Send document comments to nexus1k-docfeedback@cisco.com.

Cisco Nexus 1000V Interface Configuration Guide, Release 4.2(1) SV1(5.1)

January 20, 2012

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

Text Part Number: OL-25379-01

Cisco Nexus 1000V Interface Configuration Guide, Release 4.2(1) SV1(5.1)

CONTENTS

New and Changed Information

This chapter lists new or changed content in this document by software release, and where it is located.

Feature	Description	Changed in Release	Where Documented
Backup subgroups	You can assign up to seven backup subgroups when pinning the primary subgroup.	4.2(1)SV1(4a)	Chapter 5, “Configuring Port Channels”
Port channel relative numbering	The subgroup numbering begins at zero and is not tied to the vmnic number.	4.2(1)SV1(4a)	Chapter 5, “Configuring Port Channels”
Port channel vPC-HM	The interface **sub-group cdp** command is removed from port channel vPC-HM configuration when connecting to multiple upstream switches.	4.2(1)SV1(4)	Chapter 5, “Configuring Port Channels”
Network state tracking for vPC-HM	Pinpoints link failure on a port channel configured for vPC-HM.	4.2(1)SV1(4)	Chapter 5, “Configuring Port Channels”
VEM management of LACP	You can offload operation of the LACP protocol from the VSM to the VEMs.	4.2(1)SV1(4)	Chapter 5, “Configuring Port Channels”
LACP	You can enable the LACP port channel function by turning on the feature using the command, **feature lacp**.	4.2(1)SV1(4)	Chapter 5, “Configuring Port Channels”
System Jumbo MTU	The system jumbo MTU value is fixed at 9000 and cannot be changed.	4.2(1)SV1(4)	Chapter 2, “Configuring Interface Parameters”
Interface MTU	You can configure an interface MTU between 1500 and 9000.	4.2(1)SV1(4)	Chapter 2, “Configuring Interface Parameters”
Mapping vEthernet interfaces to connected ports	vEthernet interfaces are now mapped to connected ports by MAC address as well as DVPort number.	4.2(1)SV1(4)	Chapter 4, “Configuring Virtual Ethernet Interfaces”

Feature	Description	Changed in Release	Where Documented
Global vEthernet interface controls	You can enable or disable the following automatic vEthernet interface controls: • Deleting unused vEthernet interfaces • Purging of manual vEthernet configurations • Creating vEthernet interfaces	4.2(1)SV1(4)	Chapter 4, "Configuring Virtual Ethernet Interfaces"
Configuration limits	Configuration limits for vEthernet interfaces, vEthernet trunks, and port profiles were added.	4.0(4)SV1(2)	Chapter 7, "Interface Configuration Limits"
show interface vethernet command	The **show interface vethernet** command now displays 5 minute input and output packet/bit rate statistics for the interfaces that you specify. The configuration example showing this command output was updated to reflect this change. **Note** The **show interface ethernet** command output also provides these new statistics.	4.0(4)SV1(2)	Chapter 4, "Configuring Virtual Ethernet Interfaces"
vPC-Host Mode	Support for manual creation of subgroups.	4.0(4)SV1(2)	Chapter 5, "Configuring Port Channels"
Static Pinning	Support for attaching (or pinning) a vEthernet interface to a specific port channel subgroup.	4.0(4)SV1(2)	Chapter 5, "Configuring Port Channels"

Preface

The *Cisco Nexus 1000V Interface Configuration Guide, Release 4.2(1) SV1(5.1)*, provides information about configuring interfaces, although port profiles are the preferred method for configuring interfaces.

This preface describes the following aspects of this document:

Audience

This guide is for network administrators with the following experience and knowledge:

- An understanding of virtualization
- Using VMware tools to configure a vswitch

Note Note: Knowledge of VMware vNetwork Distributed Switch is not a prerequisite.

