Cisco Nexus 3000 Series NX-OS Interfaces Configuration Guide, Release 5.0(3)U5(1)

First Published: August 30, 2012
Last Modified: October 24, 2012

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

Text Part Number: OL-26623-03
CONTENTS

Preface ix

Audience ix

Document Conventions ix

Related Documentation for Nexus 3000 Series NX-OS Software x

Documentation Feedback xii

Obtaining Documentation and Submitting a Service Request xii

CHAPTER 1

New and Changed Information for this Release 1

New and Changed Information in this Release 1

CHAPTER 2

Configuring Layer 2 Interfaces 3

Information About Ethernet Interfaces 3

About the Interface Command 3

About the Unidirectional Link Detection Parameter 4

Default UDLD Configuration 5

UDLD Aggressive and Nonaggressive Modes 5

Interface Speed 6

SVI Autostate 6

About the Cisco Discovery Protocol 6

Default CDP Configuration 6

About the Error-Disabled State 7

About MTU Configuration 7

Configuring Ethernet Interfaces 7

Configuring the UDLD Mode 8

Changing an Interface Port Mode 9

Configuring Interface Speed 10

Disabling Link Negotiation 11
Standards for Layer 3 Interfaces 35
Feature History for Layer 3 Interfaces 35

CHAPTER 4
Configuring Port Channels 37
Information About Port Channels 37
Understanding Port Channels 37
Compatibility Requirements 38
Load Balancing Using Port Channels 40
Understanding LACP 41
   LACP Overview 41
   LACP ID Parameters 42
   Channel Modes 43
   LACP Marker Responders 44
   LACP-Enabled and Static Port Channel Differences 44
   LACP Port Channel MinLinks 44
Configuring Port Channels 45
   Creating a Port Channel 45
   Adding a Port to a Port Channel 45
   Configuring Load Balancing Using Port Channels 46
   Configuring Hardware Hashing for Multicast Traffic 47
Enabling LACP 48
   Configuring the Channel Mode for a Port 48
   Configuring LACP Port Channel MinLinks 49
   Configuring the LACP Fast Timer Rate 50
   Configuring the LACP System Priority and System ID 51
   Configuring the LACP Port Priority 52
Verifying Port Channel Configuration 52
Verifying the Load-Balancing Outgoing Port ID 53
Feature History for Port Channels 54

CHAPTER 5
Configuring IP Tunnels 55
Information About IP Tunnels 55
   GRE Tunnels 56
Licensing Requirements for IP Tunnels 56
Prerequisites for IP Tunnels 56
Chapter 6: Configuring Virtual Port Channels

Information About vPCs
- vPC Overview
- Terminology
  - vPC Terminology
Supported vPC Topologies
- Cisco Nexus 3000 Series Switch vPC Topology
vPC Domain
Peer-Keepalive Link and Messages
Compatibility Parameters for vPC Peer Links
  - Configuration Parameters That Must Be Identical
  - Configuration Parameters That Should Be Identical
Graceful Type-1 Check
Per-VLAN Consistency Check
vPC Auto-Recovery
vPC Peer Links
  - vPC Peer Link Overview
vPC Number
vPC Interactions with Other Features
  - vPC and LACP
  - vPC Peer Links and STP
CFSoE
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guidelines and Limitations for vPCs</td>
<td>72</td>
</tr>
<tr>
<td>Configuring vPCs</td>
<td>73</td>
</tr>
<tr>
<td>Enabling vPCs</td>
<td>73</td>
</tr>
<tr>
<td>Disabling vPCs</td>
<td>73</td>
</tr>
<tr>
<td>Creating a vPC Domain</td>
<td>74</td>
</tr>
<tr>
<td>Configuring a vPC Keepalive Link and Messages</td>
<td>75</td>
</tr>
<tr>
<td>Creating a vPC Peer Link</td>
<td>77</td>
</tr>
<tr>
<td>Checking the Configuration Compatibility</td>
<td>78</td>
</tr>
<tr>
<td>Enabling vPC Auto-Recovery</td>
<td>79</td>
</tr>
<tr>
<td>Configuring the Restore Time Delay</td>
<td>79</td>
</tr>
<tr>
<td>Excluding VLAN Interfaces From Shutdown When vPC Peer Link Fails</td>
<td>80</td>
</tr>
<tr>
<td>Configuring the VRF Name</td>
<td>81</td>
</tr>
<tr>
<td>Binding a VRF Instance to a vPC</td>
<td>81</td>
</tr>
<tr>
<td>Moving Other Port Channels into a vPC</td>
<td>82</td>
</tr>
<tr>
<td>Manually Configuring a vPC Domain MAC Address</td>
<td>83</td>
</tr>
<tr>
<td>Manually Configuring the System Priority</td>
<td>83</td>
</tr>
<tr>
<td>Manually Configuring a vPC Peer Switch Role</td>
<td>84</td>
</tr>
<tr>
<td>Verifying the vPC Configuration</td>
<td>85</td>
</tr>
<tr>
<td>Viewing The Graceful Type-1 Check Status</td>
<td>86</td>
</tr>
<tr>
<td>Viewing A Global Type-1 Inconsistency</td>
<td>86</td>
</tr>
<tr>
<td>Viewing An Interface-Specific Type-1 Inconsistency</td>
<td>87</td>
</tr>
<tr>
<td>Viewing a Per-VLAN Consistency Status</td>
<td>88</td>
</tr>
<tr>
<td>vPC Default Settings</td>
<td>90</td>
</tr>
</tbody>
</table>
Preface

This preface contains the following sections:

- Audience, page ix
- Document Conventions, page ix
- Related Documentation for Nexus 3000 Series NX-OS Software, page x
- Documentation Feedback, page xii
- Obtaining Documentation and Submitting a Service Request, page xii

Audience

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

Document Conventions

Command descriptions use the following conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>Bold text indicates the commands and keywords that you enter literally as shown.</td>
</tr>
<tr>
<td><em>Italic</em></td>
<td>Italic text indicates arguments for which the user supplies the values.</td>
</tr>
<tr>
<td>[x]</td>
<td>Square brackets enclose an optional element (keyword or argument).</td>
</tr>
<tr>
<td>[x</td>
<td>y]</td>
</tr>
<tr>
<td>{x</td>
<td>y}</td>
</tr>
<tr>
<td>Convention</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>[x {y</td>
<td>z}]</td>
</tr>
<tr>
<td></td>
<td>choices within optional or required elements. Braces and a vertical bar</td>
</tr>
<tr>
<td></td>
<td>within square brackets indicate a required choice within an optional</td>
</tr>
<tr>
<td></td>
<td>element.</td>
</tr>
<tr>
<td>variable</td>
<td>Indicates a variable for which you supply values, in context where italics</td>
</tr>
<tr>
<td></td>
<td>cannot be used.</td>
</tr>
<tr>
<td>string</td>
<td>A nonquoted set of characters. Do not use quotation marks around the</td>
</tr>
<tr>
<td></td>
<td>string or the string will include the quotation marks.</td>
</tr>
</tbody>
</table>

Examples use the following conventions:

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

This document uses the following conventions:

**Note**

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

**Caution**

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

---

**Related Documentation for Nexus 3000 Series NX-OS Software**

The entire Cisco NX-OS 3000 Series documentation set is available at the following URL:

Release Notes
The release notes are available at the following URL:

Installation and Upgrade Guides
The installation and upgrade guides are available at the following URL:
The documents in this category include:

- Cisco Nexus 5000 Series, Cisco Nexus 3000 Series, and Cisco Nexus 2000 Series Safety Information and Documentation
- Regulatory, Compliance, and Safety Information for the Cisco Nexus 5000 Series, Cisco Nexus 3000 Series, and Cisco Nexus 2000 Series
- Cisco Nexus 3000 Series Hardware Installation Guide

License Information

For the NX-OS end user agreement and copyright information, see License and Copyright Information for Cisco NX-OS Software, available at the following URL: http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_0/nx-os/license_agreement/nx-ossw_lisns.html.

Configuration Guides
The configuration guides are available at the following URL:
The documents in this category include:

- Fundamentals Configuration Guide
- Interfaces Configuration Guide
- Layer 2 Switching Configuration Guide
- Multicast Configuration Guide
- Quality of Service Configuration Guide
- Security Configuration Guide
- System Management Configuration Guide
- Unicast Routing Configuration Guide
- Verified Scalability Guide for Cisco NX-OS

Technical References
The technical references are available at the following URL:

Error and System Messages
The error and system message reference guides are available at the following URL:

Documentation Feedback
To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com. We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request
For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly What's New in Cisco Product Documentation, which also lists all new and revised Cisco technical documentation, at:
Subscribe to the What's New in Cisco Product Documentation as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.
New and Changed Information for this Release

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

- New and Changed Information in this Release, page 1

New and Changed Information in this Release

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

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Added or Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port-channel load-balancing hash</td>
<td>Added support for port-channel load balancing resolution.</td>
<td>5.0(3)U5(1)</td>
<td>Configuring Port Channels, on page 37</td>
</tr>
<tr>
<td>SVI Autostate disable</td>
<td>Allows user to keep a SVI up even when no interfaces are up in the corresponding VLAN.</td>
<td>5.0(3)U5(1)</td>
<td>Configuring Layer 2 Interfaces, on page 3</td>
</tr>
</tbody>
</table>
CHAPTER 2

Configuring Layer 2 Interfaces

This chapter contains the following sections:

- Information About Ethernet Interfaces, page 3
- Configuring Ethernet Interfaces, page 7
- Displaying Interface Information, page 18
- Displaying Input Packet Discard Information, page 20
- Default Physical Ethernet Settings, page 21
- MIBs for Layer 2 Interfaces, page 22

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.

