config-if)# **show running config interface port-channel** *number*	

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.

Before you begin

Ensure that you have enabled the LACP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure and enters the interface configuration mode.
Step 3	switch(config-if)# **lacp rate fast**	Configures the fast rate (one second) at which LACP control packets are sent to an LACP-supported interface.

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 Short-timeout

You can change the LACP Short-timeout value for the lacp rate fast command to modify the duration of the LACP Fast Rate timeout. Use the **lacp rate fast** command to set the fast rate at which LACP control packets are sent to an LACP-supported interface. You can change the short timeout value from 15 seconds (default timeout) to 3 seconds (IEEE802.3ad recommended standard).

Note The **lacp short-timeout** command is supported only on LACP-enabled interfaces.

BFD feature is not supported, when LACP Short-timeout is configured.

LACP rate fast is not recommended on vPC multichassis Etherchannel trunk (MCT) ports.

Before you begin

Ensure that you have enabled the LACP Rate Fast feature.

Procedure

Step 1 switch# **configure terminal**

Enters global configuration mode.

Step 2 switch(config)# **lacp short-timeout**

Configures the short-timeout value for LACP fast rate at which LACP control packets are sent to an LACP-supported interface. Valid range value is from 3 to 15 seconds. The default short-timeout value is 15 seconds.

Note The LACP Short-timeout default configuration is not displayed in the running configuration.

Step 3 (Optional) switch(config)# **show run | include lacp**

Displays the LACP configuration on an interface.

Example

This example shows how to configure the LACP short-timeout value of 3 seconds for LACP fast rate:

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

This example shows how to restore the LACP short-timeout default timeout (15 seconds) for LACP fast rate:

```
switch# configure terminal
switch(config)# no lacp short-timeout
```

Procedure to revert when BFD Feature and LACP Short-timeout are configured Together

Do not enable BFD feature along with the LACP Short-timeout configuration. This configuration is not supported. You must wait for 30 seconds after configuring **lacp** *short-timeout*. The command prompts error when you attempt to enable BFD feature.

If you have enabled the BFD feature along with the LACP short-timeout, you must do the following procedure to exit this configuration:

Procedure

Step 1 Use the **no lacp short-timeout** command to disable the lacp short-timeout.

Step 2 Use the **no feature BFD** command to disable the feature BFD.

Configuring the LACP System Priority and System ID

The LACP system ID is the combination of the LACP system priority value and the MAC address.

Before you begin

Ensure that you have enabled the LACP feature.

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 1 through 65535, and higher numbers have lower priority. The default value is 32768.
Step 3	(Optional) switch# **show lacp system-identifier**	Displays the LACP system identifier.

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

You can configure each link in the LACP port channel for the port priority.

Before you begin

Ensure that you have enabled the LACP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **lacp port-priority** *priority*	Configures the port priority for use with LACP. Valid values are 1 through 65535, and higher numbers have lower priority. The default value is 32768.

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

Before you begin

- Enable the LACP feature.
- Confirm that the port channel is in the administratively down state.
- Ensure that you are in the correct VDC. To switch to the correct VDC, enter the **switchto vdc** command.

Procedure

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config) #`
		Specifies the port channel interface to configure, and enters interface configuration mode.

Step 3	Required: **shutdown**
	Example:
	`switch(config-if)# shutdown`
	`switch(config-if) #`
		Administratively shuts down the port channel.

Step 4	Required: **no lacp graceful-convergence**
	Example:
	`switch(config-if)# no lacp graceful-convergence`
	`switch(config-if) #`
		Disables LACP graceful convergence on the specified port channel.

Step 5	Required: **no shutdown**
	Example:
	`switch(config-if)# no shutdown`
	`switch(config-if) #`
		Administratively brings the port channel up.

Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example disables 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
switch(config-if) #
```

Reenabling LACP Graceful Convergence

Before you begin

- Enable the LACP feature.
- Confirm that the port channel is in the administratively down state.
- Ensure that you are in the correct VDC. To switch to the correct VDC, enter the **switchto vdc** command.

Procedure

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config) #`
		Specifies the port channel interface to configure, and enters interface configuration mode.

Step 3	Required: **shutdown**
	Example:
	`switch(config-if)# shutdown`
	`switch(config-if) #`
		Administratively shuts down the port channel.

Step 4	Required: **lacp graceful-convergence**
	Example:
	`switch(config-if)# lacp graceful-convergence`
	`switch(config-if) #`
		Enables LACP graceful convergence on the specified port channel.

Step 5	Required: **no shutdown**
	Example:
	`switch(config-if)# no shutdown`
	`switch(config-if) #`
		Administratively brings the port channel up.

	Command or Action	Purpose
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example disables 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
switch(config-if) #
```

Verifying Port Channel Configuration

Use the following command to verify the port channel configuration information:

Command	Purpose
show interface port channel*channel-number*	Displays the status of a port channel interface.
show feature	Displays enabled features.
show resource	Displays the number of resources currently available in the system.
show lacp {**counters** | **interface** *type slot*/*port* | **neighbor** | **port-channel** | **system-identifier**}	Displays LACP information.
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 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 port-channel database	Displays information on current running of the port channel feature.

Command	Purpose
show port-channel load-balance	Displays information about load-balancing using port channels. Based on the system—Source and destination IP, and MAC addresses are used for port channel load balance calculation. By default the **source-destination-ip** address is used. Based on protocol—For non-IP traffic, only the source and destination MAC address is used, and for IP traffic, both source and destination IP and MAC addresses are used for the port channel load balance calculation.

Verifying the Load-Balancing Outgoing Port ID

Command Guidelines

The **show port-channel load-balance** command allows you to verify which ports a given frame is hashed to on a port channel. You need to specify the VLAN and the destination MAC in order to get accurate results.

Note Certain traffic flows are not subject to hashing such as when there is a single port in a port-channel.

To display the load-balancing outgoing port ID, perform one of the tasks:

Command	Purpose
switch# **show port-channel load-balance forwarding-path interface port-channel** *port-channel-id* **vlan** vlan-id **dst-ip src-ip dst-mac src-mac l4-src-port** *port-id* **l4-dst-port** *port-id*	Displays the outgoing port ID.

Example

This example shows how to display the load balancing outgoing port ID:

```
switch# show port-channel load-balance forwarding-path interface port-channel 10 vlan 1
dst-ip 1.225.225.225 src-ip 1.1.10.10 src-mac aa:bb:cc:dd:ee:ff
l4-src-port 0 l4-dst-port 1
Missing params will be substituted by 0's. Load-balance Algorithm on switch: source-dest-port
 crc8_hash:204 Outgoing port id: Ethernet 1/1 Param(s) used to calculate load balance:
dst-port: 0
src-port: 0
dst-ip: 1.225.225.225
src-ip: 1.1.10.10
dst-mac: 0000.0000.0000
src-mac: aabb.ccdd.eeff
```

Feature History for Configuring Port Channels

Table 7: Feature History for Configuring Port Channels

Feature Name	Release	Feature Information
Min Links	6.0(2)N1(2)	This feature was introduced.
LACP Short-timeout	7.3(0)N1(1)	This feature was introduced.

CHAPTER 5

Configuring Enhanced Virtual Port Channels

This chapter contains the following sections:

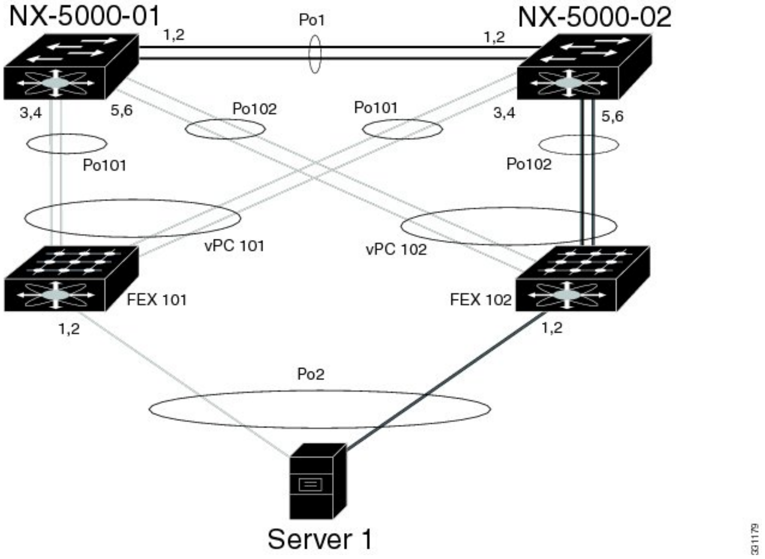
- Information About Enhanced vPCs, on page 101
- Licensing Requirements for Enhanced vPC, on page 103
- Configuring Enhanced vPCs, on page 104
- Verifying Enhanced vPCs, on page 104
- Enhanced vPC Example Configuration, on page 108

Information About Enhanced vPCs

Enhanced Virtual Port Channels Overview

The virtual port channel (vPC) feature allows the dual homed connection of a host to two fabric extenders (FEXs) or a dual homed connection of a FEX to two switches. The enhanced vPC feature, or two-layer vPC, allows both dual homing topologies to be combined simultaneously, as shown in the following figure:

Figure 9: Dual Homing Topology



With enhanced vPCs, all available paths from the hosts to the FEXs and from the FEXs to the switches are active and carry Ethernet traffic, maximizing the available bandwidth and providing redundancy at both levels.

Supported Platforms and Topologies

Supported Platforms

Enhanced vPC is supported on Cisco Nexus devices.

Any Cisco Nexus Fabric Extender can be used with Enhanced vPC.

Enhanced vPC is compatible with Layer 3 features on the switch.

Supported and Unsupported Topologies

Enhanced vPC supports the following topologies:

- A single homed server connected to a single FEX
- A dual homed server connected by a port channel to a single FEX
- A dual homed server connected by a port channel to a pair of FEXs

  This topology allows connection to any two FEXs that are connected to the same pair of switches in a vPC domain. Static port channel and Link Aggregation Control Protocol (LACP)-based port channel are supported.
- A dual homed server connected by Fibre Channel over Ethernet (FCoE) and port channel to a pair of FEXs
- A dual homed server connected by active/standby NIC teaming to a pair of FEXs

Enhanced vPC does not support the following topologies:

- A dual homed server connected to a pair of FEXs that connect to a single switch

  Although this topology becomes a functioning system when one switch has failed, it is not recommended in normal operation.
- A multi-homed server connected by a port channel to more than two FEXs

  This topology results in increased complexity with little benefit.
- You cannot have a link for non-vPC traffic in parallel with a vPC topology. This can cause errors with the traffic forwarding logic resulting in duplicate or missed packets.

Enhanced vPC Scalability

The scalability of enhanced vPC is similar to that of the dual homed FEX topology.

Each Cisco Nexus device supports up to 24 FEXs with Layer 2 configuration or Layer 3 configuration. In a dual homed FEX topology, such as that in enhanced vPC, each FEX is managed by two switches, so the pair together can support 24 FEXs.

Enhanced vPC Failure Response

The enhanced vPC topology provides a high level of resilience to the failure of system components and links as described in the following scenarios:

- Failure of One or More Port Channel Member Links

  When one member link of a port channel fails, the traffic flow is moved to the remaining port channel member links. If all member links of a port channel fail, the traffic flow is redirected to the remaining port channel of the vPC.

- Failure of One FEX

  When one FEX fails, the traffic flow from all dual homed hosts is moved to the remaining FEX.

- Failure of One Switch

  When one switch fails, the traffic flow from all dual homed FEXs is moved to the remaining switch. Traffic from the hosts is unaffected.

- Failure of Both Uplinks from a Single FEX

  When both uplinks from one FEX fails, the FEX shuts down its host ports, and the traffic flow from all dual homed hosts is moved to the other FEX.

- Failure of the vPC Peer Link

  When the vPC secondary switch detects the failure of the peer link, it checks the status of the primary switch by the peer-keepalive link. If the primary switch is unresponsive, the secondary switch maintains all traffic flows as before. If the primary switch is active, the secondary switch shuts down its interfaces to the FEXs, and the traffic flow from all dual homed FEXs is moved to the primary switch. Ethernet traffic from the hosts is unaffected in either case.

  If the secondary switch carries FCoE traffic and shuts down its interfaces to the FEXs, it also shuts down all virtual Fibre Channel (vFC) interfaces that are bound to the FEX host ports. In this case, the hosts must use multipathing to move SAN traffic to the remaining vFC interface.

- Failure of the vPC Peer-Keepalive Link

  A failure of the vPC peer-keepalive link by itself does not affect the traffic flow.

Licensing Requirements for Enhanced vPC

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	This feature does not require a license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Configuring Enhanced vPCs

Overview of Configuration Steps for Enhanced vPC

An enhanced vPC configuration consists of a combination of two standard vPC configurations: the dual homed connection of a host to two FEXs and the dual homed connection of a FEX to two switches. The required configuration tasks are listed here, but the detailed procedures for those two standard configurations are presented in the "Configuring Virtual Port Channels" chapter of this document.

To configure enhanced vPC, perform the following steps.

Note

- In procedures where the configuration must be repeated on both switches, the configuration synchronization (config-sync) feature allows you to configure one switch and have the configuration automatically synchronized to the peer switch. For more information about configuration synchronization, see the *Operations Guide* for your device.
- You cannot configure non-vPC interfaces across host ports on two different FEXs.

Procedure

Step 1 Enable the vPC and LACP features on each switch.

Step 2 Create required VLANs on each switch.

Step 3 Assign a vPC domain ID and configure the vPC peer-keepalive link on each switch.

Step 4 Configure the vPC peer link on each switch.

Step 5 Configure port channels from the first FEX to each switch.

Step 6 Configure port channels from the second FEX to each switch.

Step 7 If the enhanced vPC must accommodate FCoE traffic, associate the first FEX to one switch, and then associate the second FEX to the other switch.

