Cisco Nexus 5000 Series NX-OS Interfaces Configuration Guide, Release 5.2(1)N1(1)

First Published: 2012-07-02

Last Modified: 2018-03-30

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: 78-26881-OL

CONTENTS

Preface

The preface contains the following sections:

Audience

This publication is for network administrators who configure and maintain Cisco Nexus devices and Cisco Nexus 2000 Series Fabric Extenders.

Document Conventions

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

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).

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

Examples use the following conventions:

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

This document uses the following conventions:

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

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

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to: nexus5k-docfeedback@cisco.com.

We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request

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

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. RSS feeds are a free service.

CHAPTER 1

New and Changed Information

This chapter contains the following sections:

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 guide or of the new features in this release.

Table 1: New Features

Feature	Description	Where Documented
IPv6	Added support for IPv6 addressing.	• Routed Interfaces, on page 47 • Configuring a Subinterface, on page 52 • Configuring the Bandwidth on an Interface, on page 52 • Configuring a VLAN Interface, on page 53 • Configuring a Loopback Interface, on page 55 • Assigning an Interface to a VRF, on page 56

CHAPTER 2

Configuring Layer 2 Interfaces

This chapter contains the following sections:

Information About Ethernet Interfaces

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

The Ethernet interfaces also support Fibre Channel over Ethernet (FCoE). FCoE allows the physical Ethernet link to carry both Ethernet and Fibre Channel traffic.

The Ethernet interfaces are enabled by default.

Interface Command

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

- Interface type—All physical Ethernet interfaces use the **ethernet** keyword.
- Slot number:

  - Slot 1 includes all the fixed ports.
  - Slot 2 includes the ports on the upper expansion module (if populated).
  - Slot 3 includes the ports on the lower expansion module (if populated).
  - Slot 4 includes the ports on the lower expansion module (if populated).

- Port number— Port number within the group.

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

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

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

Information About Unified Ports

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

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

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

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

Guidelines and Limitations for Unified Ports

Unidirectional Link Detection Parameter

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

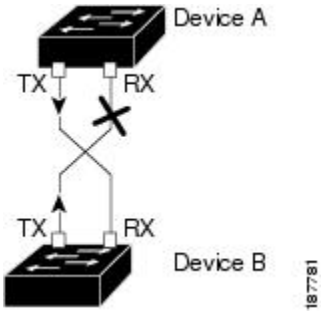
UDLD is a Layer 2 protocol that works with the Layer 1 protocols to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting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

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

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

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

Figure 1: Unidirectional Link



Default UDLD Configuration

The following table shows the default UDLD configuration.

Table 2: UDLD Default Configuration

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

UDLD Aggressive and Nonaggressive Modes

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

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

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

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

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

Interface Speed

Cisco Discovery Protocol

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

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

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

The switch supports both CDP Version 1 and Version 2.

Default CDP Configuration

The following table shows the default CDP configuration.

Table 3: Default CDP Configuration

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

Error-Disabled State

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

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

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

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

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

About Port Profiles

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

- Ethernet
- VLAN network interface
- Port channel

A command that is included in a port profile can be configured outside of the port profile. If the new configuration in the port profile conflicts with the configurations that exist outside the port profile, the commands configured for an interface in configuration terminal mode have higher priority than the commands

in the port profile. If changes are made to the interface configuration after a port profile is attached to it, and the configuration conflicts with that in the port profile, the configurations in the interface will be given priority.

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

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

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

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

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

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

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

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

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

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

Guidelines and Limitations for Port Profiles

Port profiles have the following configuration guidelines and limitations:

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

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

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

Debounce Timer Parameters

Debounce time is the amount of time that an interface waits to notify the supervisor of a link-state change, which in turn decreases traffic loss due to network reconfiguration, or helps bring up link faster or both.

You can configure the debounce timer separately for each Ethernet port and for link up and link down event separately and specify the delay time in milliseconds.

For a link going down, the interface waits to see if the link comes back up within the debounce time. For a link coming up, the interface waits for debounce link-up time before declaring it as UP.

The wait period is a time when the traffic is stopped. By default, the debounce timer is set for 100 milliseconds.

Caution When you enable the port debounce timer the link up and link down detections are delayed, resulting in a loss of traffic during the debounce period. This situation might affect the convergence and reconvergence of some protocols.

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 and a receive data field size of 2112 is displayed for Fibre Channel interfaces.

Information About Default Interfaces

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

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

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

Default Physical Ethernet Settings

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

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

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

Information About Access and Trunk Interfaces

Understanding Access and Trunk Interfaces

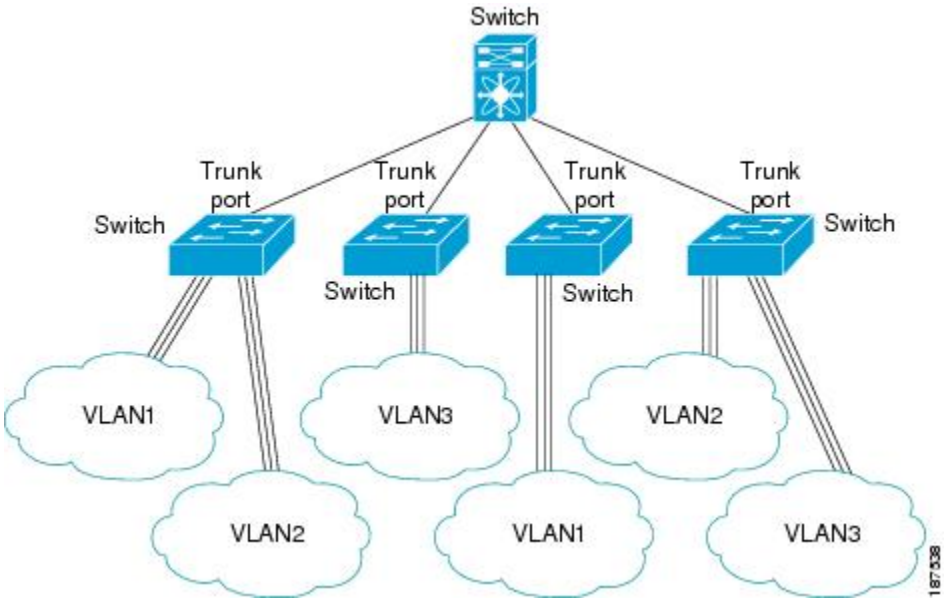
Ethernet interfaces can be configured either as access ports or a trunk ports, as follows:

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

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

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

Figure 2: Devices in a Trunking Environment



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

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

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

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

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

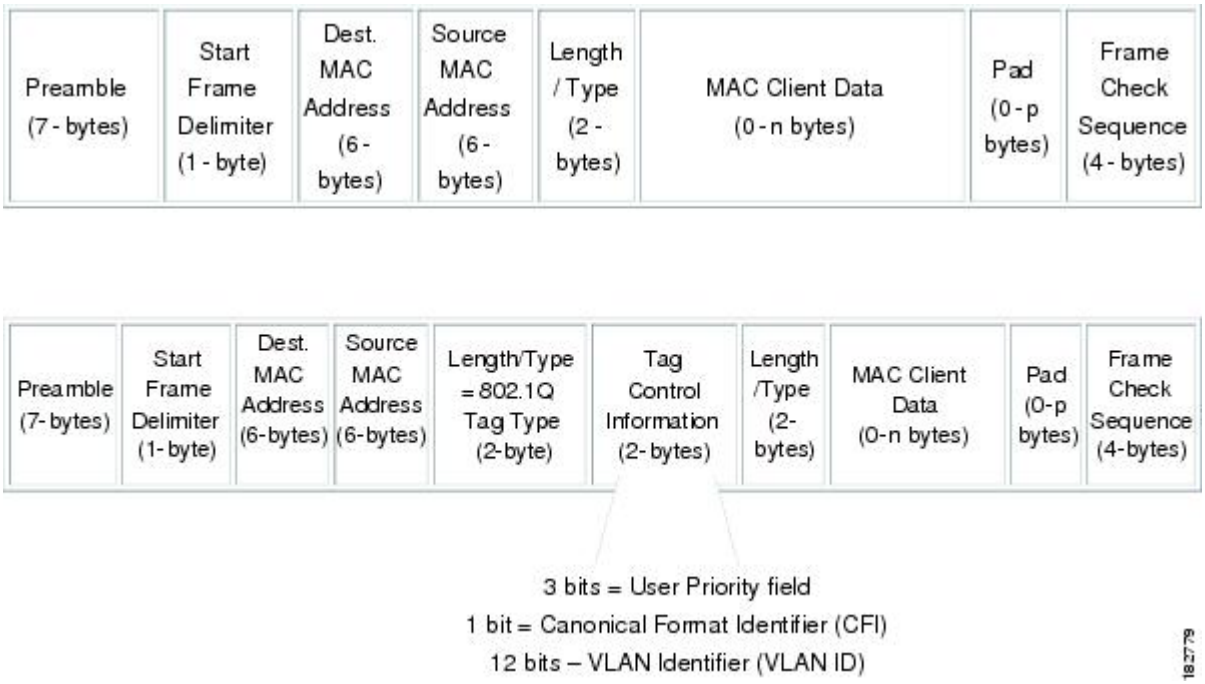
Understanding IEEE 802.1Q Encapsulation

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

To correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation (tagging) method. This tag carries information about the specific VLAN to which the frame

and packet belong. This method allows packets that are encapsulated for several different VLANs to traverse the same port and maintain traffic separation between the VLANs. The encapsulated VLAN tag also allows the trunk to move traffic end-to-end through the network on the same VLAN.

Figure 3: Header Without and With 802.1Q Tag Included



Understanding Access VLANs

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

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

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

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

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

Understanding the Native VLAN ID for Trunk Ports

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

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

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

Understanding Allowed VLANs

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

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

Understanding Native 802.1Q VLANs

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

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

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

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

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

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

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

Configuring Access and Trunk Interfaces

Configuring a LAN Interface as an Ethernet Access Port

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

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

Procedure

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

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

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

Configuring Access Host Ports

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

Before You Begin

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

Procedure

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

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

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

Configuring Trunk Ports

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

Note Cisco NX-OS supports only 802.1Q encapsulation.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {*type slot/port* | **port-channel** *number*}	Specifies an interface to configure, and enters interface configuration mode.

	Command or Action	Purpose
Step 3	switch(config-if)# **switchport mode** {**access** | **trunk**}	Sets the interface as an Ethernet trunk port. A trunk port can carry traffic in one or more VLANs on the same physical link (VLANs are based on the trunk-allowed VLANs list). By default, a trunk interface can carry traffic for all VLANs. To specify that only certain VLANs are allowed on the specified trunk, use the **switchport trunk allowed vlan** command.