Document Organization

This publication is organized as follows:

Chapter and Title	Description
Chapter 1, “Overview”	Provides an overview of Cisco Nexus 1000V interfaces.
Chapter 2, “Configuring Interface Parameters”	Describes the basic Cisco Nexus 1000V interface configuration.
Chapter 3, “Configuring Layer 2 Interfaces”	Describes how to configure Cisco Nexus 1000V access and trunk interfaces.

Chapter and Title	Description
Chapter 4, "Configuring Virtual Ethernet Interfaces"	Describes how to configure Cisco Nexus 1000V virtual Ethernet interfaces.
Chapter 5, "Configuring Port Channels"	Describes how to configure Cisco Nexus 1000V port channels.
Chapter 6, "Supported RFCs"	Lists the IETF RFCs supported in Cisco Nexus 1000V Beta 1 release.
Chapter 7, "Interface Configuration Limits"	Lists the maximum configuration limits for interface features.

Document Conventions

Command descriptions use these conventions:

boldface font	Commands and keywords are in boldface.
italic font	Arguments for which you supply values are in italics.
{ }	Elements in braces are required choices.
[ ]	Elements in square brackets are optional.
x | y | z	Alternative, mutually exclusive elements are separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Screen examples use these conventions:

`screen font`	Terminal sessions and information the device 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 for notes and cautions:

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.

Recommended Reading

Before configuring the Cisco Nexus 1000V, we recommend that you read and become familiar with the following documentation:

- *Cisco Nexus 1000V Getting Started Guide, Release 4.2(1)SV1(5.1)*
- *Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(5.1)*
- *Cisco VN-Link: Virtualization-Aware Networking* white paper

Related Documentation

This section lists the documents used with the Cisco Nexus 1000 and available on Cisco.com at the following URL:

http://www.cisco.com/en/US/products/ps9902/tsd_products_support_series_home.html

General Information

Cisco Nexus 1000V Documentation Roadmap, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V Release Notes, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V Compatibility Information, Release 4.2(1)SV1(5.1)

Cisco Nexus 1010 Management Software Release Notes, Release 4.2(1)SP1(3)

Install and Upgrade

Cisco Nexus 1000V Installation and Upgrade Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1010 Virtual Services Appliance Hardware Installation Guide

Cisco Nexus 1010 Software Installation and Upgrade Guide, Release 4.2(1)SP1(3)

Configuration Guides

Cisco Nexus 1000V High Availability and Redundancy Configuration Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V Interface Configuration Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V Layer 2 Switching Configuration Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V License Configuration Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V Network Segmentation Manager Configuration Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V Quality of Service Configuration Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V Security Configuration Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V System Management Configuration Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V VXLAN Configuration Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1010 Software Configuration Guide, Release 4.2(1)SP1(3)

Send document comments to nexus1k-docfeedback@cisco.com.

Programming Guide

Cisco Nexus 1000V XML API User Guide, Release 4.2(1)SV1(5.1)

Reference Guides

Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V MIB Quick Reference

Cisco Nexus 1010 Command Reference, Release 4.2(1)SP1(3)

Troubleshooting and Alerts

Cisco Nexus 1000V Troubleshooting Guide, Release 4.2(1)SV1(5.1)

Cisco Nexus 1000V Password Recovery Guide

Cisco NX-OS System Messages Reference

Virtual Security Gateway Documentation

Cisco Virtual Security Gateway for Nexus 1000V Series Switch

Virtual Network Management Center

Cisco Virtual Network Management Center

Network Analysis Module Documentation

Cisco Prime Network Analysis Module Software Documentation Guide, 5.1

Cisco Prime Network Analysis Module (NAM) for Nexus 1010 Installation and Configuration Guide, 5.1

Cisco Prime Network Analysis Module Command Reference Guide 5.1

Cisco Prime Network Analysis Module Software 5.1 Release Notes

Cisco Prime Network Analysis Module Software 5.1 User Guide

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

CHAPTER 1

Overview

This chapter provides an overview of the interface types supported in Cisco Nexus 1000V.