See “Configuring FCoE over Enhanced vPC” in the *Fibre Channel over Ethernet Configuration Guide* for your device.

Step 8 Configure a host port channel on each FEX.

Verifying Enhanced vPCs

Verifying the Enhanced vPC Configuration

Before bringing up a vPC, the two peer switches in the same vPC domain exchange configuration information to verify that both switches have compatible configurations for a vPC topology. Depending on the severity of the impact of possible mismatched configurations, some configuration parameters are considered as Type 1 consistency check parameters while others are considered as Type 2.

When a mismatch in Type 1 parameters is found, both peer switches suspend VLANs on the vPC ports. When a mismatch in Type 2 parameters is found, a warning syslog message is generated, but the vPC remains up and running.

Note Enhanced vPCs do not support the graceful consistency check.

For enhanced vPCs, the consistency verification for global configuration parameters is the same as for a dual homed FEX topology, and is described in the documentation for dual homed FEX. In addition to the global consistency verification, enhanced vPCs require interface level verification using tasks described in this section.

Use the following commands to verify the enhanced vPC configuration and consistency:

Command	Purpose
switch# **show feature**	Displays whether vPC is enabled.
switch# **show running-config vpc**	Displays running configuration information for vPCs.
switch# **show vpc brief**	Displays brief information on the vPCs.
switch(config)# **show vpc consistency-parameters global**	Displays the status of global vPC parameters that must be consistent across all vPC interfaces.
switch(config)# **show vpc consistency-parameters interface port-channel** *channel-number*	Displays the status of specific port channels that must be consistent across vPC devices.

For detailed information about the fields in the output of these commands, see the command reference for your device.

Verifying the Consistency of Port Channel Numbers

For enhanced vPCs, both switches must use the same port channel number for the dual homed connection to a FEX. If different port channel numbers are used, the port channel and its member ports are suspended on both switches.

Procedure

	Command or Action	Purpose
Step 1	**show running-config interface** *type/slot*[, *type/slot*[, ...]] **Example:** `switch-1# show running-config interface Ethernet110/1/1, Ethernet111/1/1`	Displays the configuration of the specified list of port channel member ports. Enter this command on both peer switches and compare the reported **channel-group** numbers to verify that they match between switches.
Step 2	**show interface** *type/slot* **Example:**	Displays the status and configuration of the specified port channel member port.

	Command or Action	Purpose
	`switch-1# show interface Ethernet110/1/1`	Enter this command on both peer switches and verify the status of the ports.

Example

This example shows how to verify the consistency of the port channel numbering between the two switches. In this example, the port channel numbering is inconsistent and the member ports are suspended:

```
switch-1# show running-config interface Ethernet110/1/1, Ethernet111/1/1

!Command: show running-config interface Ethernet110/1/1, Ethernet111/1/1
!Time: Sun Aug 28 03:38:23 2011

version 5.1(3)N1(1)

interface Ethernet110/1/1
channel-group 102

interface Ethernet111/1/1
channel-group 102

switch-2# show running-config interface Ethernet110/1/1, Ethernet111/1/1

!Command: show running-config interface Ethernet110/1/1, Ethernet111/1/1
!Time: Sun Aug 28 03:38:23 2011

version 5.1(3)N1(1)

interface Ethernet110/1/1
channel-group 101

interface Ethernet111/1/1
channel-group 101

switch-1# show interface Ethernet110/1/1
Ethernet110/1/1 is down (suspended by vpc)
  Hardware: 100/1000 Ethernet, address: 7081.0500.2402 (bia 7081.0500.2402)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
  [...]

switch-2# show interface Ethernet110/1/1
Ethernet110/1/1 is down (suspended by vpc)
  Hardware: 100/1000 Ethernet, address: 7081.0500.2402 (bia 7081.0500.2402)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
  [...]
```

Verifying Common Port Channel Members

The port channel from a FEX to the switch pair is up and operational when there is at least one common port channel member between the two switches. Any FEX interfaces that are assigned to the port channel only on one switch will be suspended.

Procedure

	Command or Action	Purpose
Step 1	**show port-channel summary** **Example:** `switch-1# show port-channel summary`	Displays a summary of the port channel interfaces.
Step 2	(Optional) **show interface** *type/slot* **Example:** `switch-1# show interface ethernet 111/1/3`	Displays the status and configuration of the specified interface.

Example

This example shows how to verify the common member ports of the vPC. In this example, the vPC is configured with one port channel member that is not common to both switches. That member port is shown as shut down, and further investigation shows that the member is suspended by the vPC. In this part of the session, the port channel is configured on each switch, with an extra port on the first switch:

```
switch-1(config)# interface ethernet 110/1/3, ethernet 111/1/3
switch-1(config-if)# channel-group 101
switch-1(config-if)# interface port-channel 101
switch-1(config-if)# switchport access vlan 20

switch-2(config)# interface ethernet 110/1/3
switch-2(config-if)# channel-group 101
switch-2(config-if)# interface port-channel 101
switch-2(config-if)# switchport access vlan 20
```

In this part of the session, the extra port is shown to be in the down state, and a display of the port details shows that the port is suspended by the vPC:

```
switch-1# 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
--------------------------------------------------------------------------------
1     Po1(SU)     Eth      LACP      Eth1/1(P)    Eth1/2(P)
[...]
101   Po101(SU)   Eth      NONE      Eth110/1/3(P)  Eth111/1/3(D)

switch-1# show interface ethernet 111/1/3
Ethernet111/1/3 is down (suspended by vpc)
  Hardware: 100/1000 Ethernet, address: 7081.0500.2582 (bia 7081.0500.2582)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
```

Verifying Interface Level Consistency for Enhanced vPCs

For enhanced vPCs, you must ensure consistency of the port mode and the shared VLAN in the port channel interface configuration.

Procedure

	Command or Action	Purpose

Step 1	**show vpc consistency-parameters port-channel** *channel-number*
	Example:
	`switch# show vpc consistency-parameters interface port-channel 101`
	`switch(config)#`
		For the specified port channel, displays the status information that must be consistent across vPC devices.

Example

This example shows how to display a comparison of the interface configuration across two peers for a vPC. In this case, VLAN 10 is allowed on both peers, but the port mode is mismatched, causing the VLAN to be suspended.

```
switch-1# show vpc consistency-parameters interface port-channel 101

    Legend:
        Type 1 : vPC will be suspended in case of mismatch

Name                        Type  Local Value            Peer Value
-------------               ----  ---------------------- -----------------------
mode                        1     on                     on
Speed                       1     1000 Mb/s              1000 Mb/s
Duplex                      1     full                   full
Port Mode                   1     access                 trunk
MTU                         1     1500                   1500
Admin port mode             1
Shut Lan                    1     No                     No
vPC+ Switch-id              1     3000                   3000
Allowed VLANs               -     10                     1-57,61-3967,4048-4093
Local suspended VLANs       -     10                     -
```

Enhanced vPC Example Configuration

The following example shows the complete configuration procedure using the topology of the enhanced vPC figure in this chapter. In the topology figure, the number pairs beside each port channel link represent the interface port numbers. For example, the switch link labeled with the numbers "3,4" represents interfaces eth1/3 and eth1/4 on the switch.

Note In procedures where the configuration must be repeated on both switches, the configuration synchronization (config-sync) feature allows you to configure one switch and have the configuration automatically synchronized to the peer switch. For more information about configuration synchronization, see the operations guide for your device.

Before you begin

Ensure that the Cisco Nexus Fabric Extenders FEX101 and FEX102 are attached and online.

Procedure

Step 1 Enable the vPC and LACP features on each switch.

Example:

```
switch-1(config)# feature vpc
switch-1(config)# feature lacp

switch-2(config)# feature vpc
switch-2(config)# feature lacp
```

Step 2 Create required VLANs on each switch.

Example:

```
switch-1(config)# vlan 10-20

switch-2(config)# vlan 10-20
```

Step 3 Assign a vPC domain ID and configure the vPC peer-keepalive link on each switch.

Example:

```
switch-1(config)# vpc domain 123
switch-1(config-vpc)# peer-keepalive destination 172.25.182.100

switch-2(config)# vpc domain 123
switch-2(config-vpc)# peer-keepalive destination 172.25.182.99
```

Note When you configure each switch, use the IP address of the peer switch as the peer-keepalive destination.

Step 4 Configure the vPC peer link on each switch.

Example:

```
switch-1(config)# interface eth1/1-2
switch-1(config-if)# channel-group 1 mode active
switch-1(config-if)# interface Po1
switch-1(config-if)# switchport mode trunk
switch-1(config-if)# switchport trunk allowed vlan 1, 10-20
switch-1(config-if)# vpc peer-link

switch-2(config)# interface eth1/1-2
```

```
switch-2(config-if)# channel-group 1 mode active
switch-2(config-if)# interface Po1
switch-2(config-if)# switchport mode trunk
switch-2(config-if)# switchport trunk allowed vlan 1, 10-20
switch-2(config-if)# vpc peer-link
```

Step 5 Configure port channels from the first FEX to each switch.

Example:

```
switch-1(config)# fex 101
switch-1(config-fex)# interface eth1/3-4
switch-1(config-if)# channel-group 101
switch-1(config-if)# interface po101
switch-1(config-if)# switchport mode fex-fabric
switch-1(config-if)# vpc 101
switch-1(config-if)# fex associate 101

switch-2(config)# fex 101
switch-2(config-fex)# interface eth1/3-4
switch-2(config-if)# channel-group 101
switch-2(config-if)# interface po101
switch-2(config-if)# switchport mode fex-fabric
switch-2(config-if)# vpc 101
switch-2(config-if)# fex associate 101
```

Step 6 Configure port channels from the second FEX to each switch.

Example:

```
switch-1(config)# fex 102
switch-1(config-fex)# interface eth1/5-6
switch-1(config-if)# channel-group 102
switch-1(config-if)# interface po102
switch-1(config-if)# switchport mode fex-fabric
switch-1(config-if)# vpc 102
switch-1(config-if)# fex associate 102

switch-2(config)# fex 102
switch-2(config-fex)# interface eth1/5-6
switch-2(config-if)# channel-group 102
switch-2(config-if)# interface po102
switch-2(config-if)# switchport mode fex-fabric
switch-2(config-if)# vpc 102
switch-2(config-if)# fex associate 102
```

Step 7 Configure a host port channel on each FEX.

Example:

```
switch-1(config)# interface eth101/1/1, eth101/1/2
switch-1(config-if)# channel-group 2 mode active
switch-1(config-if)# interface eth102/1/1, eth102/1/2
switch-1(config-if)# channel-group 2 mode active
switch-1(config-if)# int po2
switch-1(config-if)# switchport access vlan 10

switch-2(config)# interface eth101/1/1, eth101/1/2
switch-2(config-if)# channel-group 2 mode active
switch-2(config-if)# interface eth102/1/1, eth102/1/2
switch-2(config-if)# channel-group 2 mode active
```

```
switch-2(config-if)# int po2
switch-2(config-if)# switchport access vlan 10
```

CHAPTER 6

Configuring Virtual Port Channels

This chapter contains the following sections:

- Information About vPCs, on page 113
- Guidelines and Limitations for vPCs, on page 129
- Configuring vPCs, on page 130
- Configuring the vPC Peer Switch, on page 143
- Configuring Layer 3 over vPC, on page 146
- Isolating and Restoring a Switch from the vPC Complex, on page 148
- Verifying the vPC Configuration, on page 149
- vPC Default Settings, on page 155

Information About vPCs

vPC Overview

A virtual port channel (vPC) allows links that are physically connected to two Cisco Nexus devices or Cisco Nexus Fabric Extenders to appear as a single port channel by a third device (see the following figure). The third device can be a switch, server, or any other networking device. You can configure vPCs in topologies that include Cisco Nexus devices connected to Cisco Nexus Fabric Extenders. A vPC can provide multipathing, which allows you to create redundancy by enabling multiple parallel paths between nodes and load balancing traffic where alternative paths exist.

You configure the EtherChannels by using one of the following:

- No protocol
- Link Aggregation Control Protocol (LACP)

When you configure the EtherChannels in a vPC—including the vPC peer link channel—each switch can have up to 16 active links in a single EtherChannel.

Note You must enable the vPC feature before you can configure or run the vPC functionality.

To enable the vPC functionality, you must create a peer-keepalive link and a peer-link under the vPC domain for the two vPC peer switches to provide the vPC functionality.

To create a vPC peer link you configure an EtherChannel on one Cisco Nexus device by using two or more Ethernet ports. On the other switch, you configure another EtherChannel again using two or more Ethernet ports. Connecting these two EtherChannels together creates a vPC peer link.

Note We recommend that you configure the vPC peer-link EtherChannels as trunks.

The vPC domain includes both vPC peer devices, the vPC peer-keepalive link, the vPC peer link, and all of the EtherChannels in the vPC domain connected to the downstream device. You can have only one vPC domain ID on each vPC peer device.

Note Always attach all vPC devices using EtherChannels to both vPC peer devices.

A vPC provides the following benefits:

- Allows a single device to use an EtherChannel across two upstream devices
- Eliminates Spanning Tree Protocol (STP) blocked ports
- Provides a loop-free topology
- Uses all available uplink bandwidth
- Provides fast convergence if either the link or a switch fails
- Provides link-level resiliency
- Assures high availability

Terminology

vPC Terminology

The terminology used in vPCs is as follows:

- vPC—combined EtherChannel between the vPC peer devices and the downstream device.
- vPC peer device—One of a pair of devices that are connected with the special EtherChannel known as the vPC peer link.
- vPC peer link—link used to synchronize states between the vPC peer devices.
- vPC member port—Interfaces that belong to the vPCs.
- vPC domain—domain that 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. The vPC domain ID must be the same on both switches.