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

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

Configuring the Native VLAN for 802.1Q Trunking Ports

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

Procedure

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

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

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

Configuring the Allowed VLANs for Trunking Ports

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

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

Procedure

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

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

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

Configuring Native 802.1Q VLANs

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

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

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

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

Procedure

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

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

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

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

Verifying the Interface Configuration

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

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

Configuring Ethernet Interfaces

The section includes the following topics:

Configuring a Layer 3 Interface on a Cisco Nexus Device

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

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

Procedure

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

This example shows how to configure a Layer 3 interface:

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

Configuring Unified Ports

Before You Begin

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

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

Procedure

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

	Command or Action	Purpose
		Note • Changing unified ports on an expansion module (GEM) requires that you power cycle the GEM card. You do not have to reboot the entire switch for changes to take effect. • When you configure unified ports as Fibre Channel, the existing configuration for Fibre Channel interfaces and VSAN memberships are unaffected.
Step 4	switch(config-slot) # **copy running-config startup-config**	Copies the running configuration to the startup configuration.
Step 5	switch(config-slot) # **reload**	Reboots the switch.
Step 6	switch(config) # **slot** *slot number*	Identifies the slot on the switch.
Step 7	switch(config-slot) # **no port** *port number* **type fc**	Removes the unified port.

Configuring the UDLD Mode

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

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature udld**	Enables UDLD for the device.
Step 3	switch(config)# **no feature udld**	Disables UDLD for the device.
Step 4	switch(config)# **show udld global**	Displays the UDLD status for the device.

	Command or Action	Purpose
Step 5	switch(config)# **interface** *type slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 6	switch(config-if)# **udld** {**enable** | **disable** | **aggressive**}	Enables the normal UDLD mode, disables UDLD, or enables the aggressive UDLD mode.
Step 7	switch(config-if)# **show udld** *interface*	Displays the UDLD status for the interface.

This example shows how to enable UDLD for the switch:

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

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

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

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

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

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

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

This example shows how to disable UDLD for the switch:

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

Configuring Interface Speed

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

Procedure

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

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

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

Disabling Link Negotiation

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

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

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

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

Procedure

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

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

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

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

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

Configuring the CDP Characteristics

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

Procedure

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

This example shows how to configure CDP characteristics:

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

Enabling or Disabling CDP

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

Procedure

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

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

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

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

Enabling the Error-Disabled Detection

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

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

Procedure

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

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

Enabling the Error-Disabled Recovery

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit*}	Specifies a condition under which the interface automatically recovers from the err-disabled state, and the device retries bringing the interface up. The device waits 300 seconds to retry. The default is disabled.

	Command or Action	Purpose
Step 3	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	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 enable err-disabled recovery under all conditions:

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

Configuring the Error-Disabled Recovery Interval

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **errdisable recovery interval** *interval*	Specifies the interval for the interface to recover from the err-disabled state. The range is from 30 to 65535 seconds. The default is 300 seconds.
Step 3	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	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 enable err-disabled recovery under all conditions:

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

Port Profiles

Creating a Port Profile

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

Procedure

	Command or Action	Purpose

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

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

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

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

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

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

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

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

```
switch(config-port-prof)# speed 10000
switch(config-port-prof)#
```

Modifying a Port Profile

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

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

Procedure

	Command or Action	Purpose

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

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

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

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

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

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

Enabling a Specific Port Profile

Procedure

	Command or Action	Purpose

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

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

Step 3	**state enabled** *name*
	
	Example:
	`switch(config-port-prof)# state enabled`
	`switch(config-port-prof)#`
		Enables the port profile.

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

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

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

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

Inheriting a Port Profile

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

Procedure

	Command or Action	Purpose

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

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

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

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

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

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

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

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

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

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

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

Removing an Inherited Port Profile

You can remove an inherited port profile.

Procedure

	Command or Action	Purpose

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

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

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

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

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

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

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

Assigning a Port Profile to a Range of Interfaces

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**interface** [**ethernet** *slot/port* | **interface-vlan** *vlan-id* | **port-channel** *number*]	Selects the range of interfaces.
Step 3	**inherit port-profile** *name*	Assigns the specified port profile to the selected interfaces.
Step 4	**exit**	Exits port profile configuration mode.
Step 5	**show port-profile** *name*	(Optional) Displays the port profile configuration.
Step 6	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 assign the port profile named adam to Ethernet interfaces 2/3 to 2/5, 3/2, and 1/20 to 1/25:

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

Removing a Port Profile from a Range of Interfaces

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**interface** [**ethernet** *slot/port* | **interface-vlan** *vlan-id* | **port-channel** *number*]	Selects the range of interfaces.

	Command or Action	Purpose
Step 3	**no inherit port-profile** *name*	Removes the specified port profile from the selected interfaces.
Step 4	**exit**	Exits port profile configuration mode.
Step 5	**show port-profile**	(Optional) Displays the port profile configuration.
Step 6	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 tos remove the port profile named adam from Ethernet interfaces 1/3-5:

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

Configuration Examples for Port Profiles

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

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

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

version 5.0(2)N1(1)

interface Ethernet1/14

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

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

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


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

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

version 5.0(2)N1(1)

interface Ethernet1/14
  inherit port-profile alpha

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

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

version 5.0(2)N1(1)

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

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

Configuring the Debounce Timer

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

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

Procedure

Step 1 switch# **configure terminal**
Enters global configuration mode.

Step 2 switch(config)# **interface** *type slot/port*
Enters interface configuration mode for the specified interface.

Step 3 switch(config-if)# **link debounce link-up time** *milliseconds*
Delays link-up declaration by configured time, in milliseconds. The range is 1 to 5000 milliseconds.

Step 4 switch(config-if)# **link debounce time** *milliseconds*
Delays link-down notification by configured time, in milliseconds. The range is 1 to 5000 milliseconds.

Disables the debounce timer if you specify 0 milliseconds.

Step 5 switch(config-if)# **no link debounce**
Sets debounce timer to the default value of 100 milliseconds.

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

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

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

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

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

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

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

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

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

Configuring the Description Parameter

You can provide textual interface descriptions for the Ethernet ports.

Procedure

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

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

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

Disabling and Restarting Ethernet Interfaces

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

Procedure

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

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

This example shows how to disable an Ethernet port:

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

This example shows how to restart an Ethernet interface:

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

Configuring Slow Drain Device Detection and Congestion Avoidance

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

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

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

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

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

Configuring a Stuck Frame Timeout Value

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

Procedure

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

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

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

Configuring a No-Credit Timeout Value

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

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

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

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

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

Procedure

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

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

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

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

Displaying Credit Loss Counters

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

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

Displaying Credit Loss Events

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

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

Displaying Timeout Drops

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

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

Displaying the Average Credit Not Available Status

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

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

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

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

Port Monitoring

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

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

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

Enabling Port Monitor

Procedure

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

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

Configuring a Port Monitor Policy

Procedure

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

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

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

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

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

Activating a Port Monitor Policy

Procedure

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

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

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

Displaying Port Monitor Policies

Use the following command to display port monitor policies:

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

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

FCoE Slow Drain Device Detection and Congestion Avoidance

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

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

Congestion timeout

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

Pause timeout

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

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

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

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

- ```
  switch# show event manager system-policy
  Name : __ethpm_slow_drain_core
  Description : 10 Pause Events in 1 minute. Action: None by default
  Overridable : Yes
  ```
- ```
  switch# show event manager system-policy
  Name : __ethpm_slow_drain_edge
     Description : 5 Pause Events in 1 minute. Action: None by default
     Overridable : Yes
  ```

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

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

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

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

Configuring a Pause Frame Timeout Value

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

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

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

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

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

Procedure

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

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

```
switch# configure terminal
switch(config)# system default interface pause timeout 358 mode core
switch(config)# system default interface pause mode edge
```

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

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

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

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

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

Displaying Interface Information

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

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

Command	Purpose
show port--profile	Displays information about the port profiles.

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

This example shows how to display the physical Ethernet interface:

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

This example shows how to display the physical Ethernet capabilities:

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

This example shows how to display the physical Ethernet transceiver:

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

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

```
switch# show interface brief

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

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

```
switch# show interface debounce

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

This example shows how to display the CDP neighbors:

Note The default device ID field for CDP advertisement is the hostname and serial number, as in the example above.

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

CHAPTER 3

Configuring Layer 3 Interfaces

This chapter contains the following sections:

Information About Layer 3 Interfaces

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

Routed Interfaces

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

All Ethernet ports are switched interfaces by default. You can change this default behavior with the CLI setup script or through the **system default switchport** command.

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

You can 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.

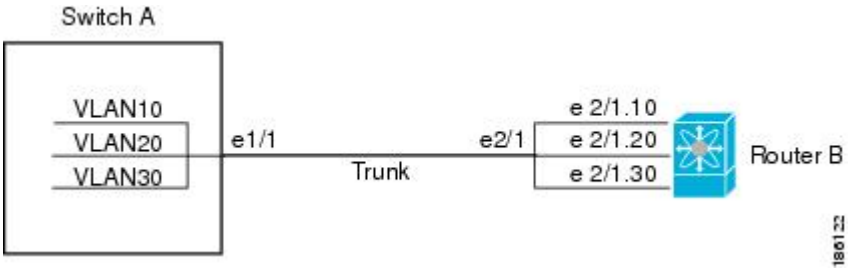
You create a subinterface with a name that consists of the parent interface name (for example, Ethernet 2/1) followed by a period and then by a number that is unique for that subinterface. For example, you could create a subinterface for Ethernet interface 2/1 named Ethernet 2/1.1 where .1 indicates 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 4: Subinterfaces for VLANs



VLAN Interfaces

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

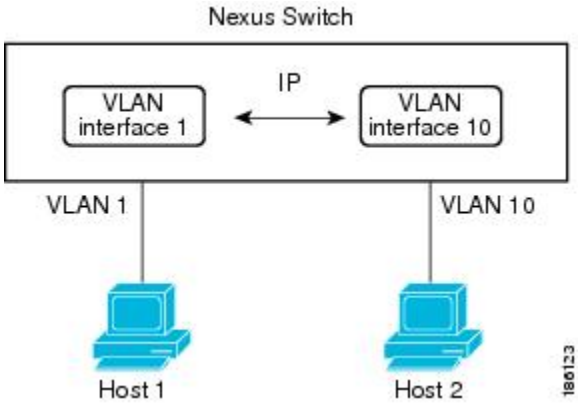
You must enable the VLAN network interface feature before you can configure it. The system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. For information about rollbacks and checkpoints, see the System Management Configuration Guide for your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

Note You cannot delete the VLAN interface for VLAN 1.

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

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

Figure 5: Connecting Two VLANs with VLAN Interfaces



Loopback Interfaces

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

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

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.

IP Addressing Scheme with Private VLANs

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

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

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

Licensing Requirements for Layer 3 Interfaces

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

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	switch(conifg-if)# **no switchport**	Configures the interface as a Layer 3 interface and deletes any configuration specific to Layer 2 on this interface. **Note** To convert a Layer 3 interface back into a Layer 2 interface, use the **switchport** command.
Step 4	switch(config-if)# [**ip**|**ipv6**]*ip-address/length*	Configures an IP address for this interface.
Step 5	switch(config-if)# **medium** {**broadcast** | **p2p**}	(Optional) Configures the interface medium as either point to point or broadcast. **Note** The default setting is broadcast, and this setting does not appear in any of the **show** commands. However, if you do change the setting to **p2p**, you will see this setting when you enter the **show running-config** command.
Step 6	switch(config-if)# **show interfaces**	(Optional) Displays the Layer 3 interface statistics.
Step 7	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 an IPv4-routed Layer 3 interface:

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

Configuring a Subinterface

Before You Begin

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

Procedure

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

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

Configuring the Bandwidth on an Interface

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

Procedure

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

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

Configuring a VLAN Interface

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature interface-vlan**	Enables VLAN interface mode.
Step 3	switch(config)# **interface vlan** *number*	Creates a VLAN interface. The *number* range is from 1 to 4094.
Step 4	switch(config-if)# [**ip** | **ipv6**] **address** *ip-address/length*	Configures an IP address for this interface.
Step 5	switch(config-if)# **no shutdown**	Brings the interface up administratively.