About the 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).
About the 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, as long as autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally at Layer 1, 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.

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

<table>
<thead>
<tr>
<th>Feature</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDLD global enable state</td>
<td>Globally disabled</td>
</tr>
<tr>
<td>UDLD aggressive mode</td>
<td>Disabled</td>
</tr>
<tr>
<td>UDLD per-port enable state for fiber-optic media</td>
<td>Enabled on all Ethernet fiber-optic LAN ports</td>
</tr>
<tr>
<td>UDLD per-port enable state for twisted-pair (copper) media</td>
<td>Disabled on all Ethernet 10/100 and 1000BASE-TX LAN ports</td>
</tr>
</tbody>
</table>

**UDLD Aggressive and Nonaggressive Modes**

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

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

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

- One side of a link has a port stuck (both transmission and receive)
- One side of a link remains up while the other side of the link is down
In these cases, the UDLD aggressive mode disables one of the ports on the link, which prevents traffic from being discarded.

## Interface Speed

Cisco Nexus 3000 Series switches have a number of fixed 10-Gigabit ports, each equipped with SFP+ interface adapters.

## SVI Autostate

The Switch Virtual Interface (SVI) represents a logical interface between the bridging function and the routing function of a VLAN in the device. By default, when a VLAN interface has multiple ports in the VLAN, the SVI goes to the down state when all the ports in the VLAN go down.

Autostate behavior is the operational state of an interface that is governed by the state of the various ports in its corresponding vlan. In other words a SVI interface on a VLAN comes up when there is at least one port in that vlan that is in STP forwarding state. Similarly, this interface goes down when the last STP forwarding port goes down or goes to another STP state.

By default, Autostate calculation is enabled. You can disable Autostate calculation for a SVI interface and change the default value.

## About the 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 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>
<thead>
<tr>
<th>Feature</th>
<th>Default Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDP interface state</td>
<td>Enabled</td>
</tr>
<tr>
<td>Feature</td>
<td>Default Setting</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>CDP timer (packet update frequency)</td>
<td>60 seconds</td>
</tr>
<tr>
<td>CDP holdtime (before discarding)</td>
<td>180 seconds</td>
</tr>
<tr>
<td>CDP Version-2 advertisements</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

### About the Error-Disabled State

An interface is in the error-disabled (err-disabled) state when the interface 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 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.

### Configuring Ethernet Interfaces

The section includes the following topics:
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.

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

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>switch(config)# feature udld</td>
<td>Enables UDLD for the device.</td>
</tr>
<tr>
<td>3</td>
<td>switch(config)# no feature udld</td>
<td>Disables UDLD for the device.</td>
</tr>
<tr>
<td>4</td>
<td>switch(config)# show udld global</td>
<td>Displays the UDLD status for the device.</td>
</tr>
<tr>
<td>5</td>
<td>switch(config)# interface type slot/port</td>
<td>Specifies an interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td>6</td>
<td>switch(config-if)# udld {enable</td>
<td>disable</td>
</tr>
<tr>
<td>7</td>
<td>switch(config-if)# show udld interface</td>
<td>Displays the UDLD status for the interface.</td>
</tr>
</tbody>
</table>

This example shows how to enable the 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
```
This example shows how to disable UDLD for the switch:

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

## Changing an Interface Port Mode

You can configure a Quad small form-factor pluggable (QSFP+) port by using the `hardware profile portmode` command. To restore the defaults, use the `no` form of this command.

### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>switch(config)# copy running-config bootflash: my-config.cfg</td>
<td>Copies the running configuration to the bootflash. You can use this file to configure your device later.</td>
</tr>
<tr>
<td>3</td>
<td>switch(config)# write erase</td>
<td>Removes all the interface configurations.</td>
</tr>
<tr>
<td>4</td>
<td>switch(config)# write erase</td>
<td>Reloads the Cisco NX-OS software.</td>
</tr>
<tr>
<td>5</td>
<td>switch(config)# [no] hardware profile portmode portmode</td>
<td>Changes the interface port mode.</td>
</tr>
<tr>
<td>6</td>
<td>switch(config)# hardware profile portmode portmode 2-tuple</td>
<td>(Optional) Displays the port names in 2-tuple mode instead of the default 3-tuple convention mode.</td>
</tr>
<tr>
<td>7</td>
<td>switch(config)# copy running-config startup-config</td>
<td>(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>8</td>
<td>switch(config)# reload</td>
<td>Reloads the Cisco NX-OS software. Manually apply all the interface configuration. You can refer to the configuration file that you saved earlier.</td>
</tr>
</tbody>
</table>

**Note** The interface numbering changes if the ports are changed from 40G mode to 4x10G mode or vice versa.

This example shows how to change the port mode to 48x10g+4x40g for QSFP+ ports:

```
switch# configure terminal
switch(config) copy running-config bootflash:my-config.cfg
switch(config)# write erase
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
switch(config)# hardware profile portmode 48x10g+4x40g
Warning: This command will take effect only after saving the configuration and reload!
```
Configuring Interface Speed

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

Procedure

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch#</td>
<td>configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>switch(config)#</td>
<td>no hardware profile portmode</td>
<td>Warning: This command will take effect only after saving the configuration and reload! Port configurations could get lost when port mode is changed!</td>
</tr>
</tbody>
</table>

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

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

**Step 3** | switch(config-if)# speed speed
Sets the speed on the interface. This command can only be applied to a physical Ethernet interface. The *speed* argument can be set to one of the following:
- 10 Mbps
- 100 Mbps
- 1 Gbps
- 10Gbps
- automatic

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

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

### Disabling Link Negotiation

You can disable link negotiation using the **no negotiate auto** command. By default, auto-negotiation is enabled on 1-Gigabit ports and disabled on 10-Gigabit ports. By default, auto-negotiation is enabled on the Cisco Nexus 3064 and 3064-X switches and disabled on the Cisco Nexus 3048 switch.

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

**Note**

We do not recommend that you enable auto negotiation on 10-Gigabit ports. Enabling auto-negotiation on 10-Gigabit ports brings the link down. By default, link negotiation is disabled on 10-Gigabit ports.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | switch# configure terminal
Enters 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). |
### Disabling SVI Autostate

You can configure a SVI to remain active even if no interfaces are up in the corresponding VLAN. This enhancement is called Autostate Disable.

When you enable or disable autostate behavior it is applied to all the SVIs in the switch unless you configure autostate per SVI.

#### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# feature interface-vlan</td>
<td>Enables the interface-vlan feature.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config)# system default interface-vlan [no] autostate</td>
<td>Configures the system to enable or disable the Autostate default behavior.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config)# interface vlan interface-vlan-number</td>
<td>(Optional) Creates a VLAN interface. The number range is from 1 to 4094.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>switch(config-if)# [no] autostate</td>
<td>(Optional) Enables or disables Autostate behavior per SVI.</td>
</tr>
</tbody>
</table>
Purpose
Command or Action
Step 6 switch(config)# show interface-vlan interface-vlan
(Optional) Displays the enabled or disabled Autostate behavior of the SVI.
Step 7 switch(config)# copy running-config startup-config
(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to disable the systems Autostate default for all the SVIs on the switch:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# system default interface-vlan no autostate
switch(config)# interface vlan 50
switch(config-if)# no autostate
switch(config)# copy running-config startup-config
```

This example shows enabled autostate configuration:

```
switch(config)# show interface-vlan 2
Vlan2 is down, line protocol is down, autostate enabled
Hardware is EtherSVI, address is 547f.ee40.a17c
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
```

### 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Step 2 switch(config)# [no] cdp advertise {v1</td>
<td>v2}</td>
</tr>
<tr>
<td>Step 3 switch(config)# [no] cdp format device-id {mac-address</td>
<td>serial-number</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# interface type slot/port</td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config-if)# cdp enable</td>
</tr>
<tr>
<td>Step 4</td>
<td>switch(config-if)# no cdp enable</td>
</tr>
</tbody>
</table>

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.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>config t</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch# config t</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Specifies a condition under which to place the interface in an err-disabled state. The default is enabled.</td>
</tr>
<tr>
<td>errdisable detect cause {all</td>
<td>link-flap</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch(config)# errdisable detect cause all</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Brings the interface down administratively. To manually recover the interface from the err-disabled state, enter this command first.</td>
</tr>
<tr>
<td>shutdown</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch(config)# shutdown</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Brings the interface up administratively and enables the interface to recover manually from the err-disabled state.</td>
</tr>
<tr>
<td>no shutdown</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch(config)# no shutdown</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Displays information about err-disabled interfaces.</td>
</tr>
<tr>
<td>show interface status err-disabled</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch(config)# show interface status err-disabled</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>copy running-config startup-config</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