- vPC peer-keepalive link—The peer-keepalive link monitors the vitality of a vPC peer Cisco Nexus device. The peer-keepalive link sends configurable, periodic keepalive messages between vPC peer devices.

  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 vPCs.

Supported vPC Topologies

Cisco Nexus Device vPC Topology

vPC Domain

To create a vPC domain, you must first create a vPC domain ID on each vPC peer switch using a number from 1 to 1000. This ID must be the same on a set of vPC peer devices.

You can configure the EtherChannels and vPC peer links by using LACP or no protocol. When possible, we recommend that you use LACP on the peer-link, because LACP provides configuration checks against a configuration mismatch on the EtherChannel.

The vPC peer switches 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 switches 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 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.

The vPC peer switches use the vPC domain ID that you configure to automatically assign a unique vPC system MAC address. The switches use the vPC system MAC addresses only for link-scope operations, such as LACP or BPDUs. You can also configure a specific MAC address for the vPC domain.

We recommend that you configure the same VPC domain ID on both peers and, the domain ID should be unique in the network. For example, if there are two different VPCs (one in access and one in aggregation) then each vPC should have a unique domain ID.

After you create a vPC domain, the Cisco NX-OS software automatically creates a system priority for the vPC domain. You can also manually configure a specific system priority for the vPC domain.

Note If you manually configure the system priority, you must ensure that you assign the same priority value on both vPC peer switches. If the vPC peer switches have different system priority values, the vPC will not come up.

Peer-Keepalive Link and Messages

The Cisco NX-OS software uses a peer-keepalive link between the vPC peers to transmit periodic, configurable keepalive messages. You must have Layer 3 connectivity between the peer switches to transmit these messages; the system cannot bring up the vPC peer link unless a peer-keepalive link is already up and running.

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 We recommend that you configure the vPC peer-keepalive link on the Cisco Nexus device to run in the management VRF using the mgmt 0 interfaces. If you configure the default VRF, ensure that the vPC peer link is not used to carry the vPC peer-keepalive messages.

Compatibility Parameters for vPC Peer Links

Many configuration and operational parameters must be identical on all interfaces in the vPC. After you enable the vPC feature and configure the peer link on both vPC peer switches, Cisco Fabric Services (CFS) messages provide a copy of the configuration on the local vPC peer switch configuration to the remote vPC peer switch. The system then determines whether any of the crucial configuration parameters differ on the two switches.

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 EtherChannels.

Configuration Parameters That Must Be Identical

The configuration parameters in this section must be configured identically on both switches at either end of the vPC peer link.

Note You must ensure that all interfaces in the vPC have the identical operational and configuration parameters listed in this section.

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 switch automatically checks for compatibility 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
- 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 (MST)
- Enable or disable state per VLAN
- STP global settings:
    - Bridge Assurance setting
    - Port type setting—We recommend that you set all vPC interfaces as normal ports
    - Loop Guard settings
- STP interface settings:
    - Port type setting
    - Loop Guard
    - Root Guard

If any of these parameters are not enabled or defined on either switch, 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 switches, a misconfiguration might cause undesirable behavior in the traffic flow:

- MAC aging timers
- Static MAC entries
- VLAN interface—Each switch on the end of the vPC peer link must have a VLAN interface configured for the same VLAN on both ends and they must be in the same administrative and operational mode. Those VLANs configured on only one switch of the peer link do not pass traffic using the vPC or peer link. You must create all VLANs on both the primary and secondary vPC switches, or the VLAN will be suspended.
- Private VLAN configuration
- All ACL configurations and parameters
- Quality of service (QoS) configuration and parameters—Local parameters; global parameters must be identical
- STP interface settings:
    - BPDU Filter
    - BPDU Guard

- Cost
- Link type
- Priority
- VLANs (Rapid PVST+)

To ensure that all the configuration parameters are compatible, we recommend that you display the configurations for each vPC peer switch once you configure the vPC.

Graceful Type-1 Check

Per-VLAN Consistency Check

Type-1 consistency checks are performed on a per-VLAN basis when spanning tree is enabled or disabled on a VLAN. VLANs that do not pass the consistency check are brought down on both the primary and secondary switches while other VLANs are not affected.

vPC Auto-Recovery

When both vPC peer switches reload and only one switch reboots, auto-recovery allows that switch to assume the role of the primary switch and the vPC links will be allowed to come up after a predetermined period of time. The reload delay period in this scenario can range from 240 to 3600 seconds.

When vPCs are disabled on a secondary vPC switch due to a peer-link failure and then the primary vPC switch fails or is unable to forward traffic, the secondary switch reenables the vPCs. In this scenario, the vPC waits for three consecutive keepalive failures to recover the vPC links.

By default, auto-recovery is enabled on vPC. If you choose to disable auto-recovery and reload the switch, the disabled auto-recovery mode will be reset and auto-recovery will be enabled again after the switch reloads.

vPC Peer Links

A vPC peer link is the link that is used to synchronize the states between the vPC peer devices.

Note You must configure the peer-keepalive link before you configure the vPC peer link or the peer link will not come up.

vPC Peer Link Overview

You can have only two switches as vPC peers; each switch can serve as a vPC peer to only one other vPC peer. The vPC peer switches can also have non-vPC links to other switches.

To make a valid configuration, you configure an EtherChannel on each switch and then configure the vPC domain. You assign the EtherChannel on each switch as a peer link. For redundancy, we recommend that you should configure at least two dedicated ports into the EtherChannel; if one of the interfaces in the vPC peer link fails, the switch automatically falls back to use another interface in the peer link.

Note We recommend that you configure the EtherChannels in trunk mode.

Many operational parameters and configuration parameters must be the same in each switch connected by a vPC peer link. Because each switch is completely independent on the management plane, you must ensure that the switches are compatible on the critical parameters. vPC peer switches have separate control planes. After configuring the vPC peer link, you should display the configuration on each vPC peer switch to ensure that the configurations are compatible.

Note You must ensure that the two switches connected by the vPC peer link have certain identical operational and configuration parameters.

When you configure the vPC peer link, the vPC peer switches negotiate that one of the connected switches is the primary switch and the other connected switch is the secondary switch. By default, the Cisco NX-OS software uses the lowest MAC address to elect the primary switch. The software takes different actions on each switch—that is, the primary and secondary—only in certain failover conditions. If the primary switch fails, the secondary switch becomes the operational primary switch when the system recovers, and the previously primary switch is now the secondary switch.

You can also configure which of the vPC switches is the primary switch. If you want to configure the role priority again to make one vPC switch the primary switch, configure the role priority on both the primary and secondary vPC switches with the appropriate values, shut down the EtherChannel that is the vPC peer link on both switches by entering the **shutdown** command, and reenable the EtherChannel on both switches by entering the **no shutdown** command.

MAC addresses that are learned over vPC links are also synchronized between the peers.

Configuration information flows across the vPC peer links using the Cisco Fabric Services over Ethernet (CFSoE) protocol. All MAC addresses for those VLANs configured on both switches are synchronized between vPC peer switches. The software uses CFSoE for this synchronization.

If the vPC peer link fails, the software checks the status of the remote vPC peer switch using the peer-keepalive link, which is a link between vPC peer switches, to ensure that both switches are up. If the vPC peer switch is up, the secondary vPC switch disables all vPC ports on its switch. The data then forwards down the remaining active links of the EtherChannel.

The software learns of a vPC peer switch 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 switches. 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 switch. The keepalive messages are used only when all the links in the peer link fail.

vPC Number

Once you have created the vPC domain ID and the vPC peer link, you can create EtherChannels to attach the downstream switch to each vPC peer switch. That is, you create one single EtherChannel on the downstream switch with half of the ports to the primary vPC peer switch and the other half of the ports to the secondary peer switch.

On each vPC peer switch, you assign the same vPC number to the EtherChannel that connects to the downstream switch. You will experience minimal traffic disruption when you are creating vPCs. To simplify the configuration, you can assign the vPC ID number for each EtherChannel to be the same as the EtherChannel itself (that is, vPC ID 10 for EtherChannel 10).

Note The vPC number that you assign to the EtherChannel that connects to the downstream switch from the vPC peer switch must be identical on both vPC peer switches.

Layer 3 over vPC

From Cisco NX-OS Release 7.3(0)N1(1), a Layer 3 (L3) device can form peering adjacency with both the vPC peers in a vPC domain. Traffic sent over a peer link will not have Time To Live (TTL) decremented.

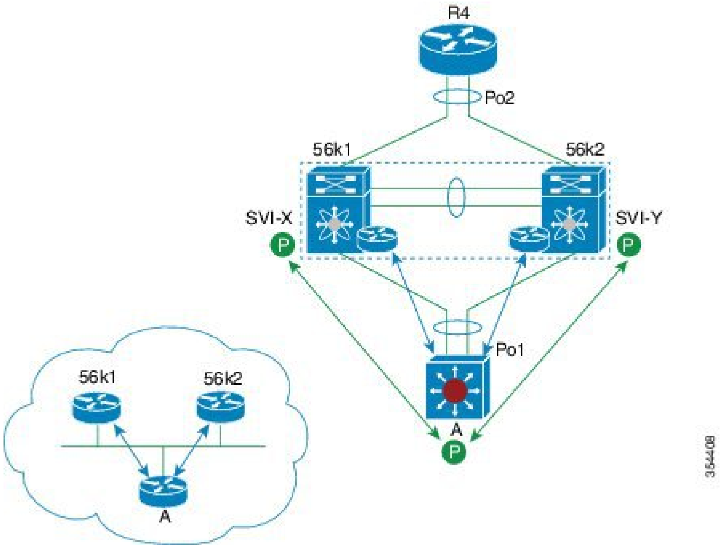
The peer-gateway feature should be enabled before configuring the Layer 3 over vPC feature.

The following are the benefits of configuring this feature:

- You can set up peer adjacency between Layer 3 device and a vPC peer without separate Layer 3 links. Both bridged and routed traffic can flow over the same link.
- Peer adjacency can be formed on vPC VLANs .
- Peer adjacency helps in faster convergence if link or device failure occurs in traffic.

The following illustration shows that peer-gateway feature allows the vPC peer (SVI X) to forward packets on behalf of the other peer (SVI Y). This saves bandwidth by avoiding traffic over the peer link.

Figure 10: Layer 3 over vPC Solution



Layer 3 over vPC: Supported Designs

The following figures illustrate the designs that are supported for configuring Layer 3 over vPC.

Figure 11: Router Peers with Both vPC Peers

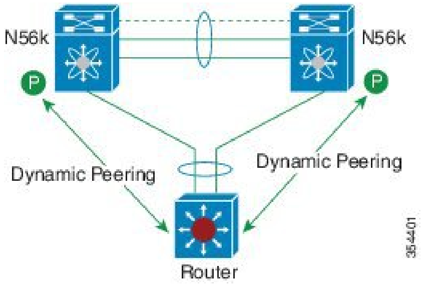
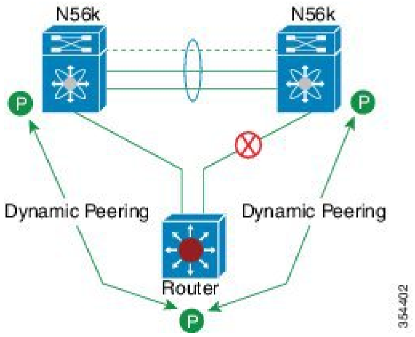


Figure 12: Peering Established Over an STP Interconnection Using a vPC VLAN.



Router peers with both vPC peers.

Figure 13: Route Peering of Orphan Device with both the vPC peers

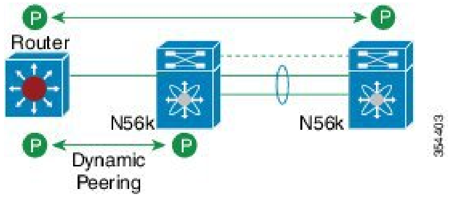
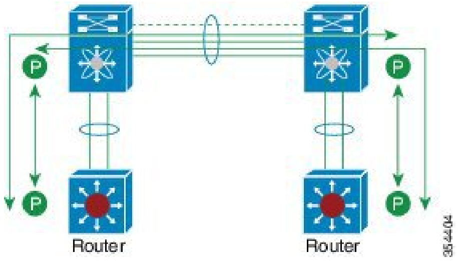
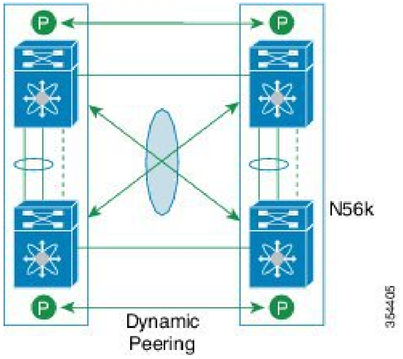


Figure 14: Peering over PC Interconnection and Over vPC peer Link Using vPC VLAN.



Both the Routers peer with both vPC peers.

Figure 15: Peering Over a vPC Interconnection (DCI Case)

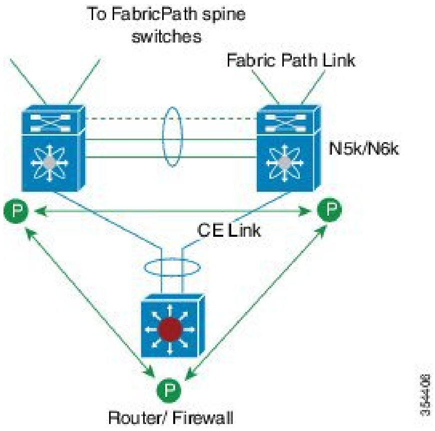


Each Nexus device peers with two vPC peers.

Peering with vPC+ (supported since Release 6.0(2)N2(1)):

- The peer link ports are configured as FabricPath core ports.
- North facing ports function as FabricPath spine port.
- North facing ports can also be Layer 3 ports in non-FabricPath topology.