This chapter includes the following sections:

Information About Interfaces

This section includes the following topics:

Ethernet Interfaces

All interfaces on the Cisco Nexus 1000V are Layer 2 Ethernet interfaces, which include access ports, trunk ports, private VLAN, and promiscuous ports.

This section includes the following topics:

Access Ports

An access port carries traffic for one VLAN. This type of port is a Layer 2 interface only. For more information about access-port interfaces, see Chapter 3, “Configuring Layer 2 Interfaces.”

Trunk Ports

A trunk port carries traffic for two or more VLANs. This type of port is a Layer 2 interface only. For more information about trunk-port interfaces, see Chapter 3, "Configuring Layer 2 Interfaces."

Private VLAN Ports

Private VLANs (PVLANs) are used to segregate Layer 2 ISP traffic and convey it to a single router interface. PVLANs achieve device isolation by applying Layer 2 forwarding constraints that allow end devices to share the same IP subnet while being Layer 2 isolated. In turn, the use of larger subnets reduces address management overhead. Three separate port designations are used. Each has its own unique set of rules that regulate the ability of each connected endpoint to communicate with other connected endpoints within the same private VLAN domain.

For more information about PVLANs, see the *Cisco Nexus 1000V Layer 2 Switching Configuration Guide, Release 4.2(1)SV1(5.1)*.

Promiscuous Ports

A promiscuous port can talk to all other types of ports. A promiscuous port can talk to isolated ports as well as community ports, and those ports can also talk to promiscuous ports.

For more information about promiscuous ports, see the *Cisco Nexus 1000V Layer 2 Switching Configuration Guide, Release 4.2(1)SV1(5.1)*

Virtual Ethernet Interfaces

Virtual Ethernet (vEthernet or vEth) interfaces are logical interfaces. Each vEthernet interface corresponds to a switch interface that is connected to a virtual port. The interface types are as follows:

- VM (interfaces connected to VM NICs)
- Service console
- vmkernel

vEthernet interfaces are created on the Cisco Nexus 1000V to represent virtual ports in use on the distributed virtual switch.

Management Interface

You can use the management Ethernet interface to connect the device to a network for remote management using a Telnet client, the Simple Network Management Protocol (SNMP), or other management agents. For more information on the management interface, see the *Cisco Nexus 1000V Getting Started Guide, Release 4.2(1)SV1(5.1)*.

Port Channel Interfaces

A port channel is a logical interface that aggregates multiple physical interfaces. You can bundle up to eight individual links to physical ports into a port channel to improve bandwidth and redundancy. You can also use port channeling to load balance traffic across these channeled physical interfaces. For more information about port channel interfaces, see Chapter 5, "Configuring Port Channels."

VEM Management of LACP

You can offload operation of the LACP protocol from the VSM to the VEMs. This prevents a situation where LACP cannot be negotiated with the upstream switch when the VEM is disconnected from the VSM (referred to as headless mode). VEM management of LACP allows the re-establishment of port channels after the reboot of a headless VEM.

Simplifying Interface Configuration with Port Profiles

A port profile is a mechanism for simplifying interface configuration. You can configure a port profile, and then assign it to multiple interfaces to give them all the same configuration. Changes to the port profile are propagated to the configuration of any interface that is assigned to it.

Note We do not recommend that you override port profile configurations by making changes to the assigned interface configurations. Only make configuration changes to interfaces to quickly test a change or to disable a port.

For more information about port profiles, see the *Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(5.1)*.

High Availability for Interfaces

Interfaces support stateful and stateless restarts. A stateful restart occurs during a supervisor switchover. After the switchover, Cisco Nexus 1000V applies the runtime configuration.

CHAPTER 2

Configuring Interface Parameters

This chapter describes how to configure the basic interface parameters or the parameters that are shared by multiple interfaces.

This chapter includes the following sections:

Note To configure Layer 2 access or trunking interfaces, see Chapter 2, "Configuring Interface Parameters."