	Command or Action	Purpose
Step 6	switch(config-if)# **show interface vlan** *number*	(Optional) Displays the VLAN interface statistics. The *number* range is from 1 to 4094.
Step 7	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
```

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

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

Procedure

	Command or Action	Purpose
Step 1	Router(config)# **interface-vlan** *primary_vlan_ID*	Enters interface configuration mode for the primary VLAN. **Note** Isolated and community VLANs are both called secondary VLANs.
Step 2	Router(config-if)# **private-vlan mapping** {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*}	Maps the secondary VLANs to the Layer 3 VLAN interface of a primary VLAN to allow Layer 3 switching of private VLAN ingress traffic. When you map secondary VLANs to the Layer 3 VLAN interface of a primary VLAN, note the following information: • The **private-vlan mapping interface** configuration command only affects private VLAN ingress traffic that is Layer 3-switched. • The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single private VLAN ID or a hyphenated range of private VLAN IDs. • Enter a *secondary_vlan_list* parameter or use the **add** keyword with a *secondary_vlan_list* parameter to map the secondary VLANs to the primary VLAN.

	Command or Action	Purpose
		• Use the **remove** keyword with a *secondary_vlan_list* parameter to clear the mapping between secondary VLANs and the primary VLAN.
Step 3	Router(config-if)# **no private-vlan mapping**	Clears the mapping between the secondary VLANs and the primary VLAN.
Step 4	Router(config-if)# **end**	Exits configuration mode.
Step 5	Router **show interface private-vlan mapping**	Verifies the configuration.

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

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

Configuring a Loopback Interface

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface loopback** *instance*	Creates a loopback interface. The *instance* range is from 0 to 1023.
Step 3	switch(config-if)# [**ip** | **ipv6**] **address** *ip-address/length*	Configures an IP address for this interface.

	Command or Action	Purpose
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.

This example shows how to create a loopback interface:

```
switch# configure terminal
switch(config)# interface loopback 0
switch(config-if)# ip address 192.0.2.100/8
switch(config-if)# copy running-config startup-config
```

Assigning an Interface to a VRF

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-typenumber*	Enters interface configuration mode.
Step 3	switch(conifg-if)#**vrf member** *vrf-name*	Adds this interface to a VRF.
Step 4	switch(config-if)# [**ip** | **ipv6**]*ip-address/length*	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 5	switch(config-if)# **show vrf** [*vrf-name*] **interface** *interface-type number*	(Optional) Displays VRF information.
Step 6	switch(config-if)# **show interfaces**	(Optional) Displays the Layer 3 interface statistics.
Step 7	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 add a Layer 3 interface to the VRF:

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

Verifying the Layer 3 Interfaces Configuration

Use one of the following commands to verify the configuration:

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

Command	Purpose
show interface vlan *number* **brief**	Displays the VLAN interface operational status.
show interface vlan *number* **description**	Displays the VLAN interface description.
show interface vlan *number* **private-vlan mapping**	Displays the VLAN interface private VLAN information.
show interface vlan *number* **status**	Displays the VLAN interface administrative status and protocol status.

Monitoring Layer 3 Interfaces

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

Command	Purpose
show interface ethernet *slot*/*port* **counters**	Displays the Layer 3 interface statistics (unicast, multicast, and broadcast).
show interface ethernet *slot*/*port* **counters brief**	Displays the Layer 3 interface input and output counters.
show interface ethernet *slot*/*port* **counters detailed** [**all**]	Displays the Layer 3 interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).
show interface ethernet *slot*/*port* **counters error**	Displays the Layer 3 interface input and output errors.
show interface ethernet *slot*/*port* **counters snmp**	Displays the Layer 3 interface counters reported by SNMP MIBs. You cannot clear these counters.
show interface ethernet *slot*/*port*.*number* **counters**	Displays the subinterface statistics (unicast, multicast, and broadcast).
show interface port-channel *channel-id*.*number* **counters**	Displays the port-channel subinterface statistics (unicast, multicast, and broadcast).
show interface loopback *number* **counters**	Displays the loopback interface input and output counters (unicast, multicast, and broadcast).
show interface loopback *number* **counters detailed** [**all**]	Displays the loopback interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).

Command	Purpose
show interface loopback *number* **counters errors**	Displays the loopback interface input and output errors.
show interface vlan *number* **counters**	Displays the VLAN interface input and output counters (unicast, multicast, and broadcast).
show interface vlan *number* **counters detailed** [*all*]	Displays the VLAN interface statistics. You can optionally include all Layer 3 packet and byte counters (unicast and multicast).
show interface vlan *counters* **snmp**	Displays the VLAN interface counters reported by SNMP MIBs. You cannot clear these counters.

Configuration Examples for Layer 3 Interfaces

This example shows how to configure Ethernet subinterfaces:

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

This example shows how to configure a VLAN interface:

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

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

This example shows how to configure a loopback interface:

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

Related Documents for Layer 3 Interfaces

Related Topics	Document Title
Command syntax	For details about command syntax, see the command reference for your device.
IP	“Configuring IP” chapter in the Unicast Routing Configuration Guide for your device.

Related Topics	Document Title
VLAN	"Configuring VLANs" chapter in the Layer 2 Switching Configuration Guide for your device.

MIBs for Layer 3 Interfaces

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

Standards for Layer 3 Interfaces

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

CHAPTER 4

Configuring Port Channels

This chapter contains the following sections:

Information About Port Channels

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

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

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

You can use static port channels, with no associated protocol, for a simplified configuration. For more efficient use of the port channel, you can use the Link Aggregation Control Protocol (LACP), which is defined in IEEE 802.3ad. When you use LACP, the link passes protocol packets.

Related Topics

LACP Overview, on page 67

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 64.

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

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

Each port can be in only one port channel. All the ports in a port channel must be compatible; they must use the same speed and operate in full-duplex mode. When you are running static port channels without LACP, the individual links are all in the on channel mode; you cannot change this mode without enabling LACP.

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

You can create a port channel directly by creating the port-channel interface, or you can create a channel group that acts to aggregate individual ports into a bundle. When you associate an interface with a channel group, Cisco NX-OS creates a matching port channel automatically if the port channel does not already exist. You can also create the port channel first. In this instance, Cisco NX-OS creates an empty channel group with the same channel number as the port channel and takes the default configuration.

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

Guidelines and Limitations for Port Channel Configuration

Port channels can be configured in one of two ways: either in global configuration mode or in switch profile mode. Consider the following guidelines and limitations when configuring port channels via the configuration synchronization feature in Cisco NX-OS:

- Once a port channel is configured using switch profile mode, it cannot be configured using global configuration (config terminal) mode.

  Note Several port channel sub-commands are not configurable in switch profile mode. These commands can be configured from global configuration mode even if the port channel is created and configured in switch profile mode.

  For example, the following command can only be configured in global configuration mode:

  switchport private-vlan association trunk *primary-vlan secondary-vlan*

- Shutdown and no shutdown can be configured in either global configuration mode or switch profile mode.

- If a port channel is created in global configuration mode, channel groups including member interfaces must also be created using global configuration mode.
- Port channels that are configured within switch profile mode may have members both inside and outside of a switch profile.
- If you want to import a member interface to a switch profile, the port channel that corresponds with the member interface must also be present within the switch profile.

For more information on switch profiles, see the .

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

  The Cisco Nexus device only supports system level MTU. This attribute cannot be changed on an individual port basis.
- Broadcast/Unicast/Multicast Storm Control setting
- Priority-Flow-Control
- Untagged CoS

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

You can only add interfaces configured with the channel mode set to on to static port channels. You can also only add interfaces configured with the channel mode as active or passive to port channels that are running LACP. You can configure these attributes on an individual member port.

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

- Bandwidth
- MAC address
- Spanning Tree Protocol

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

- Description
- CDP
- LACP port priority
- Debounce

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

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

Load Balancing Using Port Channels

Cisco NX-OS load balances traffic across all operational interfaces in a port channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. Port channels provide load balancing by default.

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 4: Port Channel Load-Balancing Criteria

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

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

Fabric Extenders are not configurable individually. Fabric extender configurations are defined on the Cisco Nexus device. In the case of the port-channel load balancing protocol, the table below illustrates which port-channel load balancing option is automatically configured on the fabric extender modules as a result of the configuration performed on the Cisco Nexus device.

The following table shows the criteria used for each configuration:

Table 5: Port channel Load-Balancing Criteria for the Cisco Nexus 2232 and Cisco Nexus 2248 Fabric Extenders

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

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

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

Note The hardware multicast hw-hash command is not supported on Cisco Nexus 3500 Series switches. It is recommended not to configure this command on these switches.

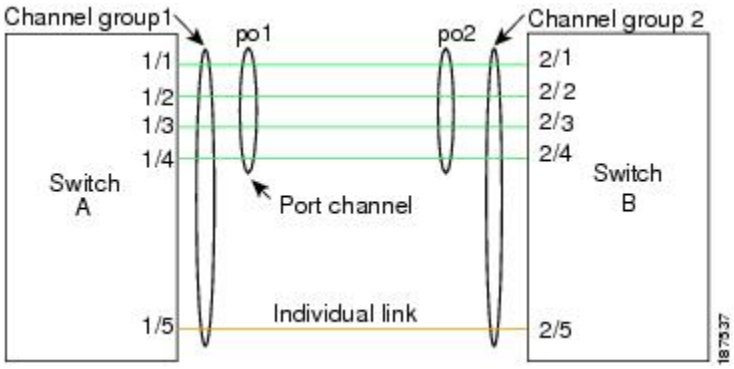
Understanding LACP

LACP Overview

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

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

Figure 6: 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 6: Channel Modes for Individual Links in a Port Channel

Channel Mode	Description
passive	LACP mode that places a port into a passive negotiating state, in which the port responds to LACP packets that it receives but does not initiate LACP negotiation.
active	LACP mode that places a port into an active negotiating state, in which the port initiates negotiations with other ports by sending LACP packets.
on	All static port channels, that is, that are not running LACP, remain in this mode. If you attempt to change the channel mode to active or passive before enabling LACP, the device returns an error message. You enable LACP on each channel by configuring the interface in that channel for the channel mode as either active or passive. When an LACP attempts to negotiate with an interface in the on state, it does not receive any LACP packets and becomes an individual link with that interface; it does not join the LACP channel group.

Both the passive and active modes allow LACP to negotiate between ports to determine if they can form a port channel, based on criteria such as the port speed and the trunking state. The passive mode is useful when you do not know whether the remote system, or partner, supports LACP.

Ports can form an LACP port channel when they are in different LACP modes as long as the modes are compatible as in the following examples:

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

LACP Marker Responders

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

LACP-Enabled and Static Port Channel Differences

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

Table 7: Port Channels with LACP Enabled and Static Port Channels

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

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

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port-channel interface to configure, and enters the interface configuration mode. The range is from 1 to 4096. Cisco NX-OS automatically creates the channel group if it does not already exist.
Step 3	switch(config)# **no interface port-channel** *channel-number*	Removes the port channel and deletes the associated channel group.

This example shows how to create a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
```

Adding a Port to a Port Channel

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Specifies the interface that you want to add to a channel group and enters the interface configuration mode.
Step 3	switch(config-if)# **switchport mode trunk**	(Optional) Configures the interface as a trunk port.
Step 4	switch(config-if)# **switchport trunk** {**allowed vlan** *vlan-id* | **native vlan** *vlan-id*}	(Optional) Configures necessary parameters for a trunk port.
Step 5	switch(config-if)# **channel-group** *channel-number*	Configures the port in a channel group and sets the mode. The channel-number range is from 1 to 4096. Cisco NX-OS creates the port channel associated with this channel group if the port channel does not already exist. This is called implicit port channel creation.
Step 6	switch(config-if)# **no channel-group**	(Optional) Removes the port from the channel group. The port reverts to its original configuration.