Figure 16: Peering with vPC+



Layer 3 over vPC: Unsupported Designs

The following figures illustrate the designs that are not supported for configuring the Layer 3 over vPC feature.

The following design is not supported because the TTL distance between the two orphan devices is 2. This leads to the loss of control packets at the vPC while forming routing adjacencies.

Figure 17: Orphan Device Peering over vPC Interconnection (DCI Case)

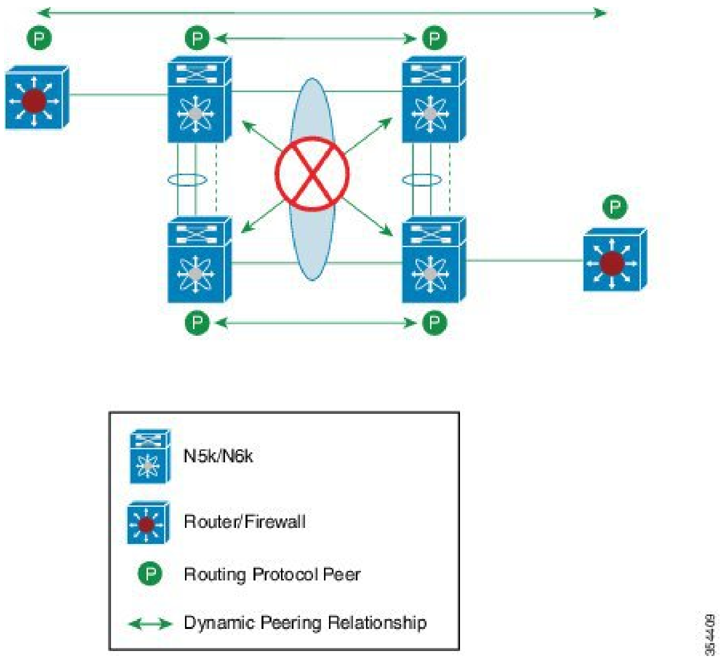


Figure 18: Peering with vPC Peers Over FEX vPC Host Interfaces

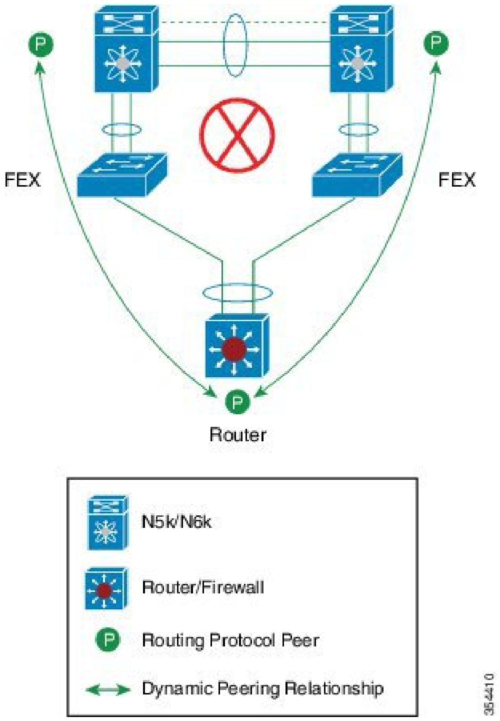


Figure 19: Peering Across vPC Interfaces with Unequal L3 metrics

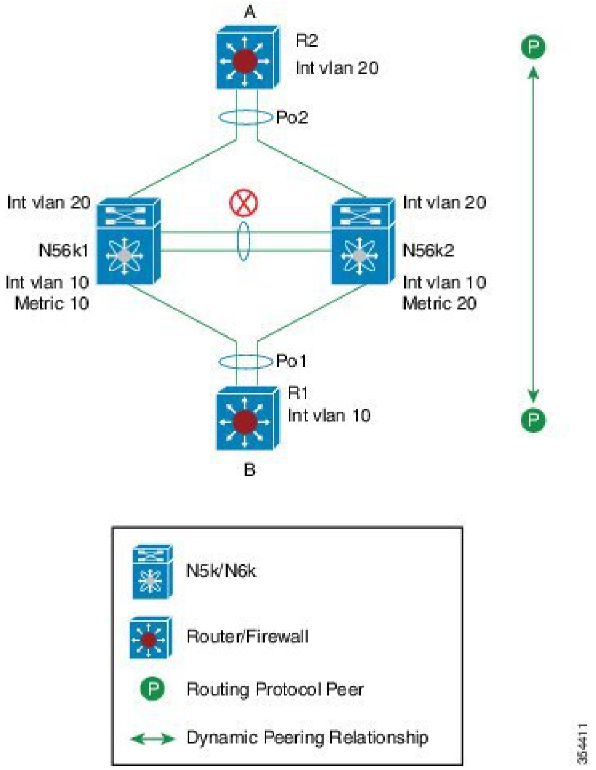


Figure 20: Peering with vPC+ Peers and STP Interconnection Using a vPC+ VLAN

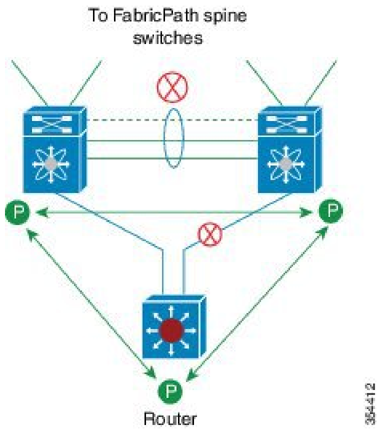


Figure 21: Route Peering with Orphan Device with Both the vPC+ peers

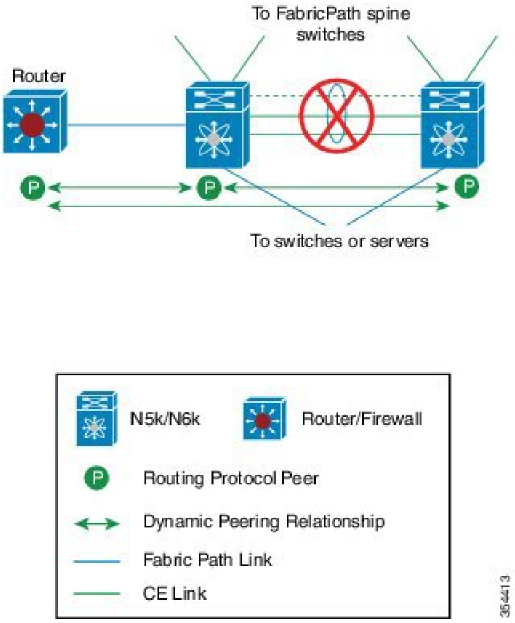
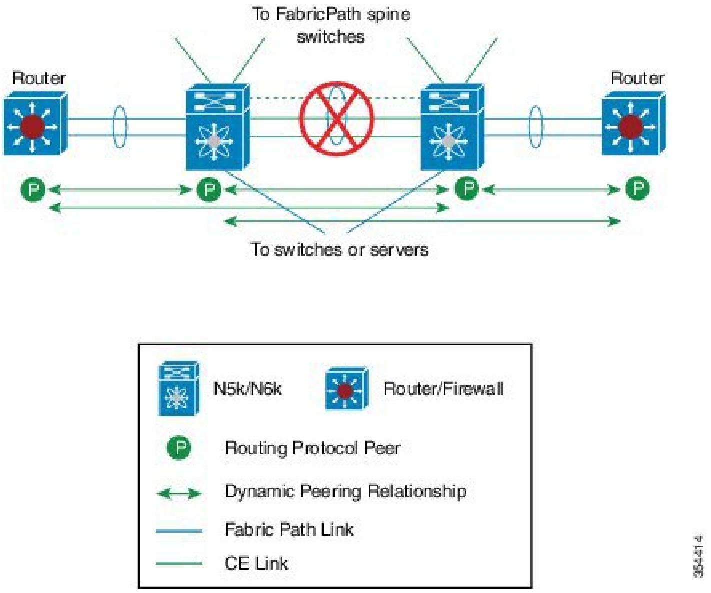


Figure 22: Peering over PC Interconnection and Over vPC+ Peer Link Using vPC VLAN



vPC Interactions with Other Features

Configuring vPC Peer Links and Links to the Core

Configure the command line interface by using 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.

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.

Before you begin

To configure a track list to switch over vPC to the remote peer when all related interfaces fail:

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode.
Step 3	switch(config-if)# **track** *track-id* **interface** *type slot*/*port* **line-protocol**	Configures the track objects on an interface (Layer 3 to core).
Step 4	switch(config-track)# **track** *track-id* **interface** *type slot*/*port* **line-protocol**	Tracks the objects on an interface (Layer 3 to core).
Step 5	switch(config)# **track** *track-id* **interface port-channel** *port* **line-protocol**	Configures the track objects on a port channel (vPC peer link).
Step 6	switch(config)# **track** *track-id* **list boolean** [**OR** | **AND**]	Creates a track list that contains all the interfaces in the track list using the Boolean OR to trigger when all the objects fail. or trigger a switchover when any core interface or peer-link goes down using Boolean AND.
Step 7	switch(config-track)# **object** *number*	Specifiecs the object number.
Step 8	switch(config-track)# **end**	Exits track configuration mode.
Step 9	switch(config)# **vpc domain** *domain-id*	Enters vPC domain configuration.
Step 10	switch(config-vpc-domain)# **track** *number*	Adds the track object to the vPC domain.
Step 11	(Optional) switch(config)# **show vpc brief**	Displays the track object.

	Command or Action	Purpose
Step 12	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a track list to trigger when all the objects fail using Boolean OR:

```
switch# configure terminal
switch(config)# interface ethernet 8/35
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
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
switch(config)# vpc domain 1
switch(config-vpc-domain)# track 44
switch(config)# copy running-config startup-config
```

vPC and LACP

The Link Aggregation Control Protocol (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 EtherChannels, including those channels from the downstream switch. We recommend that you configure LACP with active mode on the interfaces on each EtherChannel on the vPC peer switches. This configuration allows you to more easily detect compatibility between switches, unidirectional links, and multihop connections, and provides dynamic reaction to run-time changes and link failures.

The vPC peer link supports 16 EtherChannel interfaces.

Note When you manually configure the system priority, you must ensure that you assign the same priority value on both vPC peer switches. If the vPC peer switches have different system priority values, vPC does not come up.

vPC Peer Links and STP

When you first bring up the vPC functionality, STP reconverges. STP treats the vPC peer link as a special link and always includes the vPC peer link in the STP active topology.

We recommend that you set all the vPC peer link interfaces to the STP network port type so that Bridge Assurance is automatically enabled on all vPC peer links. We also recommend that you do not enable any of the STP enhancement features on VPC peer links.

You must configure a list of parameters to be identical on the vPC peer switches on both sides of the vPC peer link.

STP is distributed; that is, the protocol continues running on both vPC peer switches. However, the configuration on the vPC peer switch elected as the primary switch controls the STP process for the vPC interfaces on the secondary vPC peer switch.

The primary vPC switch synchronizes the STP state on the vPC secondary peer switch using Cisco Fabric Services over Ethernet (CFSoE).

The vPC manager performs a proposal/handshake agreement between the vPC peer switches that sets the primary and secondary switches and coordinates the two switches for STP. The primary vPC peer switch then controls the STP protocol for vPC interfaces on both the primary and secondary switches.

The Bridge Protocol Data Units (BPDUs) use the MAC address set for the vPC for the STP bridge ID in the designated bridge ID field. The vPC primary switch sends these BPDUs on the vPC interfaces.

Note Display the configuration on both sides of the vPC peer link to ensure that the settings are identical. Use the **show spanning-tree** command to display information about the vPC.

vPC and ARP

Table synchronization across vPC peers is managed in Cisco NX-OS using the reliable transport mechanism of the Cisco Fabric Services over Ethernet (CFSoE) protocol. To support faster convergence of address tables between the vPC peers, the **ip arp synchronize** command must be enabled. This convergence is designed to overcome the delay involved in ARP table restoration when the peer-link port channel flaps or when a vPC peer comes back online.

To improve performance, we recommend that you turn on the ARP sync feature. By default, it is not enabled.

To check whether or not ARP sync is enabled, enter the following command:

```
switch# show running
```

To enable ARP sync, enter the following command:

```
switch(config-vpc-domain) # ip arp synchronize
```

CFSoE

The Cisco Fabric Services over Ethernet (CFSoE) is a reliable state transport mechanism that you can use to synchronize the actions of the vPC peer devices. CFSoE carries messages and packets for many features linked with vPC, such as STP and IGMP. Information is carried in CFS/CFSoE protocol data units (PDUs).

When you enable the vPC feature, the device automatically enables CFSoE, and you do not have to configure anything. CFSoE distributions for vPCs do not need the capabilities to distribute over IP or the CFS regions. You do not need to configure anything for the CFSoE feature to work correctly on vPCs.

You can use the **show mac address-table** command to display the MAC addresses that CFSoE synchronizes for the vPC peer link.

Note Do not enter the **no cfs eth distribute** or the **no cfs distribute** command. CFSoE must be enabled for vPC functionality. If you do enter either of these commands when vPC is enabled, the system displays an error message.

When you enter the **show cfs application** command, the output displays "Physical-eth," which shows the applications that are using CFSoE.

vPC Peer Switch

The vPC peer switch feature addresses performance concerns around STP convergence. This feature allows a pair of Cisco Nexus devices to appear as a single STP root in the Layer 2 topology. This feature eliminates the need to pin the STP root to the vPC primary switch and improves vPC convergence if the vPC primary switch fails.

To avoid loops, the vPC peer link is excluded from the STP computation. In vPC peer switch mode, STP BPDUs are sent from both vPC peer devices to avoid issues related to STP BPDU timeout on the downstream switches, which can cause traffic disruption.

This feature can be used with the pure peer switch topology in which the devices all belong to the vPC.