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

```
switch(config)# errdisable detect cause all
switch(config)#
```
Enabling the Error-Disabled Recovery

You can specify the application to bring the interface out of the error-disabled (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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>config t</td>
</tr>
<tr>
<td>Example:</td>
<td>switch#config t</td>
</tr>
<tr>
<td>switch(config)#</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>errdisable recovery cause {all</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)#errdisable recovery cause all</td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>show interface status err-disabled</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)#show interface status err-disabled</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>copy running-config startup-config</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)#copy running-config startup-config</td>
</tr>
</tbody>
</table>

This example shows how to enable err-disabled recovery under all conditions:
```
switch(config)#errdisable recovery cause all
switch(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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>config t</code></td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td></td>
<td><code>Example: switch(config)# config t</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>errdisable recovery interval interval</code></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><code>Example: switch(config)#errdisable recovery interval 32 switch(config-if)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>show interface status err-disabled</code></td>
<td>Displays information about err-disabled interfaces.</td>
</tr>
<tr>
<td></td>
<td><code>Example: switch(config)#show interface status err-disabled</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>copy running-config startup-config</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td></td>
<td><code>Example: switch(config)#copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

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

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

### Configuring the Description Parameter

You can provide textual interface descriptions for the Ethernet ports.

#### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>switch# configure terminal</code></td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>switch(config)# interface type slot/port</code></td>
<td>Enters interface configuration mode for the specified interface.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>switch(config-if)# description text</code></td>
<td>Specifies the description for the interface.</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# interface type slot/port</td>
<td>Enters interface configuration mode for the specified interface.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config-if)# shutdown</td>
<td>Disables the interface.</td>
</tr>
<tr>
<td><strong>Step 4</strong> switch(config-if)# no shutdown</td>
<td>Restarts the interface.</td>
</tr>
</tbody>
</table>

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
```

### Displaying Interface Information

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch# show interface type slot/port</td>
<td>Displays the detailed configuration of the specified interface.</td>
</tr>
<tr>
<td>switch# show interface type slot/port capabilities</td>
<td>Displays detailed information about the capabilities of the specified interface. This option is only available for physical interfaces</td>
</tr>
</tbody>
</table>
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:

```bash
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:

```bash
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-(6qlt),tx-(1p6q0t)
CoS rewrite: no
ToS rewrite: no
SPAN: yes
UDLD: yes
```

---

**Table: Displaying Interface Information**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch# show interface type slot/port transceiver</code></td>
<td>Displays detailed information about the transceiver connected to the specified interface. This option is only available for physical interfaces.</td>
</tr>
<tr>
<td><code>switch# show interface brief</code></td>
<td>Displays the status of all interfaces.</td>
</tr>
<tr>
<td><code>switch# show interface flowcontrol</code></td>
<td>Displays the detailed listing of the flow control settings on all interfaces.</td>
</tr>
</tbody>
</table>
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 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 Intf  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
```

**Displaying Input Packet Discard Information**

Beginning with Cisco NX-OS Release 5.0(3)U2(1), you can get detailed information on what specific condition led to an input discard on a given interface. Use the `show hardware internal interface indiscard-stats front-port x` command to determine the condition that could be potentially responsible for the input discards that are seen on port eth1/x. The switch output shows the discards for IPv4, STP, input policy, ACL specific discard, generic receive drop, and VLAN related discards.

This example shows how to determine the condition that could be potentially responsible for the input discards:

```
switch# show hardware internal interface indiscard-stats front-port 1
+-----------------------------------------+-----------------+----------------+
| Counter Description | Count | |
+-----------------------------------------+-----------------+----------------+
  IPv4 Discards          0          |
  STP Discards           0          |
  Policy Discards        100         |
  ACL Drops              0          |
  Receive Drops          0          |
  Vlan Discards          33         |
```
Counter Information:

- IPv4 Discards--- IPv4 Discards represent errors at the IP layer, for example the IP checksum error.
- STP Discards--- STP Discards are incremented when the receive interface STP state is not forwarding the packets received.
- Policy Discards--- Policy Discards are incremented when there are discards because of input policy on the interface.
- ACL Drops--- ACL drops indicate that incoming packets match an ACL entry with a drop action.
- Receive Drops--- This drop increment represents a condition when no output port is determined for an ingress packet. Receive drops happen because of variety of reasons including IPv4, STP and policy discards. The drop counter increments in conjunction with one of the above counters or separately.
- Vlan Discard--- Vlan Discard indicates vlan-based discards. For example, a vlan tagged packet ingressing on a port which is not a member of the vlan.

This example shows how to clear all the input discard counters which is useful for debugging purposes:

```
Switch# show hardware internal interface indiscard-stats front-port 1 clear
```

Default Physical Ethernet Settings

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplex</td>
<td>Auto (full-duplex)</td>
</tr>
<tr>
<td>Encapsulation</td>
<td>ARPA</td>
</tr>
<tr>
<td>MTU1</td>
<td>1500 bytes</td>
</tr>
<tr>
<td>Port Mode</td>
<td>Access</td>
</tr>
<tr>
<td>Speed</td>
<td>Auto (10000)</td>
</tr>
</tbody>
</table>
## MIBs for Layer 2 Interfaces

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIB Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF-MIB</td>
<td>To locate and download MIBs, go to the following URL:</td>
</tr>
</tbody>
</table>

Limited support includes only the following MIB Objects:

- `ifMauType` (Read-only) `GET`
- `ifMauAutoNegSupported` (Read-only) `GET`
- `ifMauTypeListBits` (Read-only) `GET`
- `ifMauDefaultType` (Read-write) `GET-SET`
- `ifMauAutoNegAdminStatus` (Read-write) `GET-SET`
- `ifMauAutoNegCapabilityBits` (Read-only) `GET`
- `ifMauAutoNegAdvertisedBits` (Read-write) `GET-SET`
Configuring Layer 3 Interfaces

This chapter contains the following sections:

- Information About Layer 3 Interfaces, page 23
- Licensing Requirements for Layer 3 Interfaces, page 26
- Guidelines and Limitations for Layer 3 Interfaces, page 26
- Default Settings for Layer 3 Interfaces, page 27
- Configuring Layer 3 Interfaces, page 27
- Verifying the Layer 3 Interfaces Configuration, page 31
- Monitoring Layer 3 Interfaces, page 33
- Configuration Examples for Layer 3 Interfaces, page 34
- Related Documents for Layer 3 Interfaces, page 34
- MIBs for Layer 3 Interfaces, page 34
- Standards for Layer 3 Interfaces, page 35
- Feature History for Layer 3 Interfaces, page 35

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 Layer 2 (switchports) by default. You can change this default behavior using the `no switchport` command from interface configuration mode. To change multiple ports at one time, you can specify a range of interfaces and then apply the `no switchport` command.

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

You can assign a static MAC address to a Layer 3 interface. For information on configuring MAC addresses, see the Layer 2 Switching Configuration Guide for your device.

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 2: Subinterfaces for VLANs

![Figure 2: Subinterfaces for VLANs](image)

### 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 1 and VLAN 10 communicates at Layer 3 over VLAN interface 10.

**Figure 3: 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.

**Tunnel Interfaces**

Cisco NX-OS supports tunnel interfaces as IP tunnels. IP tunnels can encapsulate a same-layer or higher layer protocol and transport the result over IP through a tunnel that is created between two routers.

**Licensing Requirements for Layer 3 Interfaces**

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

**Guidelines and Limitations for 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.

## 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>switch(config)# interface ethernet slot/port</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td>switch(config-if)# no switchport</td>
<td>Configures the interface as a Layer 3 interface and deletes any configuration specific to Layer 2 on this interface. <strong>Note</strong> To convert a Layer 3 interface back into a Layer 2 interface, use the switchport command.</td>
</tr>
<tr>
<td>4</td>
<td>switch(config-if)# [ip</td>
<td>ipv6] ip-address/length</td>
</tr>
<tr>
<td>5</td>
<td>switch(config-if)# medium {broadcast</td>
<td>p2p}</td>
</tr>
<tr>
<td>6</td>
<td>switch(config-if)# show interfaces</td>
<td>Displays the Layer 3 interface statistics. <em>(Optional)</em></td>
</tr>
<tr>
<td>7</td>
<td>switch(config-if)# copy running-config startup-config</td>
<td>Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. <em>(Optional)</em></td>
</tr>
</tbody>
</table>
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

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

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# interface ethernet slot/port</td>
<td>Enters interface configuration mode. The range for the slot is from 1 to 255. The range for the port is from 1 to 128.</td>
</tr>
</tbody>
</table>
| Step 3 | switch(config-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 1000000.  
  - **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 | switch(config-if)# copy running-config startup-config | (Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