Information About the Basic Interface Parameters

This section includes the following topics:

Description Parameter

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

By default, the description for vEthernet interfaces is auto-formatted to contain information about the device connected. The description for a VNIC, for example, contains the VM name and network adapter number. You keep this default description or can also override it with a description of your choosing.

For information about setting the description parameter for port channel interfaces, see the "Adding a Description to a Port Channel Interface" section on page 5-36.

For information about configuring this parameter for other interfaces, see the "Configuring a Description" section on page 2-5.

Speed and Duplex Modes

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

For information about setting the speed and duplex modes for port channel interfaces, see the "Configuring the Speed and Duplex Settings for a Port Channel Interface" section on page 5-37.

For information about setting the speed and duplex modes for other interfaces, see the "Configuring the Interface Speed and Duplex Modes" section on page 2-6.

Port MTU Size

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

By default, The MTU size for each port is 1500 bytes, which is the IEEE 802.3 standard for Ethernet frames. Larger MTU sizes are possible for more efficient processing of data with less overhead. The larger frames, called jumbo frames, can be up to 9000 bytes in size, which is also the fixed system jumbo MTU size in the Cisco Nexus 1000V.

For a Layer 2 port, you can configure an MTU size as the system default of 1500 bytes or the system default jumbo MTU size of 9000 bytes.

For information about setting the MTU size, see the "Configuring the MTU Size for an Ethernet Interface" section on page 2-8.

Administrative Status

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

For more information, see the following sections:

- Shutting Down and Restarting a Port Channel Interface, page 5-35.

- Shutting Down and Activating an Interface, page 2-10.

Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a Layer 2 protocol that enables two devices that run CDP to learn about each other. You can use CDP to troubleshoot the network by displaying information about the neighboring devices that are linked through each interface. By default, CDP is enabled.

To configure CDP, see the "Enabling or Disabling CDP" section on page 2-11.

Port Channel Parameter

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

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

To configure port channels, see the "Configuring Port Channels" section on page 5-1.

Guidelines and Limitations

Interface parameters have the following guidelines and limitations:

- You usually configure Ethernet port speed and duplex mode parameters to auto to allow negotiation of the speed and duplex modes between ports. If you decide to configure the port speed and duplex modes manually for these ports, consider the following:
  - If you set the Ethernet port speed to auto, the device automatically sets the duplex mode to auto.
  - If you enter the **no speed** command, the device automatically sets both the speed and duplex parameters to auto (the **no speed** command produces the same results as the **speed auto** command).
  - If you configure an Ethernet port speed to a value other than auto (for example, 10, 100, or 1000 Mbps), you must configure the connecting port to match. Do not configure the connecting port to negotiate the speed.

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

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

- To specify an interface in the CLI, use the following guidelines:
  - For an Ethernet port— use **ethernet** *slot/port*, where *slot* is the module slot number and *port* is the port number.

  - For the management interface—use **mgmt 0** or **mgmt0**.
  - For a vEthernet port— use **vethernet** *number*, where *number* is a number from 1 to 1048575.
  - A space is not required between the interface type and the slot/port or interface number. For example, for the Ethernet slot 4, port 5 interface, you can specify either of the following:
    ethernet 4/5
    ethernet4/5

- Jumbo frames are only supported on the vmxnet3 driver. Attempts to change the MTU appear to succeed but the adapter always drops frames larger than 1500 bytes.For more information see the VMware KB article 1015556.

Configuring the Basic Interface Parameters

This section includes the following topics:

Specifying an Interface to Configure

You can use this procedure to specify an interface to configure.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **config t**
2. **interface** *interface*
3. **show interface** *interface*

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters global configuration mode.
Step 2	**interface** *interface* **Example:** n1000v(config)# interface ethernet 2/1 n1000v(config-if)#	Enters interface configuration mode for the specified interface.
Step 3	**show interface** *interface* **Example:** n1000v(config-if)# **show interface ethernet 2/1**	Displays the current configuration of interfaces. The *interface* argument is defined as follows: • For an Ethernet port, use **ethernet** *slot/port*, where *slot* is the module slot number and *port* is the port number. • For the management interface, use **mgmt 0** or **mgmt0**. • For a vEthernet port, use **vethernet** *number*, where *number* is a number from 1 to 1048575.