Note Peer-switch feature is supported on networks that use vPC and STP-based redundancy is not supported. If the vPC peer-link fail in a hybrid peer-switch configuration, you can lose traffic. In this scenario, the vPC peers use the same STP root ID as well same bridge ID. The access switch traffic is split in two with half going to the first vPC peer and the other half to the second vPC peer. With the peer link failed, there is no impact on north/south traffic but east-west traffic will be lost (black-holed).

For information on STP enhancement features and Rapid PVST+, see the *Layer 2 Switching Configuration Guide* for your device.

Guidelines and Limitations for vPCs

vPC has the following configuration guidelines and limitations:

- You must enable the vPC feature before you can configure vPC peer-link and vPC interfaces.
- You must configure the peer-keepalive link before the system can form the vPC peer link.
- The vPC peer-link needs to be formed using a minimum of two 10-Gigabit Ethernet interfaces.
- You can connect a pair of Cisco Nexus 5600 series switches in a vPC directly to another switch or to a server. vPC peer switches must be of the same type ((including same switch type, and LEM type). For example, you can connect a pair of Cisco Nexus 5600 series switches, but you cannot connect a Cisco Nexus 5600 series switch to a Cisco Nexus 6000 series switch, or a Cisco Nexus 5500 series switch in a vPC topology. For a Cisco Nexus 5696 switch, you can use Cisco Nexus 5600 LEMs as well as Cisco Nexus 6000 LEMs, but you cannot use different LEM versions in a vPC domain.
- Only port channels can be in vPCs. A vPC can be configured on a normal port channel (switch-to-switch vPC topology), on a port channel fabric interface (fabric extender vPC topology), and on a port channel host interface (host interface vPC topology).
- A Fabric Extender can be a member of a Host Interface vPC topology or a Fabric Extender vPC topology but not both simultaneously.
- You must configure both vPC peer switches; the configuration is not automatically synchronized between the vPC peer devices.

- Check that the necessary configuration parameters are compatible on both sides of the vPC peer link.
- You may experience minimal traffic disruption while configuring vPCs.
- You should configure all the port channels in the vPC using LACP with the interfaces in active mode.
- When the **peer-switch** command is configured and vPC keepalive messages exchanged through an SVI instead of a management interface, additional Spanning Tree Protocol (STP) configuration is required. STP needs to be disabled on the dedicated link that carries the keepalive traffic between the vPC peers. You can disable STP on the dedicated link by configuring STP BPDUfilter on the both ends of the dedicated link. We recommend that the VLAN of the vPC keepalive SVI be allowed on only the interconnecting dedicated link and disallowed on all other links, including the peer link.
- You should configure both the SVIs as active/active, otherwise this can lead to traffic blackhole.
- A Cisco Nexus 6000 Series Switch that is connected to a router and a vPC peer creates an OSPF association with the attached router but not with the vPC peer. This situation happens if a non-vpc VLAN is on a separate trunk between the VPC peers. If the non-vpc VLAN is on the vpc-peer link, then OSPF works for both vPC peers. This situation only happens when peer-gateway is enabled.
- In some vPC failure scenarios, vPC secondary switch suspends its vPC port-channels after the vPC primary switch failure. To avoid vPC secondary switch suspensions, disable vPC peer-keepalive before bringing down the vPC primary switch (in case of scheduled power down).
- When you enable the vPC Peer Gateway feature, Cisco NX-OS automatically disables IP redirects on all the interface VLANs that are mapped over a vPC VLAN, to avoid generation of IP redirect messages for packets switched through the peer gateway router.

Configuring vPCs

Enabling vPCs

You must enable the vPC feature before you can configure and use vPCs.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature vpc**	Enables vPCs on the switch.
Step 3	(Optional) switch# **show feature**	Displays which features are enabled on the switch.
Step 4	(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
```

Disabling vPCs

You can disable the vPC feature.

Note When you disable the vPC feature, the Cisco Nexus device clears all the vPC configurations.

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 switch.
Step 3	(Optional) switch# **show feature**	Displays which features are enabled on the switch.
Step 4	(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
```

Creating a vPC Domain

You must create identical vPC domain IDs on both the vPC peer devices. This domain ID is used to automatically form the vPC system MAC address.

Before you begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link.

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 switch, and enters the vpc-domain configuration mode. There is

	Command or Action	Purpose
		no default *domain-id* ; the range is from 1 to 1000. **Note** You can also use the **vpc domain** command to enter the vpc-domain configuration mode for an existing vPC domain.
Step 3	(Optional) switch# **show vpc brief**	Displays brief information about each vPC domain.
Step 4	(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
```

Configuring a vPC Keepalive Link and Messages

You can configure the destination IP for the peer-keepalive link that carries the keepalive messages. Optionally, you can configure other parameters for the keepalive messages.

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.

Ensure that both the source and destination IP addresses used for the peer-keepalive message are unique in your network and these IP addresses are reachable from the Virtual Routing and Forwarding (VRF) instance associated with the vPC peer-keepalive link.

Note We recommend that you configure a separate VRF instance and put a Layer 3 port from each vPC peer switch into that VRF instance for the vPC peer-keepalive link. Do not use the peer link itself to send vPC peer-keepalive messages.

Before you begin

Ensure that you have enabled the vPC feature.

You must configure the vPC peer-keepalive link before the system can form the vPC peer link.

You must configure both switches on either side of the vPC peer link.

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 switch if it does not already exist, and enters the vpc-domain configuration mode.
Step 3	switch(config-vpc-domain)# **peer-keepalive destination** *ipaddress* [**hold-timeout** *secs* | **interval** *msecs* {**timeout** *secs*} | **precedence** {*prec-value* | **network** | **internet** | **critical** | **flash-override** | **flash** | **immediate priority** | **routine**} | **tos** {*tos-value* | **max-reliability** | **max-throughput** | **min-delay** | **min-monetary-cost** | **normal**} | **tos-byte** *tos-byte-value*} | **source** *ipaddress* | **vrf** {*name* | **management vpc-keepalive**}]	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. The management ports and VRF are the defaults.
Step 4	(Optional) switch(config-vpc-domain)# **vpc peer-keepalive destination** *ipaddress* **source** *ipaddress*	Configures a separate VRF instance and puts a Layer 3 port from each vPC peer device into that VRF for the vPC peer-keepalive link.
Step 5	(Optional) switch# **show vpc peer-keepalive**	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

This example shows how to configure the destination IP address for the vPC-peer-keepalive link:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# peer-keepalive destination 10.10.10.42
```

This example shows how to set up the peer keepalive link connection between the primary and secondary vPC device:

```
switch(config)# vpc domain 100
switch(config-vpc-domain)# peer-keepalive destination 192.168.2.2 source 192.168.2.1
Note:--------:: Management VRF will be used as the default VRF ::--------
switch(config-vpc-domain)#
```

This example shows how to create a separate VRF named vpc_keepalive for the vPC keepalive link and how to verify the new VRF:

```
vrf context vpc_keepalive
interface Ethernet1/31
  switchport access vlan 123
interface Vlan123
```

```
  vrf member vpc_keepalive
  ip address 123.1.1.2/30
  no shutdown
vpc domain 1
  peer-keepalive destination 123.1.1.1 source 123.1.1.2 vrf
vpc_keepalive

L3-NEXUS-2# show vpc peer-keepalive

vPC keep-alive status            : peer is alive
--Peer is alive for              : (154477) seconds, (908) msec
--Send status                    : Success
--Last send at                   : 2011.01.14 19:02:50 100 ms
--Sent on interface              : Vlan123
--Receive status                 : Success
--Last receive at                : 2011.01.14 19:02:50 103 ms
--Received on interface          : Vlan123
--Last update from peer          : (0) seconds, (524) msec

vPC Keep-alive parameters
--Destination                    : 123.1.1.1
--Keepalive interval             : 1000 msec
--Keepalive timeout              : 5 seconds
--Keepalive hold timeout         : 3 seconds
--Keepalive vrf                  : vpc_keepalive
--Keepalive udp port             : 3200
--Keepalive tos                  : 192

The services provided by the switch , such as ping, ssh, telnet,
radius, are VRF aware. The VRF name need to be configured or
specified in order for the correct routing table to be used.
L3-NEXUS-2# ping 123.1.1.1 vrf vpc_keepalive
PING 123.1.1.1 (123.1.1.1): 56 data bytes
64 bytes from 123.1.1.1: icmp_seq=0 ttl=254 time=3.234 ms
64 bytes from 123.1.1.1: icmp_seq=1 ttl=254 time=4.931 ms
64 bytes from 123.1.1.1: icmp_seq=2 ttl=254 time=4.965 ms
64 bytes from 123.1.1.1: icmp_seq=3 ttl=254 time=4.971 ms
64 bytes from 123.1.1.1: icmp_seq=4 ttl=254 time=4.915 ms

--- 123.1.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 3.234/4.603/4.971 ms
```

Creating a vPC Peer Link

You can create a vPC peer link by designating the EtherChannel that you want on each switch as the peer link for the specified vPC domain. We recommend that you configure the EtherChannels 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 switch for redundancy.

Before you begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link

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 EtherChannel that you want to use as the vPC peer link for this switch, and enters the interface configuration mode.
Step 3	switch(config-if)# **vpc peer-link**	Configures the selected EtherChannel as the vPC peer link, and enters the vpc-domain configuration mode.
Step 4	(Optional) switch# **show vpc brief**	Displays information about each vPC, including information about the vPC peer link.
Step 5	(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)# vpc peer-link
```

Checking the Configuration Compatibility

After you have configured the vPC peer link on both vPC peer switches, check that the configurations are consistent on all vPC interfaces.

Note The following QoS parameters support Type 2 consistency checks:

- Network QoS—MTU and Pause
- Input Queuing —Bandwidth and Absolute Priority
- Output Queuing—Bandwidth and Absolute Priority

In the case of a Type 2 mismatch, the vPC is not suspended. Type 1 mismatches suspend the vPC.

Command or Action	Purpose
switch# **show vpc consistency-parameters** {**global** | **interface port-channel** *channel-number*}	Displays the status of those parameters that must be consistent across all vPC interfaces.

This example shows how to check that the required configurations are compatible across all the vPC interfaces:

```
switch# show vpc consistency-parameters global
     Legend:
         Type 1 : vPC will be suspended in case of mismatch
```

```
Name                        Type  Local Value            Peer Value

------------- ----  ---------------------- -----------------------
QoS                         2     ([], [], [], [], [],   ([], [], [], [], [],
                                   [])                    [])

Network QoS (MTU)           2     (1538, 0, 0, 0, 0, 0)  (1538, 0, 0, 0, 0, 0)
Network Qos (Pause)         2     (F, F, F, F, F, F)     (1538, 0, 0, 0, 0, 0)
Input Queuing (Bandwidth)   2     (100, 0, 0, 0, 0, 0)   (100, 0, 0, 0, 0, 0)
Input Queuing (Absolute     2     (F, F, F, F, F, F)     (100, 0, 0, 0, 0, 0)
Priority)
Output Queuing (Bandwidth)  2     (100, 0, 0, 0, 0, 0)   (100, 0, 0, 0, 0, 0)
Output Queuing (Absolute    2     (F, F, F, F, F, F)     (100, 0, 0, 0, 0, 0)
Priority)
STP Mode                    1     Rapid-PVST             Rapid-PVST
STP Disabled                1     None                   None
STP MST Region Name         1     ""                     ""
STP MST Region Revision     1     0                      0
STP MST Region Instance to  1
  VLAN Mapping

STP Loopguard               1     Disabled               Disabled
STP Bridge Assurance        1     Enabled                Enabled
STP Port Type, Edge         1     Normal, Disabled,      Normal, Disabled,
BPDUFilter, Edge BPDUGuard        Disabled               Disabled
STP MST Simulate PVST       1     Enabled                Enabled
Allowed VLANs               -     1,624                  1
Local suspended VLANs       -     624                    -
switch#
```

Enabling vPC Auto-Recovery

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Enters vpc-domain configuration mode for an existing vPC domain.
Step 3	switch(config-vpc-domain)# **auto-recovery reload-delay** *delay*	Enables the auto-recovery feature and sets the reload delay period. The default is disabled.

Example

This example shows how to enable the auto-recovery feature in vPC domain 10 and set the delay period for 240 seconds:

```
switch(config)# vpc domain 10
switch(config-vpc-domain)# auto-recovery reload-delay 240
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
```

This example shows how to view the status of the auto-recovery feature in vPC domain 10:

```
switch(config-vpc-domain)# show running-config vpc
!Command: show running-config vpc
!Time: Tue Dec  7 02:38:44 2010


feature vpc
vpc domain 10
  peer-keepalive destination 10.193.51.170
  auto-recovery
```

Configuring the Restore Time Delay

You can configure a restore timer that delays the vPC from coming back up until after the peer adjacency forms and the VLAN interfaces are back up. This feature avoids packet drops if the routing tables fail to converge before the vPC is once again passing traffic.

Before you begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link with the following procedures.

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 switch if it does not already exist, and enters vpc-domain configuration mode.
Step 3	switch(config-vpc-domain)# **delay restore** *time*	Configures the time delay before the vPC is restored. The restore time is the number of seconds to delay bringing up the restored vPC peer device. The range is from 1 to 3600. The default is 30 seconds.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the delay reload time for a vPC link:

```
switch(config)# vpc domain 1
switch(config-vpc-domain)# delay restore 10
switch(config-vpc-domain)#
```

Excluding VLAN Interfaces from Shutting Down a vPC Peer Link Fails

When a vPC peer-link is lost, the vPC secondary switch suspends its vPC member ports and its switch virtual interface (SVI) interfaces. All Layer 3 forwarding is disabled for all VLANs on the vPC secondary switch. You can exclude specific SVI interfaces so that they are not suspended.

Before you begin

Ensure that the VLAN interfaces have been configured.

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 switch if it does not already exist, and enters vpc-domain configuration mode.
Step 3	switch(config-vpc-domain))# **dual-active exclude interface-vlan** *range*	Specifies the VLAN interfaces that should remain up when a vPC peer-link is lost. range—Range of VLAN interfaces that you want to exclude from shutting down. The range is from 1 to 4094.

Example

This example shows how to keep the interfaces on VLAN 10 up on the vPC peer switch if a peer link fails:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# dual-active exclude interface-vlan 10
switch(config-vpc-domain)#
```

Configuring the VRF Name

The switch services, such as ping, ssh, telnet, radius, are VRF aware. You must configure the VRF name in order for the correct routing table to be used.