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

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

Configuring Load Balancing Using Port Channels

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

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-channel load-balance ethernet** {[**destination-ip** | **destination-mac** | **destination-port** | **source-dest-ip** | **source-dest-mac** | **source-dest-port** | **source-ip** | **source-mac** | **source-port**] | **crc-poly**}	Specifies the load-balancing algorithm for the device. The range depends on the device. The default is **source-dest-ip**. **Note** • source-dest-ip-only • source-dest-port-only • source-dest-ip • source-dest-port • source-dest-ip-gre The Cisco Nexus 5500 Platform switches support 8 hash polynomials that can be used for compression on the hash-parameters. Depending on variations in the hash parameters for egress traffic flows from a port channel, different polynomials could provide different load distribution results. The default hash polynomial is CRC8a. The variable can be configured as follows: • CRC8a • CRC8b • CRC8c • CRC8d • CRC8e • CRC8f • CRC8g
Step 3	switch(config)# **no port-channel load-balance ethernet**	(Optional) Restores the default load-balancing algorithm of source-dest-ip.
Step 4	switch# **show port-channel load-balance**	(Optional) Displays the port-channel load-balancing algorithm.

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

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

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

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Selects the port channel and enters the interface configuration mode.
Step 3	switch(config-if)# **[no] hardware multicast hw-hash**	Configures hardware hashing for the specified port channel.

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

```
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:

```
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

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

	Command or Action	Purpose
Step 2	switch(config)# **feature lacp**	Enables LACP on the switch.
Step 3	switch(config)# **show feature**	(Optional) Displays enabled features.

This example shows how to enable LACP:

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

Configuring the Channel Mode for a Port

You can configure the channel mode for each individual link in the LACP port channel as active or passive. This channel configuration mode allows the link to operate with LACP.

When you configure port channels with no associated protocol, all interfaces on both sides of the link remain in the on channel mode.

Before You Begin

Ensure that you have enabled the LACP feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]	Specifies the port mode for the link in a port channel. After LACP is enabled, you configure each link or the entire channel as active or passive. **force—**Specifies that the LAN port be forcefully added to the channel group. **mode—**Specifies the port channel mode of the interface. **active—**Specifies that when you enable LACP, this command enables LACP on the specified interface. The interface is in an active negotiating state in which the port initiates negotiations with other ports by sending LACP packets. **on—**(Default mode) Specifies that all port channels that are not running LACP remain in this mode. **passive—**Enables LACP only if an LACP device is detected. The interface is in a passive negotiation state in which the port responds

	Command or Action	Purpose
		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 the LACP Fast Timer Rate

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the **lacp rate** command to set the rate at which LACP control packets are sent to an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces.

Before You Begin

Ensure that you have enabled the LACP feature.

Procedure

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

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

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **lacp system-priority** *priority*	Configures the system priority for use with LACP. Valid values are 1 through 65535, and higher numbers have lower priority. The default value is 32768.
Step 3	switch# **show lacp system-identifier**	(Optional) Displays the LACP system identifier.

This example shows how to set the LACP system priority to 2500:

```
switch# configure terminal
switch(config)# lacp system-priority 2500
```

Configuring the LACP Port Priority

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

Before You Begin

Ensure that you have enabled the LACP feature.

Procedure

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

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

This example shows how to set the LACP port priority for Ethernet interface 1/4 to 40000:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# lacp port priority 40000
```

Disabling LACP Graceful Convergence

Before You Begin

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

Procedure

	Command or Action	Purpose

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

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

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

	Command or Action	Purpose

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

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

Step 6	**copy running-config startup-config** **Example:** `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.

The following example disables LACP graceful convergence on a port channel:

```
switch# configure terminal
switch(config) # interface port-channel 1
switch(config-if) # shutdown
switch(config-if) # no lacp graceful-convergence
switch(config-if) # no shutdown
switch(config-if) #
```

Reenabling LACP Graceful Convergence

Before You Begin

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

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

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

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

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

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

Step 6	**copy running-config startup-config** **Example:** `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.

The following example disables LACP graceful convergence on a port channel:

```
switch# configure terminal
switch(config) # interface port-channel 1
switch(config-if) # shutdown
switch(config-if) # lacp graceful-convergence
switch(config-if) # no shutdown
switch(config-if) #
```

Verifying Port Channel Configuration

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

Command	Purpose
show interface port channel*channel-number*	Displays the status of a port channel interface.
show feature	Displays enabled features.
show resource	Displays the number of resources currently available in the system.

Command	Purpose
show lacp {**counters** | **interface** *type slot/port* | **neighbor** | **port-channel** | **system-identifier**}	Displays LACP information.
show port-channel compatibility-parameters	Displays the parameters that must be the same among the member ports in order to join a port channel.
show port-channel database [**interface port-channel** *channel-number*]	Displays the aggregation state for one or more port-channel interfaces.
show port-channel summary	Displays a summary for the port channel interfaces.
show port-channel traffic	Displays the traffic statistics for port channels.
show port-channel usage	Displays the range of used and unused channel numbers.
show port-channel database	Displays information on current running of the port channel feature.
show port-channel load-balance	Displays information about load-balancing using port channels. Based on the system—Source and destination IP, and MAC addresses are used for port channel load balance calculation. By default the **source-destination-ip** address is used. Based on protocol—For non-IP traffic, only the source and destination MAC address is used, and for IP traffic, both source and destination IP and MAC addresses are used for the port channel load balance calculation.

Verifying the Load-Balancing Outgoing Port ID

Command Guidelines

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

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

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

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

Example

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

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

CHAPTER 5

Configuring Virtual Port Channels

This chapter contains the following sections:

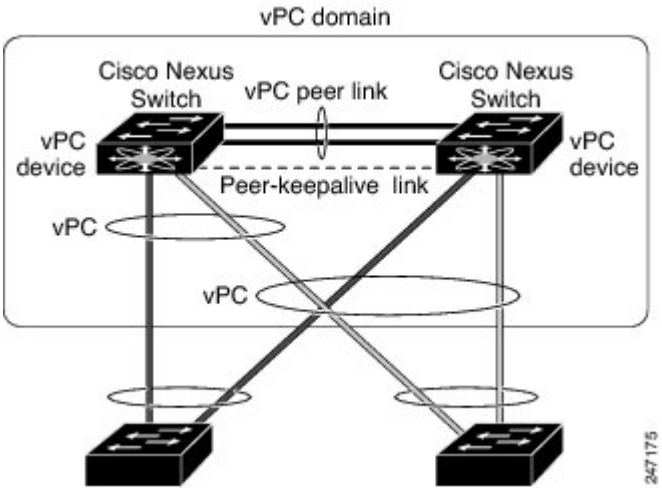
Information About vPCs

vPC Overview

A virtual port channel (vPC) allows links that are physically connected to two different Cisco Nexus devices or Cisco Nexus Fabric Extenders to appear as a single port channel by a third device (see the following figure). The third device can be a switch, server, or any other networking device. You can configure vPCs in topologies that include Cisco Nexus devices connected to Cisco Nexus Fabric Extenders. A vPC can provide multipathing,

which allows you to create redundancy by enabling multiple parallel paths between nodes and load balancing traffic where alternative paths exist.

Figure 7: vPC Architecture



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. When you configure a vPC on a Fabric Extender, only one port is allowed in an EtherChannel.

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

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

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

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

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

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

A vPC provides the following benefits:

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

Terminology

vPC Terminology

The terminology used in vPCs is as follows:

- vPC—combined EtherChannel between the vPC peer devices and the downstream device.
- vPC peer device—One of a pair of devices that are connected with the special EtherChannel known as the vPC peer link.
- vPC peer link—link used to synchronize states between the vPC peer devices.
- vPC member port—Interfaces that belong to the vPCs.
- Host vPC port—Fabric Extender host interfaces that belong to a vPC.
- vPC domain—domain that includes both vPC peer devices, the vPC peer-keepalive link, and all of the port channels in the vPC connected to the downstream devices. It is also associated to the configuration mode that you must use to assign vPC global parameters. The vPC domain ID must be the same on both switches.
- vPC peer-keepalive link—The peer-keepalive link monitors the vitality of a vPC peer Cisco Nexus device. The peer-keepalive link sends configurable, periodic keepalive messages between vPC peer devices.

  No data or synchronization traffic moves over the vPC peer-keepalive link; the only traffic on this link is a message that indicates that the originating switch is operating and running vPCs.

Fabric Extender Terminology

The terminology used for the Cisco Nexus Fabric Extender is as follows:

- Fabric interface—A 10-Gigabit Ethernet uplink port designated for connection from the Fabric Extender to its parent switch. A fabric interface cannot be used for any other purpose. It must be directly connected to the parent switch.
- EtherChannel fabric interface—An EtherChannel uplink connection from the Fabric Extender to its parent switch. This connection consists of fabric interfaces bundled into a single logical channel.

- Host interface—An Ethernet interface for server or host connectivity. These ports are 1-Gigabit Ethernet interfaces or 10-Gigabit Ethernet interfaces, depending on the fabric extender model.
- EtherChannel host interface—An EtherChannel downlink connection from the Fabric Extender host interface to a server port.

Note An EtherChannel host interface consists of only one host interface and can be configured either as a Link Aggregation Control Protocol (LACP) or non-LACP EtherChannel.

Supported vPC Topologies

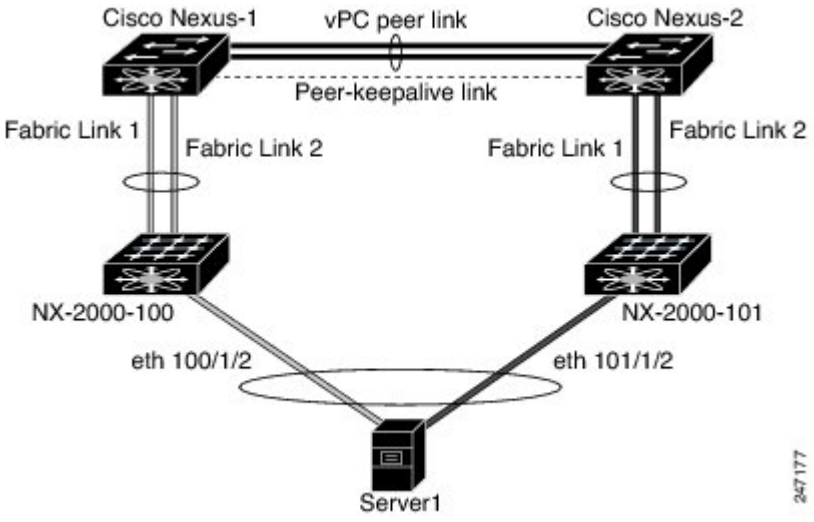
Cisco Nexus Device vPC Topology

Single Homed Fabric Extender vPC Topology

You can connect a server with dual or quad or more network adapters that are configured in a vPC to a pair of Cisco Nexus Fabric Extenders which are connected to the Cisco Nexus devices as depicted. Depending on the FEX model, you may be able to connect one or more network adapter interfaces to each fabric extender. As an example, the following figure refers to a topology built with the Cisco Nexus 2148T fabric extender, where a server has one link only to each fabric extender. A topology with Cisco Nexus 2248TP or with Cisco Nexus 2232PP fabric extender could consist of more links from the server to a single fabric extender.

. The topology that is shown in the following figure provides the vPC functionality to dual homed servers with 1-Gigabit Ethernet uplink interfaces.

Figure 8: Single Homed Fabric Extender vPC Topology



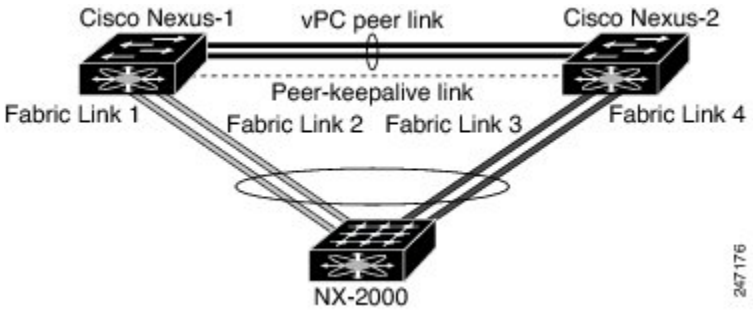
The Cisco Nexus device can support up to 12 configured single homed Fabric Extenders (576 ports) with this topology however only 480 576 dual homed host servers can be configured in a vPCs with this configuration.

Note The Cisco Nexus 2148T fabric extender does not support EtherChannels on its host interfaces. Therefore a maximum of two links can be configured in an EtherChannel from the server where each link is connected to a separate Fabric Extender.

Dual Homed Fabric Extender vPC Topology

You can connect the Cisco Nexus Fabric Extender to two upstream Cisco Nexus devices and downstream to a number of single homed servers. The topology shown in the following figure provides the vPC functionality to singly connected servers with 1-Gigabit Ethernet uplink interfaces.

Figure 9: Dual Homed Fabric Extender vPC Topology



The Cisco Nexus device can support up to 12 configured dual homed Fabric Extenders with this topology. A maximum of 576 single homed servers can be connected to this configuration.

vPC Domain

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

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

The vPC peer switches use the vPC domain ID that you configure to automatically assign a unique vPC system MAC address. Each vPC domain has a unique MAC address that is used as a unique identifier for the specific vPC-related operations, although the switches use the vPC system MAC addresses only for link-scope operations, such as LACP. We recommend that you create each vPC domain within the contiguous network with a unique domain ID. You can also configure a specific MAC address for the vPC domain, rather than having the Cisco NX-OS software assign the address.

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

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

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

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

Peer-Keepalive Link and Messages

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

You can configure a hold-timeout and a timeout value simultaneously.

Hold-timeout value—The hold-timeout value range is between 3 to 10 seconds, with a default value of 3 seconds. This timer starts when the vPC peer link goes down. The purpose of the hold-timeout period is to prevent false-positive cases.

If you configure a hold-timeout value that is lower than the timeout value, then the vPC system ignores vPC peer-keepalive messages for the hold-timeout period and considers messages for the reminder of the timeout period. If no keepalive message is received for this period, the vPC secondary device takes over the role of the primary device. For example, if the hold-timeout value is 3 seconds and the timeout value is 5 seconds, for the first 3 seconds vPC keepalive messages are ignored (such as, when accommodating a supervisor failure for a few seconds after peer link failure) and keepalive messages are considered for the remaining timeout period of 2 seconds. After this period, the vPC secondary device takes over as the primary device, in case there is no keep alive message.

Timeout value—The timeout value range is between 3 to 20 seconds, with a default value of 5 seconds. This timer starts at the end of the hold-timeout interval. If you configure a timeout value that is lower than or equal to the hold-timeout value, then the timeout duration is initiated after the hold-timeout period. For example, if the timeout value is 3 seconds and the hold-timeout value is 5 seconds, the timeout period starts after 5 seconds.

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

Compatibility Parameters for vPC Peer Links

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

Enter the **show vpc consistency-parameters** command to display the configured values on all interfaces in the vPC. The displayed configurations are only those configurations that would limit the vPC peer link and vPC from coming up.

The compatibility check process for vPCs differs from the compatibility check for regular EtherChannels.

Configuration Parameters That Must Be Identical

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

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

Enter the **show vpc consistency-parameters** command to display the configured values on all interfaces in the vPC. The displayed configurations are only those configurations that would limit the vPC peer link and vPC from coming up.

The switch automatically checks for compatibility of these parameters on the vPC interfaces. The per-interface parameters must be consistent per interface, and the global parameters must be consistent globally.

- Port-channel mode: on, off, or active
- Link speed per channel
- Duplex mode per channel
- Trunk mode per channel:
  - Native VLAN
  - VLANs allowed on trunk
  - Tagging of native VLAN traffic
- Spanning Tree Protocol (STP) mode
- STP region configuration for Multiple Spanning Tree (MST)
- Enable or disable state per VLAN
- STP global settings:
  - Bridge Assurance setting
  - Port type setting—We recommend that you set all vPC interfaces as normal ports
  - Loop Guard settings
- STP interface settings:
  - Port type setting
  - Loop Guard
  - Root Guard
- For the Fabric Extender vPC topology, all the interface level parameters mentioned above should be identically configured for host interface from both the switches.

- Fabric Extender FEX number configured on an EtherChannel fabric interface; for the Fabric Extender vPC toplogy.

If any of these parameters are not enabled or defined on either switch, the vPC consistency check ignores those parameters.

Note To ensure that none of the vPC interfaces are in the suspend mode, enter the **show vpc brief** and **show vpc consistency-parameters** commands and check the syslog messages.

Configuration Parameters That Should Be Identical

When any of the following parameters are not configured identically on both vPC peer switches, a misconfiguration might cause undesirable behavior in the traffic flow:

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

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

Graceful Type-1 Check

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.

This feature is not enabled for dual-active FEX ports. When a Type-1 mismatch occurs, VLANs are suspended on these ports on both switches.

Per-VLAN Consistency Check

Beginning with Cisco NX-OS Release 5.0(2)N2(1), 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

Beginning with Cisco NX-OS Release 5.0(2)N2(1), 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 to 3600 seconds.

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

vPC Peer Links

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

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

vPC Peer Link Overview

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

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

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

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

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

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

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

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

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

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

The software learns of a vPC peer switch failure when the keepalive messages are not returned over the peer-keepalive link.

Use a separate link (vPC peer-keepalive link) to send configurable keepalive messages between the vPC peer switches. The keepalive messages on the vPC peer-keepalive link determines whether a failure is on the vPC peer link only or on the vPC peer switch. The keepalive messages are used only when all the links in the peer link fail.

vPC Number

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

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

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

vPC Interactions with Other Features

Configuring vPC Peer Links and Links to the Core

Configure the command line interface by using a track object and a track list that is associated with the Layer 3 link to the core and on all vPC peer links on both vPC peer devices. You use this configuration to avoid dropping traffic if that particular module goes down because when all the tracked objects on the track list go down, the system does the following:

- Stops the vPC primary peer device sending peer-keepalive messages which forces the vPC secondary peer device to take over.
- Brings down all the downstream vPCs on that vPC peer device, which forces all the traffic to be rerouted in the access switch toward the other vPC peer device.

Once you configure this feature and if the module fails, the system automatically suspends all the vPC links on the primary vPC peer device and stops the peer-keepalive messages. This action forces the vPC secondary device to take over the primary role and all the vPC traffic to go to this new vPC primary device until the system stabilizes.

Create a track list that contains all the links to the core and all the vPC peer links as its object. Enable tracking for the specified vPC domain for this track list. Apply this same configuration to the other vPC peer device.

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **track** *track-id* **interface** *type slot/port* **line-protocol**	Configures the track objects on an interface (Layer 3 to core).
Step 4	switch(config-track)# **track** *track-id* **interface** *type slot/port* **line-protocol**	Tracks the objects on an interface (Layer 3 to core).

	Command or Action	Purpose
Step 5	switch(config)# **track** *track-id* **interface port-channel** *port* **line-protocol**	Configures the track objects on a port channel (vPC peer link).
Step 6	switch(config)# **track** *track-id* **list boolean** [**OR** | **AND**]	Creates a track list that contains all the interfaces in the track list using the Boolean OR to trigger when all the objects fail. or trigger a switchover when any core interface or peer-link goes down using Boolean AND.
Step 7	switch(config-track)# **object** *number*	Specifiecs the object number.
Step 8	switch(config-track)# **end**	Exits track configuration mode.
Step 9	switch(config)# **vpc domain** *domain-id*	Enters vPC domain configuration.
Step 10	switch(config-vpc-domain)# **track** *number*	Adds the track object to the vPC domain.
Step 11	switch(config)# **show vpc brief**	(Optional) Displays the track object.
Step 12	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 configure a track list to trigger when all the objects fail using Boolean OR:

```
switch# configure terminal
switch(config)# interface ethernet 8/35
switch(config-if)# track 35 interface ethernet 8/35 line-protocol
switch(config-track)# track 23 interface ethernet 8/33 line-protocol
switch(config)# track 55 interface port-channel 100 line-protocol
switch(config)# track 44 list boolean OR
switch(config-track)# object 23
switch(config-track)# object 35
switch(config-track)# object 55
switch(config-track)# end
switch(config)# vpc domain 1
switch(config-vpc-domain)# track 44
switch(config)# copy running-config startup-config
```

vPC and LACP

The Link Aggregation Control Protocol (LACP) uses the system MAC address of the vPC domain to form the LACP Aggregation Group (LAG) ID for the vPC.

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

The vPC peer link supports 16 EtherChannel interfaces.

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

Note During a vPC role change (where secondary becomes primary), if the ports are in LACP-individual state on FEX, the ports will flap and this will cause brief connectivity loss before the primary is up. To avoid this issue, ensure that port-channel member ports are not configured as LACP-individual state or LACP is not used (On mode).

vPC Peer Links and STP

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

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

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

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

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

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

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

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

vPC and ARP

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

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

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