Configuring a Description

You can use this procedure to add a description to av Ethernet, vEthernet, or management interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- A description is case-sensitive and can be up to 80 alphanumeric characters in length.

SUMMARY STEPS

1. **config t**
2. **interface** *interface*
3. **description** *string*
4. **show interface** *interface*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters global configuration mode.
Step 2	**interface** *interface* **Example:** n1000v(config)# interface ethernet 2/1 n1000v(config-if)#	Enters interface configuration mode for the specified interface.
Step 3	**description** *string* **Example:** n1000v(config-if)# description Ethernet port 3 on module 1. n1000v(config-if)#	Adds a description of up to 80 alphanumeric characters for the interface and saves it in the running configuration.
Step 4	**show interface** *interface* **Example:** n1000v(config)# show interface ethernet 2/1	Displays the interface status, which includes the description.
Step 5	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

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

```
n1000v# config t
n1000v(config)# interface ethernet 3/24
n1000v(config-if)# description server1
n1000v(config-if)#
```

Configuring the Interface Speed and Duplex Modes

You can use this procedure to configure the interface speed and duplex modes.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- The interface speed and duplex modes are interrelated, so you should configure both at the same time. To see the speeds and duplex modes that you can configure together for Ethernet and management interfaces, see the "Speed and Duplex Modes" section on page 2-2.

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

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

SUMMARY STEPS

1. **config t**
2. **interface** *interface*
3. **speed** {{**10** | **100** | **1000** | {**auto** [**10** **100** [**1000**]]}} | {**10000** | **auto**}}
4. **duplex** {**full** | **half** | **auto**}
5. **show interface** *interface*
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters the global configuration mode.

Step 2	**interface** *interface*
	
	Example:
	`n1000v(config)# interface ethernet 2/1`
	`n1000v(config-if)#`
		Enters interface configuration mode for the specified interface.

	Command	Purpose
Step 3	**speed** {{**10** | **100** | **1000** | {**auto** [**10** **100** [**1000**]]}} | {**10000** | **auto**}} **Example:** n1000v(config-if)# speed 1000 n1000v(config-if)#	Designates the port speed. • For Ethernet ports on the 48-port 10/100/1000 modules, sets the speed at 10 Mbps, 100 Mbps, or 1000 Mbps, or sets the port to auto negotiate its speed with the other 10/100/1000 port on the same link. • For Ethernet ports on the 32-port 10-Gigabit Ethernet modules, sets the speed at 10,000 Mbps (10 Gbps) or sets the port to autonegotiate its speed with the other 10-Gigabit Ethernet port on the link. • For management interfaces, sets the speed as 1000 Mbps or sets the port to autonegotiate its speed.
Step 4	**duplex** {**full** | **half** | **auto**} **Example:** n1000v(config-if)# duplex full	Specifies the duplex mode as full, half, or autonegotiate.
Step 5	**show interface** *interface* **Example:** n1000v(config)# show interface *mgmt0*	Displays the configuration
Step 6	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to set the speed of Ethernet port 1 on the 48-port 10/100/1000 module in slot 3 to 1000 Mbps and full-duplex mode:

```
n1000v# config t
n1000v(config)# interface ethernet 3/1
n1000v(config-if)# speed 1000
n1000v(config-if)# duplex full
n1000v(config-if)#
```

Configuring the MTU Size for an Ethernet Interface

You can use this procedure to configure the size of the maximum transmission unit (MTU) for a Layer 2 Ethernet interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You can specify an MTU size between 1500 and 9000 bytes for an Ethernet interface.

- Make sure the MTU value you set is supported by the VEM physical NIC. See your VMware documentation for more information about supported MTU for physical NICs.
- Jumbo frames are only supported on the vmxnet3 driver. Attempts to change the MTU appear to succeed but the adapter always drops frames larger than 1500 bytes.For more information see the VMware KB article 1015556.