You can specify the VRF name.

Procedure

	Command or Action	Purpose
Step 1	switch# **ping** *ipaddress* **vrf** *vrf-name*	Specifies the virtual routing and forwarding (VRF) name to use. The VRF name is case sensitive and can be a maximum of 32 characters..

Example

This example shows how to specifiy the VRF named vpc_keepalive:

```
switch# ping 123.1.1.1 vrf vpc_keepalive
PING 123.1.1.1 (123.1.1.1): 56 data bytes
64 bytes from 123.1.1.1: icmp_seq=0 ttl=254 time=3.234 ms
64 bytes from 123.1.1.1: icmp_seq=1 ttl=254 time=4.931 ms
64 bytes from 123.1.1.1: icmp_seq=2 ttl=254 time=4.965 ms
64 bytes from 123.1.1.1: icmp_seq=3 ttl=254 time=4.971 ms
64 bytes from 123.1.1.1: icmp_seq=4 ttl=254 time=4.915 ms

--- 123.1.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 3.234/4.603/4.971 ms
```

Binding a VRF Instance to a vPC

You can bind a VRF instance to a vPC. One reserved VLAN is required for each VRF. Without this command, the receivers in a non-vPC VLAN and the receivers connected to a Layer 3 interface might not receive multicast traffic. The non-vPC VLANs are the VLANs that are not trunked over a peerlink.

Note If you configure the **vpc bind-vrf** command to forward multicast traffic over the vPC peer link, you need to reload the switches to avoid any traffic loss.

Before you begin

Use the **show interfaces brief** command to view the interfaces that are in use on a switch. To bind the VRF instance to the vPC, you must use a VLAN that is not already in use.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc bind-vrf** *vrf-name* **vlan** *vlan-id*	Binds a VRF instance to a vPC and specifies the VLAN to bind to the vPC. The VLAN ID range is from 1 to 3967 and from 4049 to 4093.

Example

This example shows how to bind a vPC to the default VRF instance using VLAN 2:

```
switch(config)# vpc bind-vrf default vlan vlan2
```

Moving Other Port Channels into a vPC

Before you begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link with the following procedure.

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 put into the vPC to connect to the downstream switch, and enters interface configuration mode. **Note** A vPC can be configured on a normal port channel (physical vPC topology) and on a port channel host interface (host interface vPC topology)
Step 3	switch(config-if)# **vpc** *number*	Configures the selected port channel into the vPC to connect to the downstream switch. The range is from 1 to 4096. The vPC *number* that you assign to the port channel that connects to the downstream switch from the vPC peer switch must be identical on both vPC peer switches.
Step 4	(Optional) switch# **show vpc brief**	Displays information about each vPC.
Step 5	(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 that will connect to the downstream device:

```
switch# configure terminal

switch(config)# interface port-channel 20

switch(config-if)# vpc 5
```

Manually Configuring a vPC Domain MAC Address

Note Configuring the system address is an optional configuration step.

Before you begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default *domain-id*; 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	(Optional) switch# **show vpc role**	Displays the vPC system MAC address.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a vPC domain MAC address:

```
switch# configure terminal

switch(config)# vpc domain 5

switch(config-if)# system-mac 23fb.4ab5.4c4e
```

Manually Configuring the System Priority

When you create a vPC domain, the system automatically creates a vPC system priority. However, you can also manually configure a system priority for the vPC domain.

Before you begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default *domain-id*; 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	(Optional) switch# **show vpc brief**	Displays information about each vPC, including information about the vPC peer link.
Step 5	(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)# vpc domain 5
switch(config-if)# system-priority 4000
```

Manually Configuring a vPC Peer Switch Role

By default, the Cisco NX-OS software elects a primary and secondary vPC peer switch after you configure the vPC domain and both sides of the vPC peer link. However, you may want to elect a specific vPC peer switch as the primary switch for the vPC. Then, you would manually configure the role value for the vPC peer switch that you want as the primary switch to be lower than the other vPC peer switch.

vPC does not support role preemption. If the primary vPC peer switch fails, the secondary vPC peer switch takes over to become operationally the vPC primary switch. However, the original operational roles are not restored when the formerly primary vPC comes up again.

Before you begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **vpc domain** *domain-id*	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default *domain-id*; 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 65535. The default value is 32667.
Step 4	(Optional) switch# **show vpc brief**	Displays information about each vPC, including information about the vPC peer link.
Step 5	(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)# vpc domain 5

switch(config-if)# role priority 4000
```

Configuring the vPC Peer Switch

Configuring a Pure vPC Peer Switch Topology

You can configure a pure vPC peer switch topology using the **peer-switch** command and then you set the best possible (lowest) spanning tree bridge priority value.

Note The values you apply for the spanning tree priority must be identical on both vPC peers.

Before you begin

Ensure that you have enabled the vPC feature.

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **vpc domain** *domain-id*	Enters the vPC domain number that you want to configure. The system enters the vpc-domain configuration mode.
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.
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. **Note** This value must be identical on both vPC peers.
Step 5	switch(config-vpn-domain)# **exit**	Exits the vpc-domain configuration mode.
Step 6	(Optional) switch(config)# **show spanning-tree summary**	Displays a summary of the spanning tree port states including the vPC peer switch. Look for the following line in the command output: `vPC peer-switch is enabled (operational)`
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a 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)# exit
switch(config)# spanning-tree vlan 1 priority 8192
switch(config)# show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0001-VLAN0050, VLAN0100-VLAN0149, VLAN0200-VLAN0249
  VLAN0300-VLAN0349, VLAN0400-VLAN0599, VLAN0900-VLAN0999
Port Type Default                        is disable
Edge Port [PortFast] BPDU Guard Default  is disabled
Edge Port [PortFast] BPDU Filter Default is disabled
Bridge Assurance                         is enabled
Loopguard Default                        is disabled
Pathcost method used                     is short
```

```
vPC peer-switch                          is enabled (operational)
Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
VLAN0001                      0         0        0         16         16
VLAN0002                      0         0        0         16         16
switch(config)# copy running-config startup-config
switch(config)#
```

Configuring a Hybrid vPC Peer Switch Topology

You can configure a hybrid vPC and non-vPC peer switch topology by using the spanning-tree pseudo-information command to change the designated bridge ID so that it meets the STP VLAN-based load-balancing criteria and then change the root bridge ID priority to a value that is better than the best bridge priority. You then enable the peer switch. For more information, see the command reference for your device.

Note If you previously configured global spanning tree parameters and you subsequently configure spanning tree pseudo information parameters, be aware that the pseudo information parameters take precedence over the global parameters.

Before you begin

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. **Note** This configuration takes precedence over any global spanning tree configurations.
Step 3	switch(config-pseudo)# **vlan** *vlan-id* **designated priority** *priority*	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-id* **root priority** *priority*	Configures the root bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440. **Note** This value must be identical on both vPC peers to have an operational peer switch.
Step 5	switch(config-pseudo)# **exit**	Exists spanning tree pseudo information configuration mode.

	Command or Action	Purpose
Step 6	switch(config)# **vpc domain** *domain-id*	Enters the vPC domain number that you want to configure. The system enters the vpc-domain configuration mode.
Step 7	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 8	switch(config-vpc-domain)# **exit**	Exits the vpc-domain configuration mode.
Step 9	(Optional) switch(config)# **show spanning-tree summary**	Displays a summary of the spanning tree port states including the vPC peer switch. Look for the following line in the command output: `vPC peer-switch is enabled (operational)`
Step 10	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a 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)# exit
switch(config)# vpc domain 5
switch(config-vpc-domain)# peer-switch
switch(config-vpc-domain)# exit
switch(config)# copy running-config startup-config
```

Configuring Layer 3 over vPC

Before you begin

- Ensure that the peer-gateway feature is enabled and configured on both the vPC peers.
- Ensure that both the peers are running an image that supports the Layer 3 over vPC feature.
- Ensure that the **mac move notify enabled** flag is set to 1 on both the vPC peers . To view the flag status, use the **show platform fwm info global | grep "mac move"** command in Privilege Exec mode.

On Cisco Nexus devices with Cisco NX-OS Release 7.3.(0)N1(1) and 7.3.(1)N1(1), if the **mac move notify enabled** flag is set to 0, then use the **no debug platform fwm mac_move_notify_disable** command to set the flag to 1.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Enter the vpc-domain configuration mode for an existing vpc domain:

switch(config)# **vpc domain** *domain_id*

Step 3 Enable peer gateway on both the vPC peers:

switch(config-vpc-domain)# **peer-gateway**

Step 4 Enable the Layer 3 peer-router on both the vPC peers, to form peering adjacency with both the peers:

switch(config-vpc-domain)# **layer3 peer-router**

Note If you enable the layer 3 peer-router without enabling the peer-gateway feature, the following syslog message is displayed:

```
"Please enable peer-gateway before configuring this feature."
```

Step 5 Exit the vpc-domain configuration mode:

switch(config-vpc-domain)# **end**

Step 6 (Optional) Verify the configuration by displaying brief information about each vPC domain:

switch# **show vpc brief**

Example

This example shows how to configure Layer 3 over vPC:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# peer-gateway
switch(config-vpc-domain)# layer3 peer-router
switch(config-vpc-domain)# end
```

This example shows how to verify if the Layer 3 over vPC 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
```

Isolating and Restoring a Switch from the vPC Complex

Configuring vPC Shutdown

Isolates a switch from the vPC complex. Once the switch is isolated, the switch can be debugged, reloaded, or even removed physically, without affecting the vPC traffic going through the non-isolated switch.

Note When vPC switches are isolated, configuration changes are not allowed on both the isolated and non-isolated switches.

Only a disruptive upgrade is supported on an isolated switch.

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 switch and enters the vpc-domain configuration mode. There is no default domain-id. The range is from 1 to 1000.
Step 3	switch(config-vpc)# **shutdown**	Isolates the switch from the vPC domain.
Step 4	switch(config-vpc)# **copy running-config startup-config**	Saves the running configuration to the startup configuration file so that all current configuration details are available after a reboot.

Example

This example shows how to shutdown traffic on vPC domain 100.

```
switch# configure terminal
switch(config)# vpc domain 100
switch(config-vpc)# shutdown
switch(config-vpc)# copy running-config startup-config
```

Restoring a vPC Shutdown Switch

Brings an isolated switch back into the vPC complex with minimal disruption.

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 switch and enters the vpc-domain configuration mode. There is no default domain-id. The range is from 1 to 1000.
Step 3	switch(config-vpc)# **no shutdown**	Restores the switch to the vPC domain.
Step 4	switch(config-vpc)# **copy running-config startup-config**	Saves the running configuration to the startup configuration file so that all current configuration details are available after a reboot.

Example

This example shows how to restore traffic on vPC domain 100.

```
switch# configure terminal
switch(config)# vpc domain 100
switch(config-vpc)# no shutdown
switch(config-vpc)# copy running-config startup-config
```

Verifying the vPC Configuration

Use the following commands to display vPC configuration information:

Command	Purpose
switch# **show feature**	Displays whether vPC is enabled or not.
switch# **show port-channel capacity**	Displays how many EtherChannels are configured and how many are still available on the switch.
switch# **show running-config vpc**	Displays running configuration information for vPCs.
switch# **show vpc brief**	Displays brief information on the vPCs.
switch# **show vpc consistency-parameters**	Displays the status of those parameters that must be consistent across all vPC interfaces.
switch# **show vpc peer-keepalive**	Displays information on the peer-keepalive messages.
switch# **show vpc role**	Displays the peer status, the role of the local switch, the vPC system MAC address and system priority, and the MAC address and priority for the local vPC switch.

Command	Purpose
switch# **show vpc statistics**	Displays statistics on the vPCs. **Note** This command displays the vPC statistics only for the vPC peer device that you are working on.

For information about the switch output, see the Command Reference for your Cisco Nexus Series switch.

Viewing the Graceful Type-1 Check Status

This example shows how to display the current status of the graceful Type-1 consistency check:

```
switch# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                    : 10
Peer status                      : peer adjacency formed ok
vPC keep-alive status            : peer is alive
Configuration consistency status: success
Per-vlan consistency status      : success
Type-2 consistency status        : success
vPC role                         : secondary
Number of vPCs configured        : 34
Peer Gateway                     : Disabled
Dual-active excluded VLANs       : -
Graceful Consistency Check       : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1
```

Viewing a Global Type-1 Inconsistency

When a global Type-1 inconsistency occurs, the vPCs on the secondary switch are brought down. The following example shows this type of inconsistency when there is a spanning-tree mode mismatch.