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

```bash
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# feature interface-vlan</td>
<td>Enables VLAN interface mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config)# interface vlan number</td>
<td>Creates a VLAN interface. The number range is from 1 to 4094.</td>
</tr>
</tbody>
</table>
Purpose
Command or Action

Step 4
switch(config-if)# [ip | ipv6] address ip-address/length
Configures an IP address for this interface.

Step 5
switch(config-if)# show interface vlan number
(Optional)
Displays the VLAN interface statistics. The number range is from 1 to 4094.

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

This example shows how to create a VLAN interface:

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

Configuring a Loopback Interface

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

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 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
switch(config-if)# show interface loopback instance | (Optional)
Displays the loopback interface statistics. The instance range is from 0 to 1023. |
| Step 5
switch(config-if)# copy running-config startup-config | (Optional)
Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# interface interface-typenumber</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config-if)# vrf member vrf-name</td>
<td>Adds this interface to a VRF.</td>
</tr>
<tr>
<td>Step 4</td>
<td>switch(config-if)# [ip</td>
<td>ipv6]ip-address/length</td>
</tr>
<tr>
<td>Step 5</td>
<td>switch(config-if)# show vrf [vrf-name] interface interface-type number</td>
<td>(Optional) Displays VRF information.</td>
</tr>
<tr>
<td>Step 6</td>
<td>switch(config-if)# show interfaces</td>
<td>(Optional) Displays the Layer 3 interface statistics.</td>
</tr>
<tr>
<td>Step 7</td>
<td>switch(config-if)# copy running-config startup-config</td>
<td>(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

Verifying the Layer 3 Interfaces Configuration

Use one of the following commands to verify the configuration:

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
<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show interface ethernet slot/port</td>
<td>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).</td>
</tr>
<tr>
<td>show interface ethernet slot/port brief</td>
<td>Displays the Layer 3 interface operational status.</td>
</tr>
<tr>
<td>show interface ethernet slot/port capabilities</td>
<td>Displays the Layer 3 interface capabilities, including port type, speed, and duplex.</td>
</tr>
<tr>
<td>show interface ethernet slot/port description</td>
<td>Displays the Layer 3 interface description.</td>
</tr>
<tr>
<td>show interface ethernet slot/port status</td>
<td>Displays the Layer 3 interface administrative status, port mode, speed, and duplex.</td>
</tr>
<tr>
<td>show interface ethernet slot/port number</td>
<td>Displays the subinterface configuration, status, and counters (including the f-minute exponentially decayed moving average of inbound and outbound packet and byte rates).</td>
</tr>
<tr>
<td>show interface port-channel channel-id.number</td>
<td>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).</td>
</tr>
<tr>
<td>show interface loopback number</td>
<td>Displays the loopback interface configuration, status, and counters.</td>
</tr>
<tr>
<td>show interface loopback number brief</td>
<td>Displays the loopback interface operational status.</td>
</tr>
<tr>
<td>show interface loopback number description</td>
<td>Displays the loopback interface description.</td>
</tr>
<tr>
<td>show interface loopback number status</td>
<td>Displays the loopback interface administrative status and protocol status.</td>
</tr>
<tr>
<td>show interface vlan number</td>
<td>Displays the VLAN interface configuration, status, and counters.</td>
</tr>
<tr>
<td>show interface vlan number brief</td>
<td>Displays the VLAN interface operational status.</td>
</tr>
<tr>
<td>show interface vlan number description</td>
<td>Displays the VLAN interface description.</td>
</tr>
<tr>
<td>show interface vlan number private-vlan mapping</td>
<td>Displays the VLAN interface private VLAN information.</td>
</tr>
<tr>
<td>show interface vlan number status</td>
<td>Displays the VLAN interface administrative status and protocol status.</td>
</tr>
</tbody>
</table>
Monitoring Layer 3 Interfaces

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show interface ethernet slot/port counters</code></td>
<td>Displays the Layer 3 interface statistics (unicast, multicast, and broadcast).</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port counters brief</code></td>
<td>Displays the Layer 3 interface input and output counters.</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port counters detailed [all]</code></td>
<td>Displays the Layer 3 interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port counters error</code></td>
<td>Displays the Layer 3 interface input and output errors.</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port.counters snmp</code></td>
<td>Displays the Layer 3 interface counters reported by SNMP MIBs. You cannot clear these counters.</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port.number counters</code></td>
<td>Displays the subinterface statistics (unicast, multicast, and broadcast).</td>
</tr>
<tr>
<td><code>show interface port-channel channel-id.number counters</code></td>
<td>Displays the port-channel subinterface statistics (unicast, multicast, and broadcast).</td>
</tr>
<tr>
<td><code>show interface loopback number counters</code></td>
<td>Displays the loopback interface input and output counters (unicast, multicast, and broadcast).</td>
</tr>
<tr>
<td><code>show interface loopback number counters detailed [all]</code></td>
<td>Displays the loopback interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).</td>
</tr>
<tr>
<td><code>show interface loopback number counters errors</code></td>
<td>Displays the loopback interface input and output errors.</td>
</tr>
<tr>
<td><code>show interface vlan number counters</code></td>
<td>Displays the VLAN interface input and output counters (unicast, multicast, and broadcast).</td>
</tr>
<tr>
<td><code>show interface vlan number counters detailed [all]</code></td>
<td>Displays the VLAN interface statistics. You can optionally include all Layer 3 packet and byte counters (unicast and multicast).</td>
</tr>
<tr>
<td><code>show interface vlan counters snmp</code></td>
<td>Displays the VLAN interface counters reported by SNMP MIBs. You cannot clear these counters.</td>
</tr>
</tbody>
</table>
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)# no switchport
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

<table>
<thead>
<tr>
<th>Related Topics</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command syntax</td>
<td>Cisco Nexus 3000 Series Command Reference</td>
</tr>
<tr>
<td>IP</td>
<td>&quot;Configuring IP” chapter in the Cisco Nexus 3000 Series NX-OS Unicast Routing Configuration Guide</td>
</tr>
<tr>
<td>VLAN</td>
<td>&quot;Configuring VLANs” chapter in the Cisco Nexus 3000 Series NX-OS Layer 2 Switching Configuration Guide</td>
</tr>
</tbody>
</table>

MIBs for Layer 3 Interfaces

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIB Link</th>
</tr>
</thead>
</table>
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.

Feature History for Layer 3 Interfaces

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>show interface vlan vlan-id counters command</td>
<td>5.0(3)U3(1)</td>
<td>The show interface vlan vlan-id counters command has been enhanced to correctly show input and output packet counts.</td>
</tr>
</tbody>
</table>
Configuring Port Channels

This chapter contains the following sections:

- Information About Port Channels, page 37
- Configuring Port Channels, page 45
- Verifying Port Channel Configuration, page 52
- Verifying the Load-Balancing Outgoing Port ID, page 53
- Feature History for Port Channels, page 54

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 an 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 41

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 40.

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 an 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.

Note
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
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.

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

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Layer 2 Criteria</th>
<th>Layer 3 Criteria</th>
<th>Layer 4 Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination MAC</td>
<td>Destination MAC</td>
<td>Destination MAC</td>
<td>Destination MAC</td>
</tr>
<tr>
<td>Source MAC</td>
<td>Source MAC</td>
<td>Source MAC</td>
<td>Source MAC</td>
</tr>
</tbody>
</table>

- Shutdown
- SNMP traps
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Layer 2 Criteria</th>
<th>Layer 3 Criteria</th>
<th>Layer 4 Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source and destination MAC</td>
<td>Source and destination MAC</td>
<td>Source and destination MAC</td>
<td>Source and destination MAC</td>
</tr>
<tr>
<td>Destination IP</td>
<td>Destination MAC</td>
<td>Destination MAC, destination IP</td>
<td>Destination MAC, destination IP</td>
</tr>
<tr>
<td>Source IP</td>
<td>Source MAC</td>
<td>Source MAC, source IP</td>
<td>Source MAC, source IP</td>
</tr>
<tr>
<td>Source and destination IP</td>
<td>Source and destination MAC</td>
<td>Source and destination MAC, source and destination IP</td>
<td>Source and destination MAC, source and destination IP</td>
</tr>
<tr>
<td>Destination TCP/UDP port</td>
<td>Destination MAC</td>
<td>Destination MAC, destination IP</td>
<td>Destination MAC, destination IP, destination port</td>
</tr>
<tr>
<td>Source TCP/UDP port</td>
<td>Source MAC</td>
<td>Source MAC, source IP</td>
<td>Source MAC, source IP, source port</td>
</tr>
<tr>
<td>Source and destination TCP/UDP port</td>
<td>Source and destination MAC</td>
<td>Source and destination MAC, source and destination IP</td>
<td>Source and destination MAC, source and destination IP, source and destination port</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>You must enable the LACP feature before you can configure and use LACP functions.</td>
</tr>
</tbody>
</table>
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 4: 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>
<thead>
<tr>
<th>Channel Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>passive</td>
<td>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.</td>
</tr>
<tr>
<td>active</td>
<td>LACP mode that places a port into an active negotiating state, in which the port initiates negotiations with other ports by sending LACP packets.</td>
</tr>
<tr>
<td>on</td>
<td>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.</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Configurations</th>
<th>Port Channels with LACP Enabled</th>
<th>Static Port Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol applied</td>
<td>Enable globally.</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>Channel mode of links</td>
<td>Can be either:</td>
<td>Can only be On.</td>
</tr>
<tr>
<td></td>
<td>• Active</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Passive</td>
<td></td>
</tr>
</tbody>
</table>

**LACP Port Channel MinLinks**

A port channel aggregates similar ports to provide increased bandwidth in a single manageable interface. The MinLinks feature allows you to define the minimum number of interfaces from a LACP bundle that must fail before the port channel goes down.

The LACP port channel MinLinks feature does the following:

- Configures the minimum number of port channel interfaces that must be linked and bundled in the LACP port channel.
- Prevents a low-bandwidth LACP port channel from becoming active.
- Causes the LACP port channel to become inactive if only a few active members ports supply the required minimum bandwidth.
The MinLinks feature works only with LACP port channels. The device allows you to configure this feature in non-LACP port channels, but the feature is not operational.

**Configuring Port Channels**

**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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface port-channel channel-number</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config)# no interface port-channel channel-number</td>
</tr>
</tbody>
</table>

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.
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.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>switch(config)# interface type</td>
<td>Specifies the interface that you want to add to a channel group and enters the interface configuration mode.</td>
</tr>
<tr>
<td>slot/port</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# switchport mode trunk</td>
<td>(Optional) Configures the interface as a trunk port.</td>
</tr>
<tr>
<td>switch(config-if)# switchport trunk {allowed vlan vlan-id</td>
<td>native vlan vlan-id}</td>
</tr>
<tr>
<td>channel-group channel-number</td>
<td>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.</td>
</tr>
<tr>
<td>switch(config-if)# no channel-group</td>
<td>(Optional) Removes the port from the channel group. The port reverts to its original configuration.</td>
</tr>
</tbody>
</table>

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 Port Channels

#### Configuring Hardware Hashing for Multicast Traffic

By default, ingress multicast traffic on any port in the switch selects a particular port channel member to egress the traffic. You can configure hardware hashing for multicast traffic to reduce potential bandwidth issues and to provide effective load balancing of the ingress multicast traffic. Use the `hardware multicast hw-hash` command to enable hardware hashing. To restore the default, use the `no hardware multicast hw-hash` command.

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface port-channel channel-number</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# hardware multicast hw-hash</td>
</tr>
</tbody>
</table>

This example shows how to configure hardware hashing on a port channel:

```plaintext
switch# configure terminal
switch (config)# interface port-channel 21
switch(config-if)# hardware multicast hw-hash
```

This example shows how to remove hardware hashing from a port channel:

```plaintext
switch# configure terminal
switch (config)# interface port-channel 21
switch(config-if)# no hardware multicast hw-hash
```
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# feature lacp</td>
<td>Enables LACP on the switch.</td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config)# show feature</td>
<td>(Optional) Displays enabled features.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# interface type slot/port</td>
<td>Specifies the interface to configure, and enters the interface configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config-if)# channel-group channel-number [force] [mode {on</td>
<td>active</td>
</tr>
</tbody>
</table>
Purpose
Command or Action | Purpose
---|---
force—Specifies that the LAN port be forcefully added to the channel group. This option is available in Cisco NX-OS Release 5.0(2)N2(1).
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.

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 MinLinks

The MinLink feature works only with LACP port channels. The device allows you to configure this feature in non-LACP port channels, but the feature is not operational.

Important Cisco recommends that you only configure the MinLink feature on one end of your port channel. Configuring the `lacp min-links` command on both ends of the port channel might result in link flapping.
Configuring Port Channels

### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:** switch# configure terminal  
switch(config)# | |
| **Step 2** interface port-channel *number* | Specifies the interface to configure and enters interface configuration mode. |
| **Example:** switch(config) # interface port-channel  
3  
switch(config-if) # | |
| **Step 3** [no] lacp min-links *number* | Specifies the port channel interface to configure the number of minimum links and enters the interface configuration mode.  
The default value for *number* is 1. The range is from 1 to 16.  
Use the no form of this command to disable this feature. |
| **Example:** switch(config-if) # lacp min-links 3 | |
| **Step 4** show running-config interface port-channel *number* | (Optional) Displays the port channel MinLinks configuration. |
| **Example:** switch(config) # show running-config  
interface port-channel 3  
switch(config-if) # | |

This example shows how to configure the minimum number of port channel interfaces on module 3:

```plaintext
switch# configure terminal
switch(config) # interface port-channel 3
switch(config-if) # lacp min-links 3
switch(config-if) #
```
Configuring Port Channels

## 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# interface type slot/port</td>
<td>Specifies the interface to configure and enters the interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config-if)# lACP rate fast</td>
<td>Configures the fast rate (one second) at which LACP control packets are sent to an LACP-supported interface.</td>
</tr>
</tbody>
</table>

This example shows how to configure the LACP fast rate on Ethernet interface 1/4:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# lACP rate fast
```

This example shows how to restore the LACP default rate (30 seconds) on Ethernet interface 1/4:

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

### Configuring the LACP System Priority 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# lACP system-priority priority</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch# show lACP system-identifier</td>
<td>(Optional) Displays the LACP system identifier.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>switch(config)# interface type slot/port</td>
<td>Specifies the interface to configure, and enters the interface configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td>switch(config-if)# lacp port-priority priority</td>
<td>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.</td>
</tr>
</tbody>
</table>

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
```

## Verifying Port Channel Configuration

To display port channel configuration information, perform one of the following tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch# show interface port-channel channel-number</td>
<td>Displays the status of a port channel interface.</td>
</tr>
<tr>
<td>switch# show feature</td>
<td>Displays enabled features.</td>
</tr>
<tr>
<td>switch# show resource</td>
<td>Displays the number of resources currently available in the system.</td>
</tr>
<tr>
<td>switch# show lacp {counters</td>
<td>interface type slot/port</td>
</tr>
</tbody>
</table>
## Purpose

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch# show port-channel compatibility-parameters</td>
<td>Displays the parameters that must be the same among the member ports in order to join a port channel.</td>
</tr>
<tr>
<td>switch# show port-channel database [interface port-channel channel-number]</td>
<td>Displays the aggregation state for one or more port-channel interfaces.</td>
</tr>
<tr>
<td>switch# show port-channel summary</td>
<td>Displays a summary for the port channel interfaces.</td>
</tr>
<tr>
<td>switch# show port-channel traffic</td>
<td>Displays the traffic statistics for port channels.</td>
</tr>
<tr>
<td>switch# show port-channel usage</td>
<td>Displays the range of used and unused channel numbers.</td>
</tr>
<tr>
<td>switch# show port-channel database</td>
<td>Displays information on current running of the port channel feature.</td>
</tr>
<tr>
<td>switch# show port-channel load-balance</td>
<td>Displays information about load-balancing using port channels.</td>
</tr>
</tbody>
</table>

---

### 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, for example when there is a single port in a port-channel.

To display the load-balancing outgoing port ID, perform one of the tasks listed in the table below.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 14-src-port port-id 14-dst-port port-id ether-type ether-type ip-proto ip-proto</td>
<td>Displays the outgoing port ID.</td>
</tr>
</tbody>
</table>

#### Example

This example shows the output of the short `port-channel load-balance` command.

```
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 ether-type ether-type ip-proto ip-proto
```

Missing params will be substituted by 0's. Load-balance Algorithm on switch: source-dest-port
Example

This example shows the output of the short `port-channel load-balance` command.