```
switch# show running
```

To enable ARP sync, enter the following command:

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

CFSoE

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

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

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

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

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

vPC Peer Switch

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

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

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

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

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

Guidelines and Limitations for vPCs

vPC has the following configuration guidelines and limitations:

- You must enable the vPC feature before you can configure vPC peer-link and vPC interfaces.
- You must configure the peer-keepalive link before the system can form the vPC peer link.
- The vPC peer-link needs to be formed using a minimum of two 10-Gigabit Ethernet interfaces.
- You can connect a pair of Cisco Nexus 5000 Series switches or a pair of Cisco Nexus 5500 Series switches in a vPC directly to another switch or to a server. vPC peer switches must be of the same type, for example, you can connect a pair of Nexus 5000 series switches or a pair of Nexus 5500 Series switches but you cannot connect a Nexus 5000 Series switch to a Nexus 5500 Series switch in a vPC topology.
- Only port channels can be in vPCs. A vPC can be configured on a normal port channel (switch-to-switch vPC topology), on a port channel fabric interface (fabric extender vPC topology), and on a port channel host interface (host interface vPC topology).
- A Fabric Extender can be a member of a Host Interface vPC topology or a Fabric Extender vPC topology but not both simultaneously.
- You must configure both vPC peer switches; the configuration is not automatically synchronized between the vPC peer devices.
- Check that the necessary configuration parameters are compatible on both sides of the vPC peer link.
- You may experience minimal traffic disruption while configuring vPCs.
- You should configure all the port channels in the vPC using LACP with the interfaces in active mode.
- When the **peer-switch** command is configured and vPC keepalive messages exchanged through an SVI instead of a management interface, additional Spanning Tree Protocol (STP) configuration is required. STP needs to be disabled on the dedicated link that carries the keepalive traffic between the vPC peers. You can disable STP on the dedicated link by configuring STP BPDUfilter on the both ends of the dedicated link. We recommend that the VLAN of the vPC keepalive SVI be allowed on only the interconnecting dedicated link and disallowed on all other links, including the peer link.
- You cannot have a link for non-vPC traffic in parallel with a vPC topology. This can cause errors with the traffic forwarding logic resulting in duplicate or missed packets.
- You cannot configure non-vPC interfaces across host ports on two different FEXs.

Configuring vPCs

Enabling vPCs

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

Procedure

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

This example shows how to enable the vPC feature:

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

Disabling vPCs

You can disable the vPC feature.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature vpc**	Disables vPCs on the switch.
Step 3	switch# **show feature**	(Optional) Displays which features are enabled on the switch.
Step 4	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to disable the vPC feature:

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

Creating a vPC Domain

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

Before You Begin

Ensure that you have enabled the vPC feature.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Creates a vPC domain on the switch, and enters the vpc-domain configuration mode. There is no default *domain-id* ; the range is from 1 to 1000. **Note** You can also use the **vpc domain** command to enter the vpc-domain configuration mode for an existing vPC domain.
Step 3	switch# **show vpc brief**	(Optional) Displays brief information about each vPC domain.
Step 4	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to create a vPC domain:

```
switch# configure terminal
switch(config)# vpc domain 5
```

Configuring a vPC Keepalive Link and Messages

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

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

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

Note We recommend that you configure a separate VRF instance and put a Layer 3 port from each vPC peer switch into that VRF instance for the vPC peer-keepalive link. Do not use the peer link itself to send vPC peer-keepalive messages. For information on creating and configuring VRFs, see the Unicast Routing Configuration Guide for your device.

Before You Begin

Ensure that you have enabled the vPC feature.

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Creates a vPC domain on the switch if it does not already exist, and enters the vpc-domain configuration mode.
Step 3	switch(config-vpc-domain)# **peer-keepalive destination** *ipaddress* [**hold-timeout** *secs* | **interval** *msecs* {**timeout** *secs*} | **precedence** {*prec-value* | **network** | **internet** | **critical** | **flash-override** | **flash** | **immediate priority** | **routine**} | **tos** {*tos-value* | **max-reliability** | **max-throughput** | **min-delay** | **min-monetary-cost** | **normal**} | **tos-byte** *tos-byte-value*} | **source** *ipaddress* | **vrf** {*name* | **management vpc-keepalive**}]	Configures the IPv4 address for the remote end of the vPC peer-keepalive link. **Note** The system does not form the vPC peer link until you configure a vPC peer-keepalive link. The management ports and VRF are the defaults.
Step 4	switch(config-vpc-domain)# **vpc peer-keepalive destination** *ipaddress* **source** *ipaddress*	(Optional) Configures a separate VRF instance and puts a Layer 3 port from each vPC peer device into that VRF for the vPC peer-keepalive link.
Step 5	switch# **show vpc peer-keepalive**	(Optional) Displays information about the configuration for the keepalive messages.
Step 6	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

```
switch# configure terminal
```

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

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

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

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

```
vrf context vpc_keepalive
interface Ethernet1/31
  switchport access vlan 123
interface Vlan123
  vrf member vpc_keepalive
  ip address 123.1.1.2/30
  no shutdown
vpc domain 1
  peer-keepalive destination 123.1.1.1 source 123.1.1.2 vrf
vpc_keepalive

L3-NEXUS-2# show vpc peer-keepalive

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

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

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

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

Creating a vPC Peer Link

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

as the vPC peer link in trunk mode and that you use two ports on separate modules on each vPC peer switch for redundancy.

Before You Begin

Ensure that you have enabled the vPC feature.

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

Procedure

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

This example shows how to configure a vPC peer link:

```
switch# configure terminal
switch(config)# interface port-channel 20
switch(config-if)# vpc peer-link
```

Checking the Configuration Compatibility

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

Note Beginning with Cisco NX-OS Release 5.0(2)N1(1), the The following QoS parameters support Type 2 consistency checks:

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

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

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

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

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

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

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

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

This example shows how to check that the required configurations are compatible for an EtherChannel interface:

```
switch#  show vpc consistency-parameters interface port-channel 20

    Legend:
        Type 1 : vPC will be suspended in case of mismatch
Name                        Type  Local Value            Peer Value
-------------               ----  ---------------------- -----------------------
Fex id                      1     20                     20
STP Port Type               1     Default                Default
STP Port Guard              1     None                   None
STP MST Simulate PVST       1     Default                Default
mode                        1     on                     on
Speed                       1     10 Gb/s                10 Gb/s
Duplex                      1     full                   full
Port Mode                   1     fex-fabric             fex-fabric
Shut Lan                    1     No                     No
Allowed VLANs               -     1,3-3967,4048-4093     1-3967,4048-4093
```

Enabling vPC Auto-Recovery

Procedure

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

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(2)N2(1)

feature vpc
vpc domain 10
  peer-keepalive destination 10.193.51.170
  auto-recovery
```

Suspending Orphan Ports on a Secondary Switch in a vPC Topology

You can suspend a non-virtual port channel (vPC) port when a vPC secondary peer link goes down. A non-vPC port or an orphan port is a port that is not part of a vPC.

Note When a port is configured as an orphan port, the port will flap. This occurs because the system reevaluates whether the port can be brought up, given the constraints of the orphan port. For example, a Multichassis Etherchannel Trunk (MCT) needs to be up and election needs to be complete.

Before You Begin

Enable the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot*/*port*	Specifies the port that you want to configure and enters interface configuration mode.
Step 3	switch(config-if)# **vpc orphan-port suspend**	Suspends the specified port if the secondary switch goes down. **Note** The **vpc-orphan-port suspend** command is supported only on physical ports, and not supported on port-channel member ports. **Note** From Cisco NX-OS Release 7.3(1)N1(1), **vpc-orphan-port suspend** command is supported on Layer 3 ports too.
Step 4	switch(config-if)# **exit**	Exits interface configuration mode.
Step 5	switch# **show vpc orphan-port**	(Optional) Displays the orphan port configuration.
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 suspend an orphan port:

```
switch# configure terminal
switch(config)# interface ethernet 1/20
switch(config-if)# vpc orphan-port suspend
```

This example shows how to display ports that are not part of the vPC but that share common VLANs with ports that are part of the vPC:

```
switch# configure terminal
switch(config)# show vpc orphan-ports
Note:
--------::Going through port database. Please be patient.::--------
VLAN Orphan Ports
------- -------------------------
1 Po600
2 Po600
3 Po600
4 Po600
5 Po600
6 Po600
7 Po600
8 Po600
9 Po600
10 Po600
11 Po600
12 Po600
13 Po600
14 Po600
...
```

Creating an EtherChannel Host Interface

To connect to a downstream server from a Cisco Nexus Fabric Extender you can create a EtherChannel host interface. An EtherChannel host interface can have only one host interface as a member depending on the fabric extender model. The Cisco Nexus 2148T allows only one interface member per fabric extender, newer fabric extenders allow up to 8 members of the same port-channel on a single fabric extender. You need to create an EtherChannel host interface to configure a vPC on it that uses the Fabric Extender topology.

Before You Begin

Ensure that you have enabled the vPC feature.

Ensure that the connected Fabric Extender is online.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *chassis*/*slot*/*port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **channel-group** *channel-number* **mode** {**active** | **passive** | **on**}	Creates an EtherChannel host interface on the selected host interface.
Step 4	switch(config-if)# **show port-channel summary**	(Optional) Displays information about each EtherChannel host interface.
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.

This example shows how to configure an EtherChannel host interface:

```
switch# configure terminal
switch(config)# interface ethernet 101/1/20
switch(config-if)# channel-group 7 mode active
```

Moving Other Port Channels into a vPC

Before You Begin

Ensure that you have enabled the vPC feature.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Selects the port channel that you want to put into the vPC to connect to the downstream switch, and enters interface configuration mode. **Note** A vPC can be configured on a normal port channel (physical vPC topology), on an port channel fabric interface (fabric extender vPC topology), and on a port channel host interface (host interface vPC topology)
Step 3	switch(config-if)# **vpc** *number*	Configures the selected port channel into the vPC to connect to the downstream switch. The range is from 1 to 4096. The vPC *number* that you assign to the port channel that connects to the downstream switch from the vPC peer switch must be identical on both vPC peer switches.
Step 4	switch# **show vpc brief**	(Optional) Displays information about each vPC.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

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

```
switch# configure terminal
switch(config)# interface port-channel 20
switch(config-if)# vpc 5
```

Manually Configuring a vPC Domain MAC Address

Note Configuring the system address is an optional configuration step.

Before You Begin

Ensure that you have enabled the vPC feature.

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

Procedure

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

	Command or Action	Purpose
Step 2	switch(config)# **vpc domain** *domain-id*	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default *domain-id*; the range is from 1 to 1000.
Step 3	switch(config-vpc-domain)# **system-mac** *mac-address*	Enters the MAC address that you want for the specified vPC domain in the following format: aaaa.bbbb.cccc.
Step 4	switch# **show vpc role**	(Optional) Displays the vPC system MAC address.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a vPC domain MAC address:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-if)# system-mac 23fb.4ab5.4c4e
```

Manually Configuring the System Priority

When you create a vPC domain, the system automatically creates a vPC system priority. However, you can also manually configure a system priority for the vPC domain.