The example shows how to display the status of the suspended vPC VLANs on the secondary switch:

```
switch(config)# show vpc
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                    : 10
Peer status                      : peer adjacency formed ok
vPC keep-alive status            : peer is alive
Configuration consistency status: failed
Per-vlan consistency status      : success
Configuration consistency reason: vPC type-1 configuration incompatible - STP
                                  Mode inconsistent
Type-2 consistency status        : success
vPC role                         : secondary
Number of vPCs configured        : 2
Peer Gateway                     : Disabled
Dual-active excluded VLANs       : -
Graceful Consistency Check       : Enabled
```

```
vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1-10

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        down*  failed      Global compat check failed -
30     Po30        down*  failed      Global compat check failed -
```

The example shows how to display the inconsistent status (the VLANs on the primary vPC are not suspended) on the primary switch:

```
switch(config)# show vpc
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                   : 10
Peer status                     : peer adjacency formed ok
vPC keep-alive status           : peer is alive
Configuration consistency status: failed
Per-vlan consistency status     : success
Configuration consistency reason: vPC type-1 configuration incompatible - STP Mo
de inconsistent
Type-2 consistency status       : success
vPC role                        : primary
Number of vPCs configured       : 2
Peer Gateway                    : Disabled
Dual-active excluded VLANs      : -
Graceful Consistency Check      : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1-10

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     failed      Global compat check failed 1-10
30     Po30        up     failed      Global compat check failed 1-10
```

Viewing an Interface-Specific Type-1 Inconsistency

When an interface-specific Type-1 inconsistency occurs, the vPC port on the secondary switch is brought down while the primary switch vPC ports remain up.The following example shows this type of inconsistency when there is a switchport mode mismatch.

This example shows how to display the status of the suspended vPC VLAN on the secondary switch:

```
switch(config-if)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                   : 10
```

```
Peer status                     : peer adjacency formed ok
vPC keep-alive status           : peer is alive
Configuration consistency status: success
Per-vlan consistency status     : success
Type-2 consistency status       : success
vPC role                        : secondary
Number of vPCs configured       : 2
Peer Gateway                    : Disabled
Dual-active excluded VLANs      : -
Graceful Consistency Check      : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     success     success                    1
30     Po30        down*  failed      Compatibility check failed -
                                       for port mode
```

This example shows how to display the inconsistent status (the VLANs on the primary vPC are not suspended) on the primary switch:

```
switch(config-if)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                   : 10
Peer status                     : peer adjacency formed ok
vPC keep-alive status           : peer is alive
Configuration consistency status: success
Per-vlan consistency status     : success
Type-2 consistency status       : success
vPC role                        : primary
Number of vPCs configured       : 2
Peer Gateway                    : Disabled
Dual-active excluded VLANs      : -
Graceful Consistency Check      : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     success     success                    1
30     Po30        up     failed      Compatibility check failed 1
                                       for port mode
```

Viewing a Per-VLAN Consistency Status

To view the per-VLAN consistency or inconsistency status, enter the **show vpc consistency-parameters vlans** command.

Example

This example shows how to display the consistent status of the VLANs on the primary and the secondary switches.

```
switch(config-if)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                   : 10
Peer status                     : peer adjacency formed ok
vPC keep-alive status           : peer is alive
Configuration consistency status: success
Per-vlan consistency status     : success
Type-2 consistency status       : success
vPC role                        : secondary
Number of vPCs configured       : 2
Peer Gateway                    : Disabled
Dual-active excluded VLANs      : -
Graceful Consistency Check      : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1-10

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     success     success                    1-10
30     Po30        up     success     success                    1-10
```

Entering **no spanning-tree vlan 5** command triggers the inconsistency on the primary and secondary VLANs:

```
switch(config)# no spanning-tree vlan 5
```

This example shows how to display the per-VLAN consistency status as Failed on the secondary switch:

```
switch(config)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                   : 10
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                    : Disabled
Dual-active excluded VLANs      : -
```

```
Graceful Consistency Check      : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1-4,6-10

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     success     success                    1-4,6-10
30     Po30        up     success     success                    1-4,6-10
```

This example shows how to display the per-VLAN consistency status as Failed on the primary switch:

```
switch(config)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                   : 10
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                        : primary
Number of vPCs configured       : 2
Peer Gateway                    : Disabled
Dual-active excluded VLANs      : -
Graceful Consistency Check      : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1-4,6-10

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     success     success                    1-4,6-10
30     Po30        up     success     success                    1-4,6-10
```

This example shows the inconsistency as STP Disabled:

```
switch(config)# show vpc consistency-parameters vlans

Name                        Type  Reason Code             Pass Vlans

-------------               ----  ---------------------- -----------------------
STP Mode                    1     success                0-4095
STP Disabled                1     vPC type-1             0-4,6-4095
                                  configuration
                                  incompatible - STP is
                                  enabled or disabled on
                                   some or all vlans
STP MST Region Name         1     success                0-4095
STP MST Region Revision     1     success                0-4095
STP MST Region Instance to  1     success                0-4095
 VLAN Mapping
STP Loopguard               1     success                0-4095
STP Bridge Assurance        1     success                0-4095
STP Port Type, Edge         1     success                0-4095
BPDUFilter, Edge BPDUGuard
STP MST Simulate PVST       1     success                0-4095
Pass Vlans                  -                            0-4,6-4095
```

Viewing Dual Active Detection Status

When MCT (Multichassis EtherChannel Trunk) is down and keepalive is up, dual active situation arises. The dual active detection status is set to 1 on operational secondary device.

This example displays dual active detection status on operational secondary device:

```
switch# show vpc role
vPC Role status
----------------------------------------------------
vPC role : primary, operational secondary
Dual Active Detection Status : 1
vPC system-mac : 00:23:04:ee:be:01
vPC system-priority : 32667
vPC local system-mac : 8c:60:4f:17:e6:41
vPC local role-priority : 3000
Leaf-ACC-4#
```

This example displays the dual active detection status on operational primary device:

```
switch# show vpc role
vPC Role status
----------------------------------------------------
vPC role : secondary, operational primary
Dual Active Detection Status : 0
vPC system-mac : 00:23:04:ee:be:01
vPC system-priority : 32667
vPC local system-mac : 8c:60:4f:17:e5:fc
vPC local role-priority : 4000
```

vPC Default Settings

The following table lists the default settings for vPC parameters.

Table 8: Default vPC Parameters

Parameters	Default
vPC system priority	32667
vPC peer-keepalive message	Disabled
vPC peer-keepalive interval	1 second
vPC peer-keepalive timeout	5 seconds
vPC peer-keepalive UDP port	3200

CHAPTER 7

Configuring Linecard Expansion Modules

This chapter contains the following sections:

- Configuring Breakout , on page 157

Configuring Breakout

Information About Linecard Expansion Modules

The Linecard Expansion Module (LEM) is a field replaceable module. LEM is supported only on the Cisco Nexus 5696Q Series model. There are 8 LEM slots on the device and each LEM slot has 12 to 40 Gigabit Ethernet ports that can break out into 48 to 10 Gigabit Ethernet ports per LEM. The module can be either in 10 Gigabit Ethernet mode or in a 40 Gigabit Ethernet mode. A power-off followed by a power-on of the module is required to change the mode. The Cisco Nexus 5696Q Series model also supports a 100 Gigabit Ethernet LEM, but the 100 Gigabit Ethernet LEM does not support the Breakout feature.

Configuring Breakout in a Port

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface breakout slot** *slot* **port** *port-range* **map 10g-4x**	Configures the breakout feature in a port. *slot*—Slot number of port depending on the chassis model. Valid values are from 1 to 8. *port-range*—Single port or range of ports on which breakout is configured. Valid values are from 1 to 48.

	Command or Action	Purpose
		Note On the Cisco Nexus 56128 Switch, you can configure group of two ports. For example, 49-50, 51-52, or 25-26. On the Cisco Nexus 5672UP Switch, you can configure group of three ports. For example, 1-3, or 4-6. On the Cisco Nexus 5624Q Switch, you can configure group of three ports beginning with 1-3, 4-6, 7-9, and 10-12. You cannot enter a port-range of 2-4 or 8-10.
Step 3	Required: switch(config)# **poweroff module** *module-number*	Power-off and power-on the module for the interface breakout to be effective. If the breakout feature is configured on the baseboard module, save the running configuration to startup configuration using the **copy running-config startup-config** command and then reload the switch. **Note** If the breakout feature is not configured on the baseboard module, then an additional reload is required.
Step 4	Required: switch(config)# **no poweroff module** *module-number*	Brings the module up.
Step 5	Required: switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a breakout on the Cisco Nexus 5600 Series Switch in a 10 Gigabit Ethernet mode of operation.

```
switch# configure terminal
switch(config)# interface breakout slot 1 port 1-48 map 10g-4x
switch(config)# poweroff module 1
switch(config)# no poweroff module 1
switch(config)# copy running-config startup-config
```

Removing the Breakout Configuration

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no interface breakout slot** *slot* **port** *port-range* **map 10g-4x**	Removes the breakout configurations for a port module and returns the interface to 40 Gigabit Ethernet mode of operation. *slot*—Slot number of the module depending on the chassis model. Valid values are from 1 to 8. **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	Required: switch(config)# **poweroff module** *module-number*	Power-off and power-on the module for the interface breakout to be effective. If the breakout feature is configured on baseboard module, save the running configuration to startup configuration using the **copy running-config startup-config** command and then reload the switch. **Note** If the breakout feature is not configured on the baseboard module, then an additional reload is required.
Step 4	Required: switch(config)# **no poweroff module** *module-number*	Brings the module up.
Step 5	Required: switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