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot/port*
3. **mtu** *size*
4. **show interface ethernet** *slot/port*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	
	Example:
	`n1000v(config)# interface ethernet 3/1`
	`n1000v(config-if)#`
		Specifies an Ethernet interface to configure, and enters interface configuration mode.

Step 3	**mtu** *size* **Example:** `n1000v(config-if)# mtu 9000`	Specifies an MTU size between 1500 (the default) and 9000 bytes.
Step 4	**show interface ethernet** *slot/port* **Example:** `n1000v(config-if)# show interface type slot/port`	Displays the interface status, which includes the MTU size.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to configure the Ethernet interface 3/1 with the default MTU size of 1500 bytes:

```
n1000v# config t
n1000v(config)# interface ethernet 3/1
n1000v(config-if)# mtu 1500
n1000v(config-if)#
```

Shutting Down and Activating an Interface

You can use this procedure to shut down and restart Ethernet or management interfaces.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- When you shut down an interface, it becomes disabled and the output of monitoring commands show it as being down.
- To activate an interface that has been shut down, you must restart the device.

SUMMARY STEPS

1. **config t**
2. **interface** *interface*
3. **shutdown**
4. **show interface** *interface*
5. **no shutdown**
6. **show interface** *interface*
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**interface** *interface*
	
	Example 1:
	`n1000v(config)# interface ethernet 2/1`
	`n1000v(config-if)#`
		Specifies the interface that you are configuring.
		The *interface* argument is defined as follows:
		• For an Ethernet port, use **ethernet** *slot/port*, where *slot* is the module slot number and *port* is the port number.
		• For the management interface, use **mgmt 0** or **mgmt0**.

Step 3	**shutdown** **Example:** `n1000v(config-if)# shutdown`	Disables the interface in the running configuration.
Step 4	**show interface** *interface* **Example:** `n1000v(config-if)# show interface ethernet 2/1`	Displays the interface status, which includes the administrative status.

	Command	Purpose
Step 5	**no shutdown** **Example:** `n1000v(config-if)# no shutdown`	Reenables the interface in the running configuration.
Step 6	**show interface** *interface* **Example:** `n1000v(config-if)# show interface ethernet 2/1`	Displays the interface status, which includes the administrative status. The *interface* argument is defined as follows: • For an Ethernet port, use **ethernet** *slot/port*, where *slot* is the module slot number and *port* is the port number. • For the management interface, use **mgmt 0** or **mgmt0**.
Step 7	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

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

```
n1000v# config t
n1000v(config)# interface ethernet 3/1
n1000v(config-if)# shutdown
n1000v(config-if)# no shutdown
n1000v(config-if)#
```

Enabling or Disabling CDP

You can use this procedure to enable or disable the Cisco Discovery Protocol (CDP) for Ethernet and management interfaces.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Make sure that CDP is enabled at both ends of the link.

SUMMARY STEPS

1. **config t**
2. **interface** *interface*
3. **cdp enable**
   no cdp enable
4. **show cdp interface** *interface*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters global configuration mode.
Step 2	**interface** *interface* **Example 1:** n1000v(config)# interface ethernet 3/1 n1000v(config-if)#	Specifies the interface that you are configuring. The *interface* argument is defined as follows: • For an Ethernet port, use **ethernet** *slot/port*, where *slot* is the module slot number and *port* is the port number. • For the management interface, use **mgmt 0** or **mgmt0**.
Step 3	**cdp enable** **Example:** n1000v(config-if)# cdp enable	Enables CDP for the interface in the running configuration. To work, this parameter must be enabled for both interfaces on the same link.
	no cdp enable **Example:** n1000v(config-if)# no cdp enable	Disables CDP for the interface in the running configuration. As soon as you disable CDP for one of two interfaces, CDP is disabled for the link.
Step 4	**show cdp interface** *interface* **Example:** n1000v(config-if)# show cdp interface *interface*	Displays the CDP status for the interface in the running configuration. The *interface* argument is defined as follows: • For an Ethernet port, use **ethernet** *slot/port*, where *slot* is the module slot number and *port* is the port number. • For the management interface, use **mgmt 0** or **mgmt0**.
Step 5	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to enable CDP for Ethernet port 3/1:

```
n1000v# config t
n1000v(config)# interface ethernet 3/1
n1000v(config-if)# cdp enable
n1000v(config-if)#
```