```
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 ether-type 0x0800 ip-proto 0x11
    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
ether-type: 0x0800
proto-type: 0x11
```

---

### Feature History for Port Channels

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Links</td>
<td>5.0(3)U3(1)</td>
<td>Added information about setting up and using the Minimum Links feature.</td>
</tr>
</tbody>
</table>
CHAPTER 5

Configuring IP Tunnels

This chapter contains the following sections:

- Information About IP Tunnels, page 55
- Licensing Requirements for IP Tunnels, page 56
- Prerequisites for IP Tunnels, page 56
- Guidelines and Limitations for IP Tunnels, page 57
- Default Settings for IP Tunneling, page 57
- Configuring IP Tunnels, page 57
- Verifying the IP Tunnel Configuration, page 60
- Configuration Examples for IP Tunneling, page 61
- Related Documents for IP Tunnels, page 61
- Standards for IP Tunnels, page 61
- Feature History for Configuring IP Tunnels, page 62

Information About IP Tunnels

IP tunnels can encapsulate a same-layer or higher-layer protocol and transport the result over IP through a tunnel created between two devices.

IP tunnels consists of the following three main components:

- Passenger protocol—The protocol that needs to be encapsulated. IPv4 is an example of a passenger protocol.
- Carrier protocol—The protocol that is used to encapsulate the passenger protocol. Cisco NX-OS supports generic routing encapsulation (GRE) as a carrier protocol.
- Transport protocol—The protocol that is used to carry the encapsulated protocol. IPv4 is an example of a transport protocol.
An IP tunnel takes a passenger protocol, such as IPv4, and encapsulates that protocol within a carrier protocol, such as GRE. The device then transmits this carrier protocol over a transport protocol, such as IPv4.

You configure a tunnel interface with matching characteristics on each end of the tunnel.

You must enable the tunnel feature before you can configure it.

**GRE Tunnels**

You can use GRE as the carrier protocol for a variety of passenger protocols.

The figure shows the IP tunnel components for a GRE tunnel. The original passenger protocol packet becomes the GRE payload and the device adds a GRE header to the packet. The device then adds the transport protocol header to the packet and transmits it.

*Figure 5: GRE PDU*

### Licensing Requirements for IP Tunnels

<table>
<thead>
<tr>
<th>Product</th>
<th>License Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco NX-OS</td>
<td>IP tunnels require an Enterprise Services license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the <em>Cisco NX-OS Licensing Guide</em>.</td>
</tr>
</tbody>
</table>

### Prerequisites for IP Tunnels

IP tunnels have the following prerequisites:

- You must be familiar with TCP/IP fundamentals to configure IP tunnels.
- You are logged on to the switch.
- You have installed the Enterprise Services license for Cisco NX-OS.
- You must enable the tunneling feature in a device before you can configure and enable any IP tunnels.
Guidelines and Limitations for IP Tunnels

Guidelines and Limitations for IP Tunnels

IP tunnels have the following configuration guidelines and limitations:

- Cisco NX-OS software supports the GRE header defined in IETF RFC 2784. Cisco NX-OS software does not support tunnel keys and other options from IETF RFC 1701.
- The Cisco Nexus device supports a maximum of eight tunnels.
- The Cisco Nexus device does not support the following features:
  - Path maximum transmission unit (MTU) discovery
  - Statistics
  - Access control lists (ACLs)
  - Unicast reverse path forwarding (URPF)
  - Multicast traffic and associated multicast protocols such as Internet Group Management Protocol (IGMP) and Protocol Independent Multicast (PIM).
- Cisco NX-OS software does not support the Web Cache Control Protocol (WCCP) on tunnel interfaces.

Default Settings for IP Tunneling

Default Settings for IP Tunneling

The following table lists the default settings for IP tunnel parameters.

Table 6: Default IP Tunnel Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunnel feature</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

Configuring IP Tunnels

Enabling Tunneling

Before You Begin

You must enable the tunneling feature before you can configure any IP tunnels.
### Creating a Tunnel Interface

You can create a tunnel interface and then configure this logical interface for your IP tunnel.

#### Before You Begin

Both the tunnel source and the tunnel destination must exist within the same virtual routing and forwarding (VRF) instance.

Ensure that you have enabled the tunneling feature.

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# feature tunnel</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config)# exit</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config)# show feature tunnel</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>switch# copy running-config startup-config</td>
</tr>
</tbody>
</table>

This example shows how to enable the tunnel feature:

```plaintext
switch# configure terminal
switch(config)# feature tunnel
switch(config)# exit
switch(config)# copy running-config startup-config
```
### Configuring IP Tunnels

#### Creating a GRE Tunnel

You can set a tunnel interface to GRE tunnel mode.

**Before You Begin**

Ensure that you have enabled the tunneling feature.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Switch(config)# interface tunnel number</td>
<td>Enters a tunnel interface configuration mode.</td>
</tr>
<tr>
<td>Switch(config)# tunnel mode gre ip</td>
<td>Sets this tunnel mode to GRE.</td>
</tr>
<tr>
<td>Switch(config)# show interface tunnel number</td>
<td>(Optional) Displays the tunnel interface statistics.</td>
</tr>
<tr>
<td>Switch(config-if)# copy running-config startup-config</td>
<td>(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

This example shows how to create the tunnel interface to GRE and set the GRE tunnel keepalives:

```
switch# configure terminal
switch(config)# interface tunnel 1
```

---

This example shows how to create the tunnel interface:

```
switch# configure terminal
switch(config)# interface tunnel 1
switch(config)# tunnel source ethernet 1/2
switch(config)# tunnel destination 192.0.2.1
switch(config)# copy running-config startup-config
```
Assigning VRF Membership to a Tunnel Interface

You can add a tunnel interface to a VRF.

**Before You Begin**

Ensure that you have enabled the tunneling feature.

Assign the IP address for a tunnel interface after you have configured the interface for a VRF.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: <code>switch# configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2: <code>switch(config)# interface tunnel number</code></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Step 3: <code>switch(config)# vrf member vrf-name</code></td>
<td>Adds this interface to a VRF.</td>
</tr>
<tr>
<td>Step 4: <code>switch(config)# ip address ip-prefix/length</code></td>
<td>Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.</td>
</tr>
<tr>
<td>Step 5: <code>switch(config)# show vrf [vrf-name] interface interface-type number</code></td>
<td>(Optional) Displays VRF information.</td>
</tr>
<tr>
<td>Step 6: <code>switch(config-if)# copy running-config startup-config</code></td>
<td>(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

This example shows how to add a tunnel interface to the VRF:

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

**Verifying the IP Tunnel Configuration**

To verify IP tunnel configuration information, perform one of the following tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show interface tunnel number</code></td>
<td>Displays the configuration for the tunnel interface (MTU, protocol, transport, and VRF). Displays input and output packets, bytes, and packet rates.</td>
</tr>
</tbody>
</table>
### Command | Purpose
--- | ---
**show interface tunnel number brief** | Displays the operational status, IP address, encapsulation type, and MTU of the tunnel interface.

**show interface tunnel number description** | Displays the configured description of the tunnel interface.

**show interface tunnel number status** | Displays the operational status of the tunnel interface.

**show interface tunnel number status err-disabled** | Displays the error disabled status of the tunnel interface.

---

### Configuration Examples for IP Tunneling

This example shows a simple GRE tunnel. Ethernet 1/2 is the tunnel source for router A and the tunnel destination for router B. Ethernet interface 1/3 is the tunnel source for router B and the tunnel destination for router A.

**router A:**
```plaintext
feature tunnel
interface tunnel 0
  ip address 209.165.20.2/8
  tunnel source ethernet 1/2
  tunnel destination 192.0.2.2
  tunnel mode gre ip
interface ethernet1/2
  ip address 192.0.2.55/8
```

**router B:**
```plaintext
feature tunnel
interface tunnel 0
  ip address 209.165.20.1/8
  tunnel source ethernet 1/3
  tunnel destination 192.0.2.55
  tunnel mode gre ip
interface ethernet 1/3
  ip address 192.0.2.2/8
```

---

### Related Documents for IP Tunnels

<table>
<thead>
<tr>
<th>Related Topics</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Tunnel commands</td>
<td><em>Cisco Nexus 3000 Series Interfaces Command Reference</em></td>
</tr>
</tbody>
</table>

---

### Standards for IP Tunnels

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.
# Feature History for Configuring IP Tunnels

## Table 7: Feature History for Configuring IP Tunnels

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP tunnels</td>
<td>5.0(3)U4(1)</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>
Configuring Virtual Port Channels

This chapter contains the following sections:

- Information About vPCs, page 63
- Guidelines and Limitations for vPCs, page 72
- Configuring vPCs, page 73
- Verifying the vPC Configuration, page 85
- vPC Default Settings, page 90

Information About vPCs

vPC Overview

A virtual port channel (vPC) allows links that are physically connected to two different Cisco Nexus 3000 Series switches 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. 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 3000 Series switch 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**—The 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**—The link used to synchronize states between the vPC peer devices.
- **vPC member port**—Interfaces that belong to the vPCs.
- **vPC domain**—This domain includes both vPC peer devices, the vPC peer-keepalive link, and all of the port channels in the vPC connected to the downstream devices. It is also associated to the configuration mode that you must use to assign vPC global parameters. 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 3000 Series 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 3000 Series Switch vPC Topology

You can connect a pair of Cisco Nexus 3000 Series switches in a vPC directly to another switch or to a server. Up to 8 interfaces could be connected to each Cisco Nexus 3000 Series switch providing 16 interfaces bundled for the vPC pair. The topology that is shown in the following figure provides the vPC functionality to dual connected switches or servers with 10-Gigabit or 1-Gigabit Ethernet uplink interfaces.

![Figure 6: Switch-to-Switch vPC Topology](image)

The switch connected to the pair of Cisco Nexus 3000 Series switches can be any standards-based Ethernet switch. Common environments to use this configuration include Blade Chassis with dual switches connected to the pair of Cisco Nexus 3000 Series switches through vPC or Unified Computing Systems connected to the pair of Cisco Nexus 3000 Series switches.

### 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.
Cisco recommends 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.

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.

If one of the vPC peer switches fails, the vPC peer switch on the other side of the vPC peer link senses the failure when it does not receive any peer-keepalive messages. The default interval time for the vPC peer-keepalive message is 1 second. You can configure the interval between 400 milliseconds and 10 seconds. You can also configure a timeout value with a range of 3 to 20 seconds; the default timeout value is 5 seconds. The peer-keepalive status is checked only when the peer-link goes down.

The vPC peer-keepalive can be carried either in the management or default VRF on the Cisco Nexus 3000 Series switch. When you configure the switches to use the management VRF, the source and destination for the keepalive messages are the mgmt 0 interface IP addresses. When you configure the switches to use the default VRF, an SVI must be created to act as the source and destination addresses for the vPC peer-keepalive messages. Ensure that both the source and destination IP addresses used for the peer-keepalive messages are unique in your network and these IP addresses are reachable from the VRF associated with the vPC peer-keepalive link.

We recommend that you configure the vPC peer-keepalive link on the Cisco Nexus 3000 Series switch 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.

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 check 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.
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 may 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**

When a consistency check fails, vPCs are brought down only on the secondary vPC switch. The VLANs remain up on the primary switch and Type-1 configurations can be performed without traffic disruption. This feature is used both in the case of global as well as interface-specific Type-1 inconsistencies.
Per-VLAN Consistency Check

Some 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

The vPC auto-recovery feature re-enables vPC links in the following scenarios:

- 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 240-3600 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 re-enables the vPCs. In this scenario, the vPC waits for three consecutive keep-alive failures to recover the vPC links.

The vPC auto-recovery feature is disabled by default.

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.
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 connecting to the downstream switch from the vPC peer switch must be identical on both vPC peer switches.
vPC Interactions with Other Features

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 manually configuring 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 will 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.
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.

**Guidelines and Limitations for vPCs**

vPC has the following configuration guidelines and limitations:

- vPC is not qualified with IPv6.
- 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.
- Cisco recommends 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.
- Only Port Channels can be in vPCs. A vPC can be configured on a normal Port Channel (switch-to-switch vPC topology) and on an Port Channel host interface (host interface vPC topology).
- 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 might 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.
- You might experience traffic disruption when the first member of a vPC is brought up.
Configuring vPCs

Enabling vPCs

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

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# feature vpc</td>
<td>Enables vPCs on the switch.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch# show feature</td>
<td>(Optional) Displays which features are enabled on the switch.</td>
</tr>
<tr>
<td><strong>Step 4</strong> switch# copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

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 3000 Series switch clears all the vPC configurations.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# no feature vpc</td>
<td>Disables vPCs on the switch.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch# show feature</td>
<td>(Optional) Displays which features are enabled on the switch.</td>
</tr>
</tbody>
</table>
### 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 with the following procedure.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>switch(config)# vpc domain domain-id</td>
<td>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. <strong>Note</strong> You can also use the vpc domain command to enter the vpc-domain configuration mode for an existing vPC domain.</td>
</tr>
<tr>
<td>3</td>
<td>switch# show vpc brief</td>
<td>(Optional) Displays brief information about each vPC domain.</td>
</tr>
<tr>
<td>4</td>
<td>switch# copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

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) 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 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 with the following procedure.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# vpc domain domain-id</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-vpc-domain)# peer-keepalive destination ipaddress [hold-timeout secs</td>
</tr>
<tr>
<td></td>
<td>interval msecs</td>
</tr>
<tr>
<td></td>
<td>precedence {prec-value</td>
</tr>
<tr>
<td></td>
<td>network</td>
</tr>
<tr>
<td></td>
<td>internet</td>
</tr>
<tr>
<td></td>
<td>critical</td>
</tr>
<tr>
<td></td>
<td>flash-override</td>
</tr>
<tr>
<td></td>
<td>flash</td>
</tr>
<tr>
<td></td>
<td>immediate priority</td>
</tr>
<tr>
<td></td>
<td>routine}</td>
</tr>
<tr>
<td></td>
<td>tos {tos-value</td>
</tr>
<tr>
<td></td>
<td>max-reliability</td>
</tr>
<tr>
<td></td>
<td>max-throughput</td>
</tr>
<tr>
<td></td>
<td>min-delay</td>
</tr>
<tr>
<td></td>
<td>min-monetary-cost</td>
</tr>
<tr>
<td></td>
<td>normal}</td>
</tr>
<tr>
<td></td>
<td>tos-byte</td>
</tr>
<tr>
<td></td>
<td>tos-byte-value}</td>
</tr>
<tr>
<td></td>
<td>source ipaddress</td>
</tr>
<tr>
<td></td>
<td>vrf {name</td>
</tr>
<tr>
<td></td>
<td>management vpc-keepalive}</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-vpc-domain)# vpc peer-keepalive destination ipaddress source ipaddress</td>
</tr>
</tbody>
</table>
### Configuring a vPC Keepalive Link and Messages

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 5</td>
<td>switch# show vpc peer-keepalive</td>
<td>(Optional) Displays information about the configuration for the keepalive messages.</td>
</tr>
<tr>
<td>Step 6</td>
<td>switch# copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

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.

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# sh 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,
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 with the following procedures

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# interface port-channel channel-number</td>
<td>Selects the EtherChannel that you want to use as the vPC peer link for this switch, and enters the interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config-if)# vpc peer-link</td>
<td>Configures the selected EtherChannel as the vPC peer link, and enters the vpc-domain configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong> switch# show vpc brief</td>
<td>(Optional) Displays information about each vPC, including information about the vPC peer link.</td>
</tr>
<tr>
<td><strong>Step 5</strong> switch# copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

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.

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch# show vpc</td>
<td>Displays the status of those parameters that must be</td>
</tr>
<tr>
<td>consistency-parameters</td>
<td>consistent across all vPC interfaces.</td>
</tr>
<tr>
<td>interface port-channel</td>
<td></td>
</tr>
<tr>
<td>channel-number</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Local Value</th>
<th>Peer Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>QoS</td>
<td>2</td>
<td>(1538, 0, 0,</td>
<td>(1538, 0, 0,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0, 0, 0, 0)</td>
<td>0, 0, 0, 0)</td>
</tr>
<tr>
<td>STP Mode</td>
<td>1</td>
<td>Rapid-PVST</td>
<td>Rapid-PVST</td>
</tr>
<tr>
<td>STP Disabled</td>
<td>1</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>STP MST Region Name</td>
<td>1</td>
<td>&quot;&quot;</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>STP MST Region Revision</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VLAN Mapping</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STP Loopguard</td>
<td>1</td>
<td>Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>STP Bridge Assurance</td>
<td>1</td>
<td>Enabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>STP Port Type, Edge BPDUGuard</td>
<td>1</td>
<td>Normal, Disabled,</td>
<td>Normal, Disabled,</td>
</tr>
<tr>
<td>STP MST Simulate PVST</td>
<td>1</td>
<td>Enabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Allowed VLANS</td>
<td>-</td>
<td>1,624</td>
<td>1</td>
</tr>
<tr>
<td>Local suspended VLANS</td>
<td>-</td>
<td>624</td>
<td>-</td>
</tr>
</tbody>
</table>

switch#
Enabling vPC Auto-Recovery

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# vpc domain domain-id</td>
<td>Enters vpc-domain configuration mode for an existing vPC domain.</td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config-vpc-domain)# auto-recovery reload-delay delay</td>
<td>Enables the auto-recovery feature and sets the reload delay period. The default is disabled.</td>
</tr>
</tbody>
</table>

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

version 5.0(3)U2(1)
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 when the routing tables may not be converged 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.
### Excluding VLAN Interfaces From Shutdown When vPC Peer Link Fails

When a vPC peer-link is lost, the vPC secondary switch suspends its vPC member ports and its 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 <code>switch# configure terminal</code></td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Step 2 <code>switch(config)# vpc domain domain-id</code></td>
<td>Creates a vPC domain on the switch if it does not already exist, and enters the vpc-domain configuration mode.</td>
</tr>
<tr>
<td>Step 3 <code>switch(config-vpc-domain)# delay restore time</code></td>
<td>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.</td>
</tr>
</tbody>
</table>
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
```

**Configuring the VRF Name**

The switch services, such as ping, ssh, telnet, radius, are VRF aware. The VRF name must be configured in order for the correct routing table to be used.

You can specify the VRF name.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# ping ipaddress vrf vrf-name</td>
</tr>
</tbody>
</table>

This example shows how to specify 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 may not receive multicast traffic. The non-vPC VLANs are the VLANs that are not trunked over a peer-link.

**Before You Begin**

Use the `show interfaces brief` command to view the interfaces that are in use on a switch. To bind the VRF to the vPC, you must use a VLAN that is not already in use.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
</tbody>
</table>

Enters configuration mode.
### Configuring Virtual Port Channels

#### 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Selects the port channel that you want to put into the vPC to connect to the downstream switch, and enters the interface configuration mode.</td>
</tr>
<tr>
<td>switch(config)# interface port-channel channel-number</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>A vPC can be configured on a normal port channel (physical vPC topology) and on an port channel host interface (host interface vPC topology)</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures the selected port channel into the vPC to connect to the downstream switch. The range is from 1 to 4096.</td>
</tr>
<tr>
<td>switch(config-if)# vpc number</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Displays information about each vPC.</td>
</tr>
<tr>
<td>switch# show vpc brief</td>
<td>(Optional)</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>switch# copy running-config startup-config</td>
<td>(Optional)</td>
</tr>
</tbody>
</table>

This example shows how to configure a port channel that will connect to the downstream device:

```bash
switch# configure terminal
switch(config)# interface port-channel 20
switch(config-if)# vpc 5
```
Manually Configuring a vPC Domain MAC Address

Configuring the system-mac is an optional configuration step. This section explains how to configure it in case you want to.