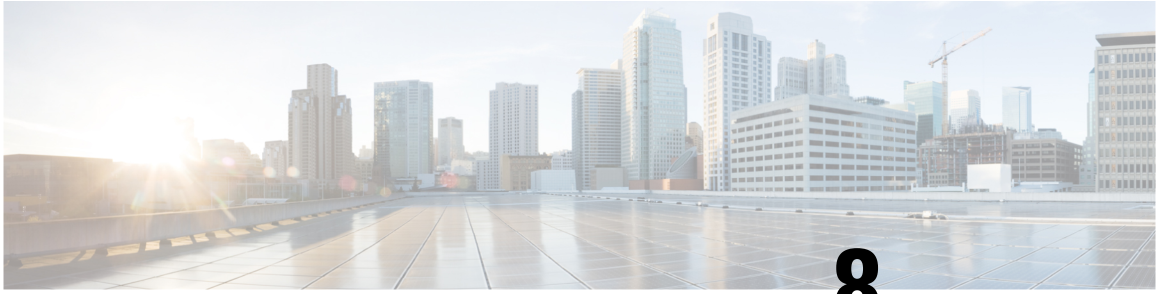
This example shows how to remove the breakout configuration in a port and return to the 40 Gigabit Ethernet mode of operation on the Cisco Nexus 5600 Series Switch.

```
switch# configure terminal
switch(config)# no interface breakout slot 1 port 1-48 map 10g-4x
switch(config)# poweroff module 1
switch(config)# no poweroff module 1
switch(config)# 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.

Command	Purpose
show interface eth1/2 capabilities	Displays information about the interface configuration.
show interface brief	Displays a brief summary of the interface configuration.



CHAPTER 8

Configuring Q-in-Q VLAN Tunnels

This chapter contains the following sections:

- Information About Q-in-Q VLAN Tunnels, on page 161
- Information About Layer 2 Protocol Tunneling, on page 164
- Licensing Requirements for Q-in-Q Tunnels, on page 166
- Guidelines and Limitations for Q-in-Q VLAN Tunnels, on page 166
- Configuring Q-in-Q Tunnels and Layer 2 Protocol Tunneling , on page 168
- Verifying the Q-in-Q Configuration, on page 173
- Configuration Examples for Q-in-Q and Layer 2 Protocol Tunneling, on page 174

Information About Q-in-Q VLAN Tunnels

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.

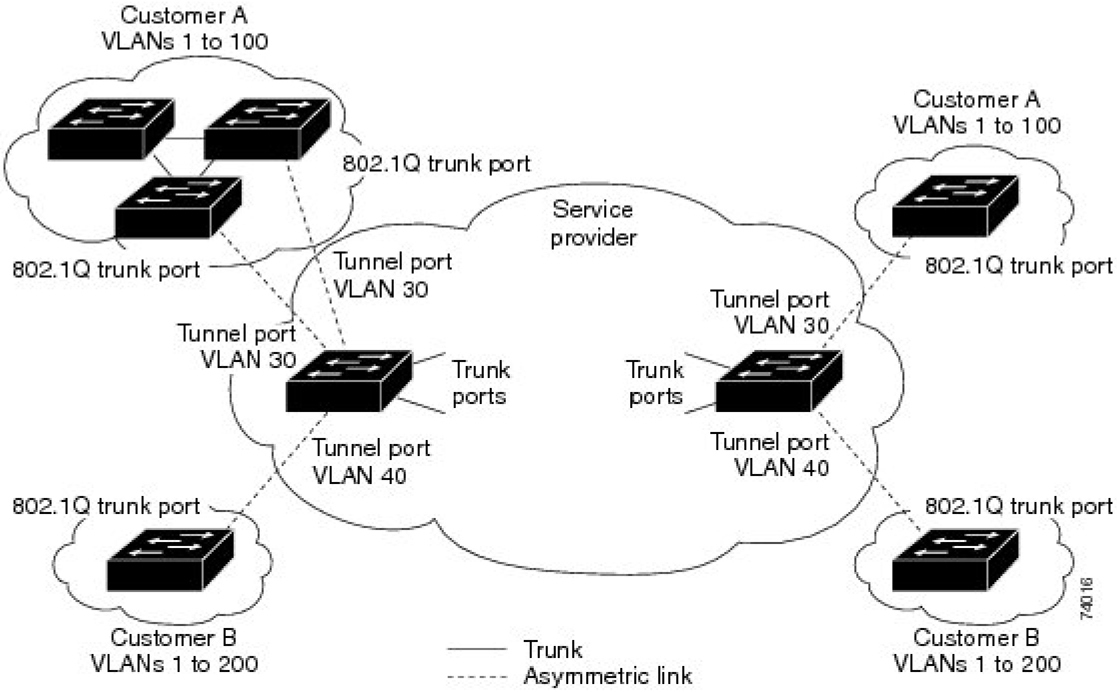
Using the 802.1Q tunneling feature, service providers can use a single VLAN to support customers who have multiple VLANs. Customer VLAN IDs are preserved and the traffic from different customers is segregated within the service-provider infrastructure even when they appear to be on the same VLAN.

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.

Note Selective Q-in-Q tunneling is not supported. All frames that enter the tunnel port are subject to Q-in-Q tagging.

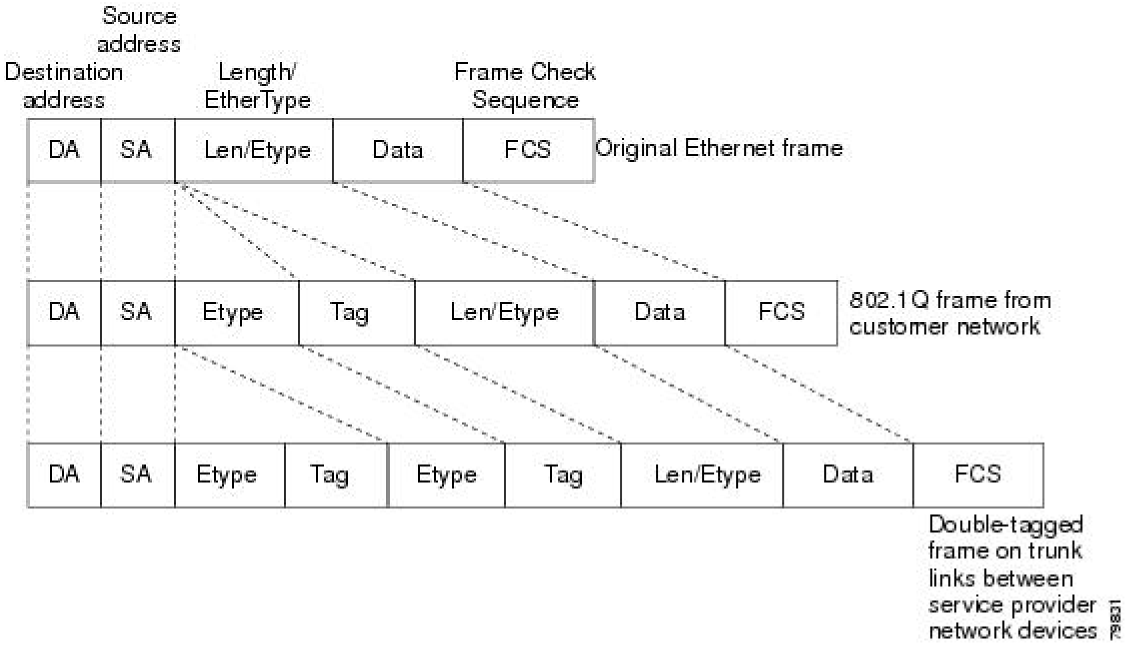
Figure 23: 802.1Q-in-Q Tunnel Ports



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 following figure.

Figure 24: 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.

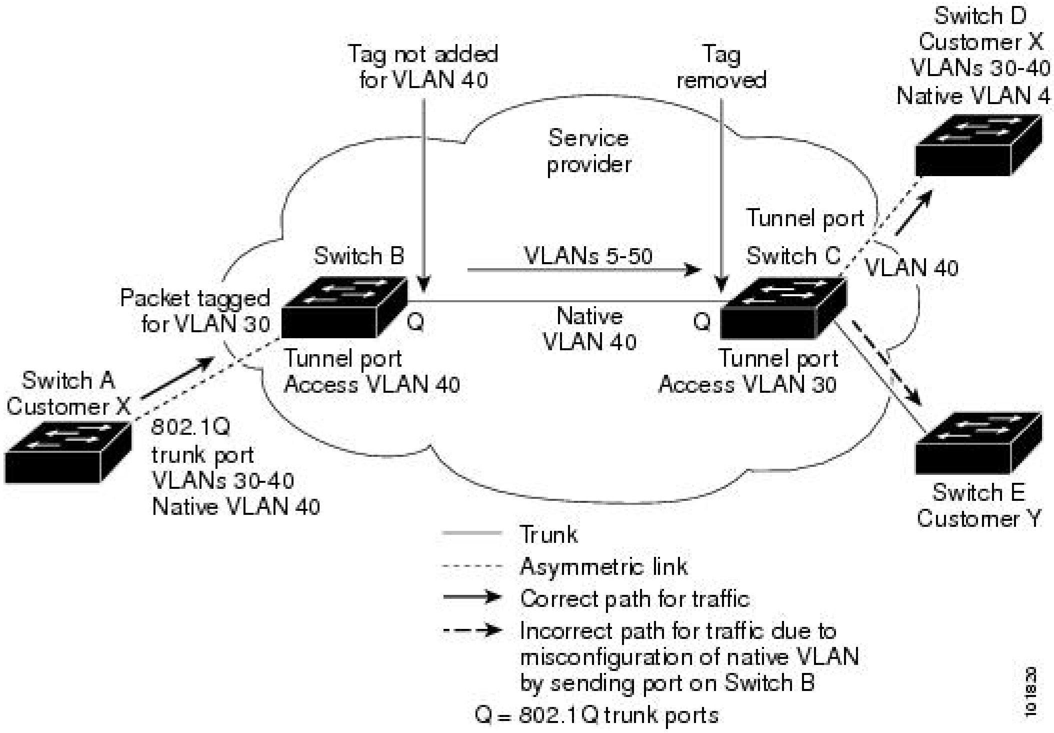
Note Hierarchical tagging, or multi-level dot1q tagging Q-in-Q, is not supported.

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 following figure, 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 25: 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.

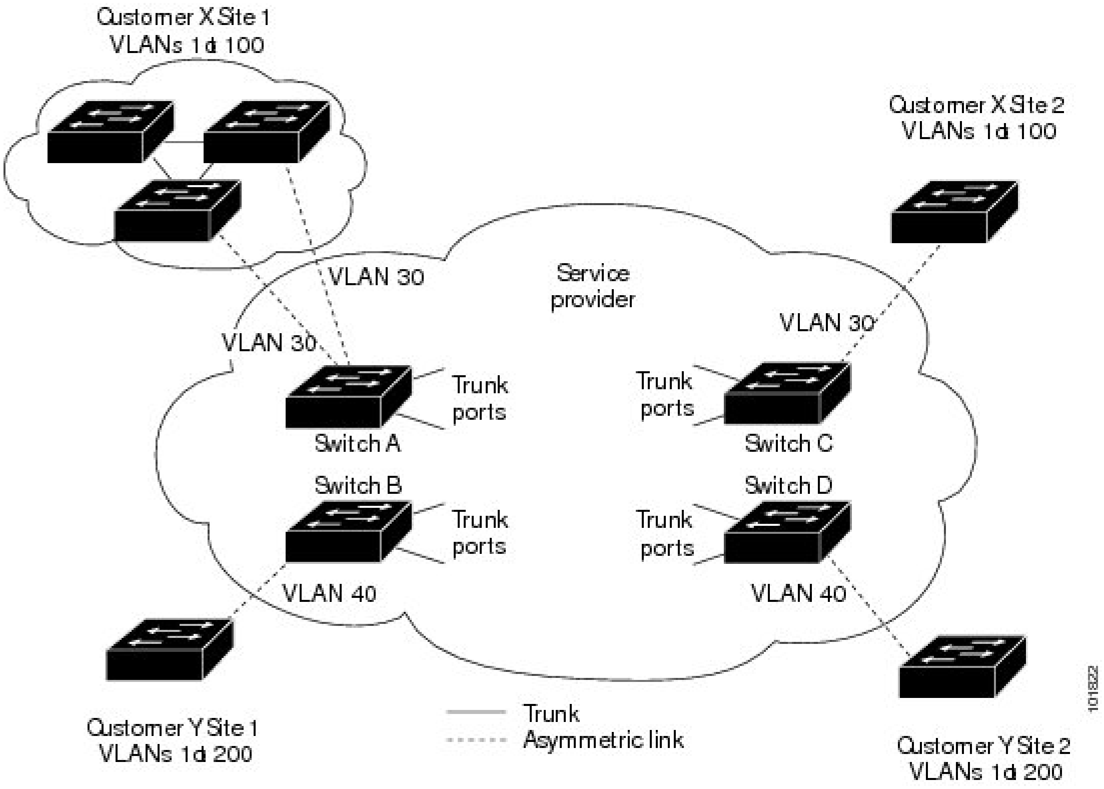
Bridge protocol data units (BPDUs) for CDP, STP, or VTP cross the service-provider infrastructure and are delivered to customer switches on the outbound side of the service-provider network. Identical packets are received by all customer ports on the same VLANs.

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 supervisorCPU. See the "Configuring the Rate Limit for Layer 2 Protocol Tunnel Ports" section on page 9-14.

For example, in the following figure, 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.

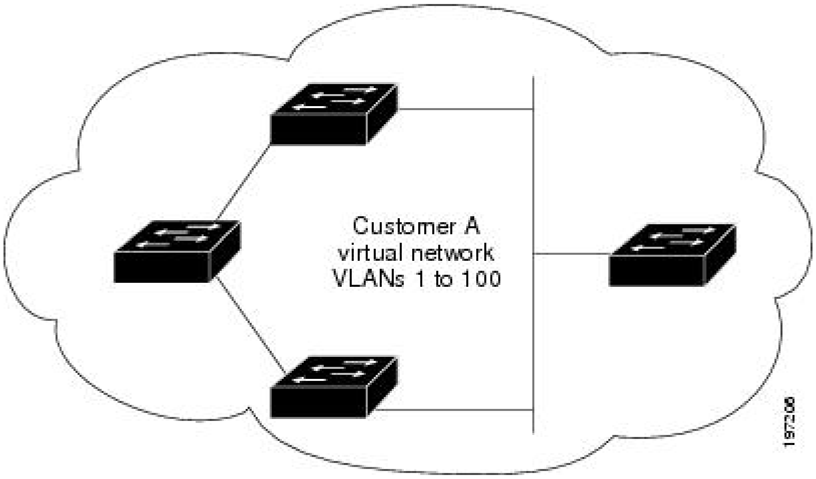
Figure 26: Layer 2 Protocol Tunneling



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 following figure shows the resulting topology on the customer's network when BPDU tunneling is not enabled.

Figure 27: Virtual Network Topology Without BPDU Tunneling



Licensing Requirements for Q-in-Q Tunnels

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	802.1Q-in-Q VLAN tunneling and L2 protocol tunneling require no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the Cisco NX-OS Licensing Guide.

Guidelines and Limitations for Q-in-Q VLAN 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 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 VLANs 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 on the trunk side of the service provider ports (not the dot1q-tunnel port) to prevent any native VLAN mis-configurations.
- Within a same forwarding instance, if dot1q tunnel configured on multiple ports is unconfigured then these ports go into an error-disabled state. The ports have to be flapped a couple of times to recover from the error-disable state.
- 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.
- STP is not be supported on inner VLAN.
- No loop detection mechanism in the fabric.
- Cisco Discovery Protocol (CDP) is incompatible with Q-in-Q. When a port is configured as an 802.1Q tunnel port, CDP must be disabled on the interface.
- Quality of Service (QoS) cannot detect the received Class of Service (CoS) value in the 802.1Q 2-byte Tag Control Information field.
- On an asymmetrical link, CDP reports a native VLAN mismatch if the VLAN of the tunnel port does not match the native VLAN of the .1Q trunk. The .1Q tunnel feature does not require that the VLANs match. Ignore the messages in this configuration.
- Layer 2 portchannels can be configured as Tunnel ports, but all the members of the port channel must to be connected to the same service provider edge switch.
- In break-out configurations, all the ports in the same forwarding instance are configured with dot1qtunnel on an all/none basis. Ports lacking a dot1q tunnel configuration are brought down and err-disabled.
- All ports on a FEX are configured as all/none in dot1qtunnel. Those missing dot1qtunnel stay error disabled.
-
- All members of a port-channel propagate runtime configurations to other ports belonging to same fwm block.
- Configuring Dot1qtunnel on a FEX HIF on one switch of an AA FEX topology, resets that FEX on the other switch before bringing it online.
- Q-in-Q VLAN tunnels are supported only on Cisco Nexus 6000 and Cisco Nexus 5600 platforms.
- You cannot configure Layer 2 protocol features to forward spanning tree protocol (STP) bridge protocol data units (BPDU)'s or cisco discovery protocol (CDP) packets across the Q-in-Q VLAN tunnels.

Configuring Q-in-Q Tunnels and Layer 2 Protocol Tunneling

Creating an 802.1Q Tunnel Port

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**	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)# **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**	Saves the change persistently through reboots and restarts by copying 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

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). This is an optional 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.
Step 4	(Optional) switch(config-if)# **switchport dot1q ethertype** *value*	Sets the EtherType for the Q-in-Q tunnel on the port.
Step 5	switch(config-if)# **no switchport dot1q ethertype**	(Optional) 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)# **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)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example show 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

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	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.

	Command or Action	Purpose
Step 10	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable 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 Layer 2 Protocol Tunnel Ports

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **l2protocol tunnel cos** *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** *cos-value*	Sets the global CoS value to default.
Step 4	**exit**	Exits configuration mode.
Step 5	(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 6	(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 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to 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, you might need to adjust this value to prevent STP errors on your 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-second*	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	switch(config)# **no hardware rate-limiter layer-2 l2pt** *packets-per-second*	Resets the threshold values to the default of 500 packets per second.

Configuring Thresholds for Layer 2 Protocol Tunnel Ports

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	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,

	Command or Action	Purpose
		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**] *packets-per-sec*	Resets the threshold values to 0 and disables the shutdown threshold.
Step 7	switch(config-if)# **l2protocol tunnel shutdown-threshold** [**cdp** | **stp** | **vtp**]	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)# **no 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.
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**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Verifying the Q-in-Q Configuration

To display Q-in-Q tunnel and Layer 2 protocol tunneling configuration information, perform one of the following tasks:

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.

Command	Purpose
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.

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#
```