The following example shows how to disable CDP for Ethernet port 3/1:

```
n1000v# config t
n1000v(config)# interface ethernet 3/1
n1000v(config-if)# no cdp enable
n1000v(config-if)#
```

Clearing the Interface Counters

You can use this procedure to clear the Ethernet, vEthernet, and management interface counters.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode, configuration mode, or interface configuration mode.

SUMMARY STEPS

1. **clear counters** *interface-type interface-id*
2. **show interface** *interface*

DETAILED STEPS

	Command	Purpose

Step 1	**clear counters** *interface*
	
	Example:
	`n1000v# clear counters ethernet 2/1`
	`n1000v#`
		Clears the counters for the specified interface:
		• **ethernet** *slot/port*
		• **vethernet** *number*
		• **mgmt 0** or **mgmt0**

Step 2	**show interface** *interface* **Example:** `n1000v# show interface ethernet 2/1`	Displays the interface status, which includes the counters, for the specified interface: • **ethernet** *slot/port* • **vethernet** *number* • **mgmt 0** or **mgmt0**

EXAMPLES

The following example shows how to clear and reset the counters on Ethernet port 5/5:

```
n1000v# clear counters ethernet 5/5
n1000v#
```

Verifying the Basic Interface Parameters

Use the commands listed here to display and verify the basic interface parameters.

Command	Purpose
show cdp	Displays the CDP status.
show interface *interface*	Displays the configured states of one or all interfaces.
show interface brief	Displays a table of interface states.
show interface switchport	Displays the status of Layer 2 ports.

Feature History for Basic Interface Parameters

This section provides the feature history for basic interface parameters.

Feature Name	Releases	Feature Information
System jumbo MTU	4.2(1)SV1(4)	The system jumbo MTU is fixed at 9000 and cannot be changed.
Interface MTU	4.2(1)SV1(4)	The interface MTU can be configured as a value between 1500 and 9000.
Basic interface parameters	4.0(4)SV1(1)	This feature was introduced.

CHAPTER 3

Configuring Layer 2 Interfaces

This chapter describes how to configure Layer 2 switching ports as access or trunk ports.

This chapter includes the following sections:

Note For information about configuring a Switched Port Analyzer (SPAN) destination interface, see the *Cisco Nexus 1000V System Management Configuration Guide, Release 4.2(1)SV1(5.1)*.

Note for information about VLANs, MAC address tables, and private VLANs, see the *Cisco Nexus 1000V Layer 2 Switching Configuration Guide, Release 4.2(1)SV1(5.1)*.

Note for information about configuring vEthernet interfaces, see the "Configuring Virtual Ethernet Interfaces" section on page 4-1.

Information About Access and Trunk Interfaces

This section includes the following topics:

- High Availability, page 3-3

Access and Trunk Interfaces

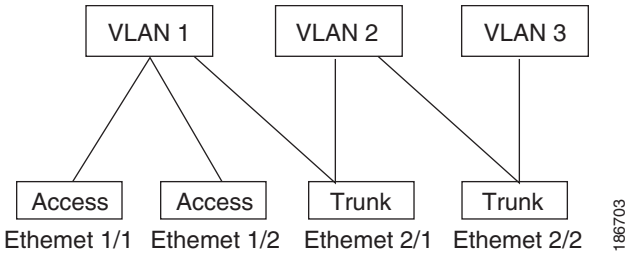
A Layer 2 port can be configured as an access or a trunk port as follows:

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

By default, all ports on the Cisco Nexus 1000V are Layer 2 ports. You can change the default port mode (access or trunk). See the *Cisco Nexus 1000V Getting Started Guide, Release 4.2(1)SV1(5.1)* for information about setting the default port mode.