**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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# vpc domain domain-id</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-vpc-domain)# system-mac mac-address</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch# show vpc role</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>switch# copy running-config startup-config</td>
</tr>
</tbody>
</table>

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 with the following procedure.
Procedure

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>switch(config)# vpc domain domain-id</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>switch(config-vpc-domain)# system-priority priority</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>switch# show vpc brief</td>
<td>(Optional) Displays information about each vPC, including information about the vPC peer link.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>switch# copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

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 with the following procedure.
Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# vpc domain domain-id</td>
<td>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.</td>
</tr>
<tr>
<td>Step 3</td>
<td>switch(config-vpc-domain)# role priority priority</td>
<td>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.</td>
</tr>
<tr>
<td>Step 4</td>
<td>switch# show vpc brief</td>
<td>(Optional) Displays information about each vPC, including information about the vPC peer link.</td>
</tr>
<tr>
<td>Step 5</td>
<td>switch# copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

This example shows how to configure a vPC peer link:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-if)# role priority 4000
```

Verifying the vPC Configuration

Use the following commands to display vPC configuration information:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch# show feature</td>
<td>Displays whether vPC is enabled or not.</td>
</tr>
<tr>
<td>switch# show port-channel capacity</td>
<td>Displays how many EtherChannels are configured and how many are still available on the switch.</td>
</tr>
<tr>
<td>switch# show running-config vpc</td>
<td>Displays running configuration information for vPCs.</td>
</tr>
<tr>
<td>switch# show vpc brief</td>
<td>Displays brief information on the vPCs.</td>
</tr>
<tr>
<td>switch# show vpc consistency-parameters</td>
<td>Displays the status of those parameters that must be consistent across all vPC interfaces.</td>
</tr>
<tr>
<td>switch# show vpc peer-keepalive</td>
<td>Displays information on the peer-keepalive messages.</td>
</tr>
<tr>
<td>Command</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>switch# show vpc role</td>
<td>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.</td>
</tr>
<tr>
<td>switch# show vpc statistics</td>
<td>Displays statistics on the vPCs. Note This command displays the vPC statistics only for the vPC peer device that you are working on.</td>
</tr>
</tbody>
</table>

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

```
<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Po1</td>
<td>up</td>
<td>1</td>
</tr>
</tbody>
</table>
```

### 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
Per-vlan consistency status : success
Configuration consistency status : failed
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
-------------------------------------------------------------
<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Po1</td>
<td>up</td>
<td>1-10</td>
</tr>
</tbody>
</table>

vPC status
-------------------------------------------------------------
<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Consistency Reason</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Po20</td>
<td>down*</td>
<td>failed</td>
<td>1-10</td>
</tr>
<tr>
<td>30</td>
<td>Po30</td>
<td>down*</td>
<td>failed</td>
<td>1-10</td>
</tr>
</tbody>
</table>

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
-------------------------------------------------------------
<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Po1</td>
<td>up</td>
<td>1-10</td>
</tr>
</tbody>
</table>

vPC status
-------------------------------------------------------------
<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Consistency Reason</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Po20</td>
<td>up</td>
<td>failed</td>
<td>1-10</td>
</tr>
<tr>
<td>30</td>
<td>Po30</td>
<td>up</td>
<td>failed</td>
<td>1-10</td>
</tr>
</tbody>
</table>

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
Viewing a Per-VLAN Consistency Status

To view the per-VLAN consistency or inconsistency status, enter the `show vpc consistency-parameters vlans` command.

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
```

---

### Viewing a Per-VLAN Consistency Status

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**

<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Po1</td>
<td>up</td>
<td>1</td>
</tr>
</tbody>
</table>

**vPC status**

<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Consistency</th>
<th>Reason</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Po20</td>
<td>up</td>
<td>success</td>
<td>success</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>Po30</td>
<td>down*</td>
<td>failed</td>
<td>Compatibility check failed</td>
<td>1</td>
</tr>
</tbody>
</table>

for port mode

---

Cisco Nexus 3000 Series NX-OS Interfaces Configuration Guide, Release 5.0(3)U5(1)

Viewing a Per-VLAN Consistency Status

To view the per-VLAN consistency or inconsistency status, enter the `show vpc consistency-parameters vlans` command.

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

<table>
<thead>
<tr>
<th>vPC Peer-link status</th>
</tr>
</thead>
<tbody>
<tr>
<td>id Port Status Active vlans</td>
</tr>
<tr>
<td>-- ---- ------ --------------------------------------------------</td>
</tr>
<tr>
<td>1 Po1 up 1-10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>vPC status</th>
</tr>
</thead>
<tbody>
<tr>
<td>id Port Status Consistency Reason Active vlans</td>
</tr>
<tr>
<td>------- ----------- ------ ----------- -------------------------- -----------</td>
</tr>
<tr>
<td>20 Po20 up success success 1-10</td>
</tr>
<tr>
<td>30 Po30 up success success 1-10</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>vPC Peer-link status</th>
</tr>
</thead>
<tbody>
<tr>
<td>id Port Status Active vlans</td>
</tr>
<tr>
<td>-- ---- ------ --------------------------------------------------</td>
</tr>
<tr>
<td>1 Po1 up 1-4,6-10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>vPC status</th>
</tr>
</thead>
<tbody>
<tr>
<td>id Port Status Consistency Reason Active vlans</td>
</tr>
<tr>
<td>------- ----------- ------ ----------- -------------------------- -----------</td>
</tr>
<tr>
<td>20 Po20 up success success 1-4,6-10</td>
</tr>
<tr>
<td>30 Po30 up success success 1-4,6-10</td>
</tr>
</tbody>
</table>

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
vPC Default Settings

The following table lists the default settings for vPC parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>vPC system priority</td>
<td>32667</td>
</tr>
<tr>
<td>vPC peer-keepalive message</td>
<td>Disabled</td>
</tr>
<tr>
<td>vPC peer-keepalive interval</td>
<td>1 second</td>
</tr>
<tr>
<td>vPC peer-keepalive timeout</td>
<td>5 seconds</td>
</tr>
<tr>
<td>vPC peer-keepalive UDP port</td>
<td>3200</td>
</tr>
</tbody>
</table>
INDEX

A
adding ports 45
  port channels 45

B
bandwidth 29
  configuring 29

C
changed information 1
  description 1
channel mode 48
  port channels 48
channel modes 43
  port channels 43
configuration 31
  Layer 3 interfaces 31
    verifying 31
configuration examples 34, 61
  ip tunneling 61
  Layer 3 interfaces 34
configuring 27, 28, 29, 30
  interface bandwidth 29
  loopback interfaces 30
  routed interfaces 27
  subinterfaces 28
  VLAN interfaces 29
configuring LACP 48

D
default settings 27, 57
  ip tunnels 57
  Layer 3 interfaces 27

E
Ethernet interfaces 6
  interface speed 6

F
feature history 35, 54, 62
  ip tunnels 62
  Layer 3 interfaces 35
    port channels 54

G
gre tunnel, configuring 59
  interfaces 59
  gre tunnels 56
    interfaces 56
  guidelines 57
    ip tunnels 57
  guidelines and limitations 26, 72
    Layer 3 interfaces 26
      vPCs 72

H
hardware hashing 47
  multicast traffic 47

I
interface information, displaying 18
  layer 2 18
  interface speed 6, 10
    configuring 10
    Ethernet interfaces 6
port channels (continued)
  channel mode 48
  compatibility requirements 38
  configuring bandwidth 29
  creating 45
  feature history 54
  hardware hashing 47
  LACP 41
  LACP-enabled vs static 44
  load balancing 40, 46
    port channels 40
  moving into a vPC 82
  STP 37
port mode 9
  interface 9
prerequisites 56
  ip tunnels 56

R

related documents 34
  Layer 3 interfaces 34
routed interfaces 23, 27, 29
  configuring 27
  configuring bandwidth 29

S

SFP+ transceiver 6
Small form-factor pluggable (plus) transceiver 6
standards 35, 61
  ip tunnels 61
  Layer 3 interfaces 35
STP 37
  port channel 37
subinterfaces 24, 28, 29
  configuring 28
  configuring bandwidth 29

svi autostate 6
  layer 2 6
svi autostate, disabling 12
  layer 2 12

T

tunnel interface 60
  vrf membership, assigning 60
tunnel interfaces 26
tunnel interfaces, creating 58
  interfaces 58

U

UDLD 4, 5
  aggressive mode 5
  defined 4
  nonaggressive mode 5
UDLD modeA 8
  configuring 8
Unidirectional Link Detection 4

V

verifying 31
  Layer 3 interface configuration 31
VLAN 25
  interfaces 25
VLAN interfaces 29
  configuring 29
vPCs 72, 82
  guidelines and limitations 72
  moving port channels into 82
VRF 31
  assigning an interface to 31
  vrf membership, assigning 60
    tunnel interface 60