Before You Begin

Ensure that you have enabled the vPC feature.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default *domain-id*; the range is from 1 to 1000.
Step 3	switch(config-vpc-domain)# **system-priority** *priority*	Enters the system priority that you want for the specified vPC domain. The range of values is from 1 to 65535. The default value is 32667.

	Command or Action	Purpose
Step 4	switch# **show vpc brief**	(Optional) Displays information about each vPC, including information about the vPC peer link.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a vPC peer link:

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

Manually Configuring a vPC Peer Switch Role

By default, the Cisco NX-OS software elects a primary and secondary vPC peer switch after you configure the vPC domain and both sides of the vPC peer link. However, you may want to elect a specific vPC peer switch as the primary switch for the vPC. Then, you would manually configure the role value for the vPC peer switch that you want as the primary switch to be lower than the other vPC peer switch.

vPC does not support role preemption. If the primary vPC peer switch fails, the secondary vPC peer switch takes over to become operationally the vPC primary switch. However, the original operational roles are not restored when the formerly primary vPC comes up again.

Before You Begin

Ensure that you have enabled the vPC feature.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default *domain-id*; the range is from 1 to 1000.
Step 3	switch(config-vpc-domain)# **role priority** *priority*	Enters the role priority that you want for the vPC system priority. The range of values is from 1 to 65535. The default value is 32667.

	Command or Action	Purpose
Step 4	switch# **show vpc brief**	(Optional) Displays information about each vPC, including information about the vPC peer link.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a vPC peer link:

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

Configuring the vPC Peer Switch

Configuring a Pure vPC Peer Switch Topology

You can configure a pure vPC peer switch topology using the **peer-switch** command and then you set the best possible (lowest) spanning tree bridge priority value.

Note The values you apply for the spanning tree priority must be identical on both vPC peers.

Before You Begin

Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Enters the vPC domain number that you want to configure. The system enters the vpc-domain configuration mode.
Step 3	switch(config-vpc-domain)# **peer-switch**	Enables the vPC switch pair to appear as a single STP root in the Layer 2 topology. Use the **no** form of the command to disable the peer switch vPC topology.
Step 4	switch(config-vpc-domain)# **spanning-tree vlan** *vlan-range* **priority** *value*	Configures the bridge priority of the VLAN. Valid values are multiples of 4096. The default value is 32768.

	Command or Action	Purpose
		Note This value must be identical on both vPC peers.
Step 5	switch(config-vpn-domain)# **exit**	Exits the vpc-domain configuration mode.
Step 6	switch(config)# **show spanning-tree summary**	(Optional) Displays a summary of the spanning tree port states including the vPC peer switch. Look for the following line in the command output: `vPC peer-switch is enabled (operational)`
Step 7	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 configure a pure vPC peer switch topology:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# vpc domain 5
switch(config-vpc-domain)# peer-switch
2010 Apr 28 14:44:44 switch %STP-2-VPC_PEERSWITCH_CONFIG_ENABLED: vPC peer-switch
configuration is enabled. Please make sure to configure spanning tree "bridge" priority as

per recommended guidelines to make vPC peer-switch operational.
switch(config-vpc-domain)# exit
switch(config)# spanning-tree vlan 1 priority 8192
switch(config)# show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0001-VLAN0050, VLAN0100-VLAN0149, VLAN0200-VLAN0249
  VLAN0300-VLAN0349, VLAN0400-VLAN0599, VLAN0900-VLAN0999
Port Type Default                        is disable
Edge Port [PortFast] BPDU Guard Default  is disabled
Edge Port [PortFast] BPDU Filter Default is disabled
Bridge Assurance                         is enabled
Loopguard Default                        is disabled
Pathcost method used                     is short
vPC peer-switch                          is enabled (operational)
Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
VLAN0001                     0         0        0         16         16
VLAN0002                     0         0        0         16         16
switch(config)# copy running-config startup-config
switch(config)#
```

Configuring a Hybrid vPC Peer Switch Topology

You can configure a hybrid vPC and non-vPC peer switch topology by using the spanning-tree pseudo-information command to change the designated bridge ID so that it meets the STP VLAN-based load-balancing criteria and then change the root bridge ID priority to a value that is better than the best bridge priority. You then enable the peer switch. For more information, see the command reference for your device.

Note If you previously configured global spanning tree parameters and you subsequently configure spanning tree pseudo information parameters, be aware that the pseudo information parameters take precedence over the global parameters.

Before You Begin

Ensure that you have enabled the vPC feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **spanning-tree pseudo-information**	Configures the spanning tree pseudo information. **Note** This configuration takes precedence over any global spanning tree configurations.
Step 3	switch(config-pseudo)# **vlan** *vlan-id* **designated priority** *priority*	Configures the designated bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440.
Step 4	switch(config-pseudo)# **vlan** *vlan-id* **root priority** *priority*	Configures the root bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440. **Note** This value must be identical on both vPC peers to have an operational peer switch.
Step 5	switch(config-pseudo)# **exit**	Exists spanning tree pseudo information configuration mode.
Step 6	switch(config)# **vpc domain** *domain-id*	Enters the vPC domain number that you want to configure. The system enters the vpc-domain configuration mode.
Step 7	switch(config-vpc-domain)# **peer-switch**	Enables the vPC switch pair to appear as a single STP root in the Layer 2 topology. Use the **no** form of the command to disable the peer switch vPC topology.
Step 8	switch(config-vpc-domain)# **exit**	Exits the vpc-domain configuration mode.
Step 9	switch(config)# **show spanning-tree summary**	(Optional) Displays a summary of the spanning tree port states including the vPC peer switch. Look for the following line in the command output: `vPC peer-switch is enabled (operational)`
Step 10	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 configure a hybrid vPC peer switch topology:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# spanning-tree pseudo-information
switch(config-pseudo)# vlan 1 designated priority 8192
switch(config-pseudo)# vlan 1 root priority 4096
switch(config-pseudo)# exit
switch(config)# vpc domain 5
switch(config-vpc-domain)# peer-switch
switch(config-vpc-domain)# exit
switch(config)# copy running-config startup-config
```

Verifying the vPC Configuration

Use the following commands to display vPC configuration information:

Command	Purpose
switch# **show feature**	Displays whether vPC is enabled or not.
switch# **show port-channel capacity**	Displays how many EtherChannels are configured and how many are still available on the switch.
switch# **show running-config vpc**	Displays running configuration information for vPCs.
switch# **show vpc brief**	Displays brief information on the vPCs.
switch# **show vpc consistency-parameters**	Displays the status of those parameters that must be consistent across all vPC interfaces.
switch# **show vpc peer-keepalive**	Displays information on the peer-keepalive messages.
switch# **show vpc role**	Displays the peer status, the role of the local switch, the vPC system MAC address and system priority, and the MAC address and priority for the local vPC switch.
switch# **show vpc statistics**	Displays statistics on the vPCs. **Note** This command displays the vPC statistics only for the vPC peer device that you are working on.

For information about the switch output, see the Command Reference for your Cisco Nexus Series switch.

Viewing the Graceful Type-1 Check Status

This example shows how to display the current status of the graceful Type-1 consistency check:

```
switch# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link
```

```
vPC domain id                    : 10
Peer status                      : peer adjacency formed ok
vPC keep-alive status            : peer is alive
Configuration consistency status: success
Per-vlan consistency status      : success
Type-2 consistency status        : success
vPC role                         : secondary
Number of vPCs configured        : 34
Peer Gateway                     : Disabled
Dual-active excluded VLANs       : -
Graceful Consistency Check       : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ -------------------------------------------------
1    Po1    up     1
```

Viewing a Global Type-1 Inconsistency

When a global Type-1 inconsistency occurs, the vPCs on the secondary switch are brought down. The following example shows this type of inconsistency when there is a spanning-tree mode mismatch.

The example shows how to display the status of the suspended vPC VLANs on the secondary switch:

```
switch(config)# show vpc
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                    : 10
Peer status                      : peer adjacency formed ok
vPC keep-alive status            : peer is alive
Configuration consistency status: failed
Per-vlan consistency status      : success
Configuration consistency reason: vPC type-1 configuration incompatible - STP
                                  Mode inconsistent
Type-2 consistency status        : success
vPC role                         : secondary
Number of vPCs configured        : 2
Peer Gateway                     : Disabled
Dual-active excluded VLANs       : -
Graceful Consistency Check       : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ -------------------------------------------------
1    Po1    up     1-10

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        down*  failed      Global compat check failed -
30     Po30        down*  failed      Global compat check failed -
```

The example shows how to display the inconsistent status (the VLANs on the primary vPC are not suspended) on the primary switch:

```
switch(config)# show vpc
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                    : 10
Peer status                      : peer adjacency formed ok
vPC keep-alive status            : peer is alive
Configuration consistency status: failed
Per-vlan consistency status      : success
```

```
Configuration consistency reason: vPC type-1 configuration incompatible - STP Mo
de inconsistent
Type-2 consistency status       : success
vPC role                        : primary
Number of vPCs configured       : 2
Peer Gateway                    : Disabled
Dual-active excluded VLANs      : -
Graceful Consistency Check      : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ -------------------------------------------------
1    Po1    up     1-10

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     failed      Global compat check failed 1-10
30     Po30        up     failed      Global compat check failed 1-10
```

Viewing an Interface-Specific Type-1 Inconsistency

When an interface-specific Type-1 inconsistency occurs, the vPC port on the secondary switch is brought down while the primary switch vPC ports remain up.The following example shows this type of inconsistency when there is a switchport mode mismatch.

This example shows how to display the status of the suspended vPC VLAN on the secondary switch:

```
switch(config-if)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                   : 10
Peer status                     : peer adjacency formed ok
vPC keep-alive status           : peer is alive
Configuration consistency status: success
Per-vlan consistency status     : success
Type-2 consistency status       : success
vPC role                        : secondary
Number of vPCs configured       : 2
Peer Gateway                    : Disabled
Dual-active excluded VLANs      : -
Graceful Consistency Check      : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ -------------------------------------------------
1    Po1    up     1

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     success     success                    1
30     Po30        down*  failed      Compatibility check failed -
                                       for port mode
```

This example shows how to display the inconsistent status (the VLANs on the primary vPC are not suspended) on the primary switch:

```
switch(config-if)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link
```

```
vPC domain id                     : 10
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status: success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : primary
Number of vPCs configured         : 2
Peer Gateway                      : Disabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ -------------------------------------------------
1    Po1    up     1

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     success     success                    1
30     Po30        up     failed      Compatibility check failed 1
                                        for port mode
```

Viewing a Per-VLAN Consistency Status

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

This example shows how to display the consistent status of the VLANs on the primary and the secondary switches.

```
switch(config-if)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 10
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status: success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : secondary
Number of vPCs configured         : 2
Peer Gateway                      : Disabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ -------------------------------------------------
1    Po1    up     1-10

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     success     success                    1-10
30     Po30        up     success     success                    1-10
```

Entering **no spanning-tree vlan 5** command triggers the inconsistency on the primary and secondary VLANs:

```
switch(config)# no spanning-tree vlan 5
```

This example shows how to display the per-VLAN consistency status as Failed on the secondary switch:

```
switch(config)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 10
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status: success
Per-vlan consistency status       : failed
Type-2 consistency status         : success
vPC role                          : secondary
Number of vPCs configured         : 2
Peer Gateway                      : Disabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ -------------------------------------------------
1    Po1    up     1-4,6-10

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     success     success                    1-4,6-10
30     Po30        up     success     success                    1-4,6-10
```

This example shows how to display the per-VLAN consistency status as Failed on the primary switch:

```
switch(config)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 10
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status: success
Per-vlan consistency status       : failed
Type-2 consistency status         : success
vPC role                          : primary
Number of vPCs configured         : 2
Peer Gateway                      : Disabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ -------------------------------------------------
1    Po1    up     1-4,6-10

vPC status
----------------------------------------------------------------------------
id     Port        Status Consistency Reason                     Active vlans
------ ----------- ------ ----------- -------------------------- -----------
20     Po20        up     success     success                    1-4,6-10
30     Po30        up     success     success                    1-4,6-10
```

This example shows the inconsistency as STP Disabled:

```
switch(config)# show vpc consistency-parameters vlans

Name                        Type  Reason Code            Pass Vlans

-------------               ----  ---------------------- ------------------------
STP Mode                    1     success                0-4095
STP Disabled                1     vPC type-1             0-4,6-4095
                                  configuration
```

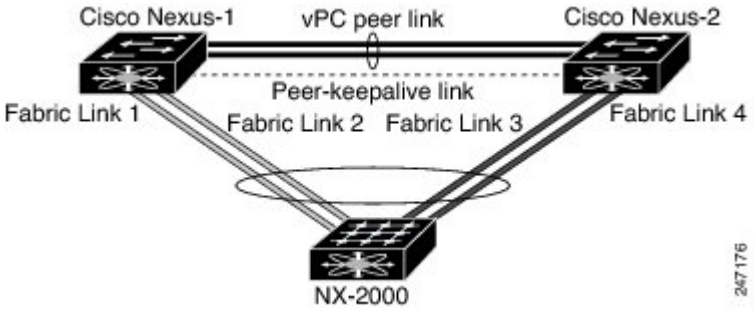
```
                                    incompatible - STP is
                                    enabled or disabled on
                                     some or all vlans
STP MST Region Name         1       success                 0-4095
STP MST Region Revision     1       success                 0-4095
STP MST Region Instance to  1       success                 0-4095
 VLAN Mapping
STP Loopguard               1       success                 0-4095
STP Bridge Assurance        1       success                 0-4095
STP Port Type, Edge         1       success                 0-4095
BPDUFilter, Edge BPDUGuard
STP MST Simulate PVST       1       success                 0-4095
Pass Vlans                  -                               0-4,6-4095
```

vPC Example Configurations

Dual Homed Fabric Extender vPC Configuration Example

The following example shows how to configure the dual homed Fabric Extender vPC topology using the management VRF to carry the peer-keepalive messages on switch CiscoNexus-1 as shown in following figure:

Figure 10: vPC Configuration Example



Before You Begin

Ensure that the Cisco Nexus 2000 Series Fabric Extender NX-2000-100 is attached and online.

Procedure

Step 1 Enable vPC and LACP.

```
CiscoNexus-1# configure terminal
CiscoNexus-1(config)# feature lacp
CiscoNexus-1(config)# feature vpc
```

Step 2 Create the vPC domain and add the vPC peer-keepalive link.

```
CiscoNexus-1(config)# vpc domain 1
CiscoNexus-1(config-vpc-domain)# peer-keepalive destination 10.10.10.237
CiscoNexus-1(config-vpc-domain)# exit
```

Step 3 Configure the vPC peer link as a two port Etherchannel.

```
CiscoNexus-1(config)# interface ethernet 1/1-2
CiscoNexus-1(config-if-range)# switchport mode trunk
CiscoNexus-1(config-if-range)# switchport trunk allowed vlan 20-50
CiscoNexus-1(config-if-range)# switchport trunk native vlan 20
CiscoNexus-1(config-if-range)# channel-group 20 mode active
CiscoNexus-1(config-if-range)# exit
CiscoNexus-1(config)# interface port-channel 20
CiscoNexus-1(config-if)# vpc peer-link
CiscoNexus-1(config-if)# exit
```

Step 4 Create a Fabric Extender identifier (for example, "100").

```
CiscoNexus-1(config)# fex 100
CiscoNexus-1(config-fex)# pinning max-links 1
CiscoNexus-1(fex)# exit
```

Step 5 Configure the fabric EtherChannel links for the Fabric Extender 100.

```
CiscoNexus-1(config)# interface ethernet 1/20
CiscoNexus-1(config-if)# channel-group 100
CiscoNexus-1(config-if)# exit
CiscoNexus-1(config)# interface port-channel 100
CiscoNexus-1(config-if)# switchport mode fex-fabric
CiscoNexus-1(config-if)# vpc 100
CiscoNexus-1(config-if)# fex associate 100
CiscoNexus-1(config-if)# exit
```

Step 6 Configure each host interface port on the Fabric Extender 100 on both Cisco Nexus devices as for all the other steps.

```
CiscoNexus-1(config)# interface ethernet 100/1/1-48
CiscoNexus-1(config-if)# switchport mode access
CiscoNexus-1(config-if)# switchport access vlan 50
CiscoNexus-1(config-if)# no shutdown
CiscoNexus-1(config-if)# exit
```

Step 7 Save the configuration.

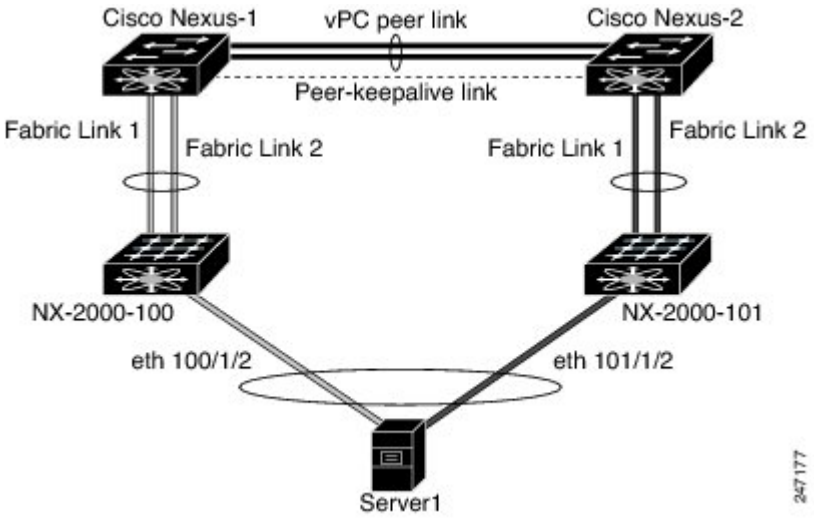
```
CiscoNexus-1(config)# copy running-config startup-config
```

Repeat all the above steps for the CiscoNexus-2 switch.

Single Homed Fabric Extender vPC Configuration Example

The following example shows how to configure the single homed Fabric Extender vPC topology using the default VRF to carry the peer-keepalive messages on switch CiscoNexus-1 as shown in following figure:

Figure 11: vPC Configuration Example



Note The following example only shows the configuration of CiscoNexus-1 which is connected to the Fabric Extender NX-2000-100. You must repeat these steps on its vPC peer, CiscoNexus-2, which is connected to the Fabric Extender NX-2000-101.

Before You Begin

Ensure that the Cisco Nexus 2000 Series Fabric Extenders NX-2000-100 and NX-2000-101 are attached and online.

Procedure

Step 1 Enable vPC and LACP.

```
CiscoNexus-1# configure terminal
CiscoNexus-1(config)# feature lacp
CiscoNexus-1(config)# feature vpc
```

Step 2 Enable SVI interfaces, create the VLAN and SVI to be used by the vPC peer-keepalive link.

```
CiscoNexus-1(config)# feature interface-vlan
CiscoNexus-1(config)# vlan 900
CiscoNexus-1(config-vlan)# int vlan 900
CiscoNexus-1(config-if)# ip address 10.10.10.236 255.255.255.0
CiscoNexus-1(config-if)# no shutdown
CiscoNexus-1(config-if)# exit
```

Step 3 Create the vPC domain and add the vPC peer-keepalive link in the default VRF.

```
CiscoNexus-1(config)# vpc domain 30
CiscoNexus-1(config-vpc-domain)# peer-keepalive destination 10.10.10.237 source 10.10.10.236
 vrf default
CiscoNexus-1(config-vpc-domain)# exit
```

Note VLAN 900 must **not** be trunked across the vPC peer-link because it carries the vPC peer-keepalive messages. There must be an alternative path between switches CiscoNexus-1 and CiscoNexus-2 for the vPC peer-keepalive messages.

Step 4 Configure the vPC peer link as a two port Etherchannel.

```
CiscoNexus-1(config)# interface ethernet 1/1-2
CiscoNexus-1(config-if-range)# switchport mode trunk
CiscoNexus-1(config-if-range)# switchport trunk allowed vlan 20-50
CiscoNexus-1(config-if-range)# switchport trunk native vlan 20
CiscoNexus-1(config-if-range)# channel-group 30 mode active
CiscoNexus-1(config-if-range)# exit
CiscoNexus-1(config)# interface port-channel 30
CiscoNexus-1(config-if)# vpc peer-link
CiscoNexus-1(config-if)# exit
```

Step 5 Configure the Fabric Extender NX-2000-100.

```
CiscoNexus-1(config)# fex 100
CiscoNexus-1(config-fex)# pinning max-links 1
CiscoNexus-1(fex)# exit
```

Step 6 Configure the fabric EtherChannel links for the Fabric Extender NX-2000-100.

```
CiscoNexus-1(config)# interface ethernet 1/20-21
CiscoNexus-1(config-if)# channel-group 100
CiscoNexus-1(config-if)# exit
CiscoNexus-1(config)# interface port-channel 100
CiscoNexus-1(config-if)# switchport mode fex-fabric
CiscoNexus-1(config-if)# fex associate 100
CiscoNexus-1(config-if)# exit
```

Step 7 Configure a vPC server port on on the Fabric Extender NX-2000-100.

```
CiscoNexus-1(config-if)# interface ethernet 100/1/1
CiscoNexus-1(config-if)# switchport mode trunk
CiscoNexus-1(config-if)# switchport trunk native vlan 100
CiscoNexus-1(config-if)# switchport trunk allowed vlan 100-105
CiscoNexus-1(config-if)# channel-group 600
CiscoNexus-1(config-if)# no shutdown
CiscoNexus-1(config-if)# exit
CiscoNexus-1(config)# interface port-channel 600
CiscoNexus-1(config-if)# vpc 600
CiscoNexus-1(config-if)# no shutdown
CiscoNexus-1(config-if)# exit
```

Step 8 Save the configuration.

```
CiscoNexus-1(config)# copy running-config startup-config
```

vPC Default Settings

The following table lists the default settings for vPC parameters.

Table 8: Default vPC Parameters

Parameters	Default
vPC system priority	32667
vPC peer-keepalive message	Disabled
vPC peer-keepalive interval	1 second
vPC peer-keepalive timeout	5 seconds
vPC peer-keepalive UDP port	3200

INDEX

G

H

I

L

M

N

P

R

S

T

U

V