Figure 3-1 shows how you can use trunk ports in the network. The trunk port carries traffic for two or more VLANs.

Figure 3-1 Trunk and Access Ports and VLAN Traffic



In order to correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation, or tagging, method (see the "IEEE 802.1Q Encapsulation" section on page 3-2 for more information).

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

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

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

IEEE 802.1Q Encapsulation

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

To correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation, or tagging, method that uses a tag that is inserted into the frame header (see Figure 3-2 and Figure 3-3). This tag carries information about the specific VLAN to which the frame and packet belong. This method allows packets that are encapsulated for several different VLANs to traverse the same port and maintain traffic separation between the VLANs. Also, the encapsulated VLAN tag allows the trunk to move traffic end to end through the network on the same VLAN.

Figure 3-2 Header Without 802.1Q Tag

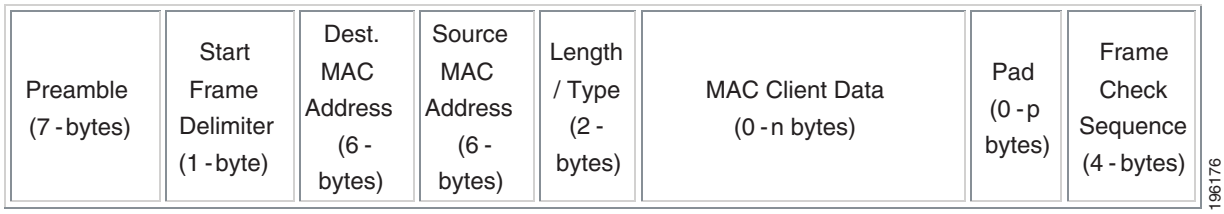
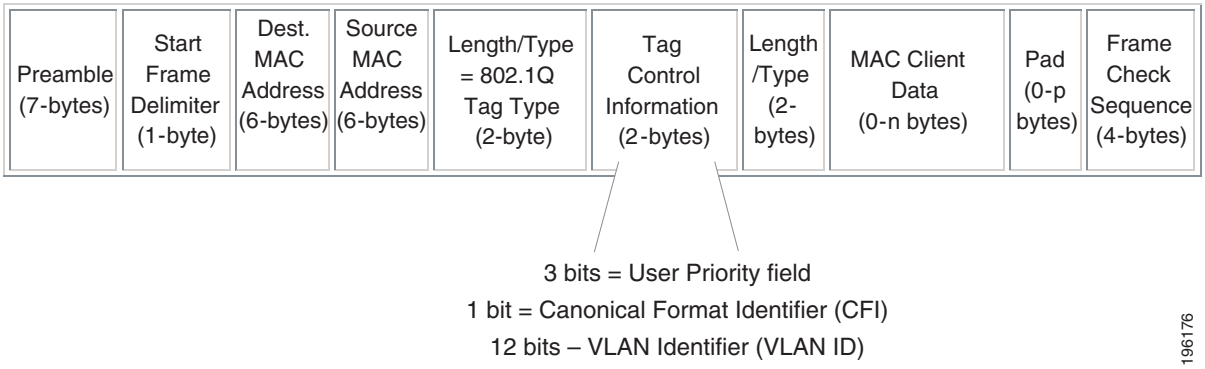


Figure 3-3 Header With 802.1Q Tag



High Availability

The software supports high availability for Layer 2 ports.

Prerequisites for VLAN Trunking

VLAN trunking has this prerequisite:

- You are logged into the CLI.

Guidelines and Limitations

VLAN trunking has the following guidelines and limitations:

- Do not connect devices with access links because access links may partition a VLAN.
- When connecting Cisco switches through an 802.1Q trunk, make sure that the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning tree loops might result.
- You can group trunk ports into port channel groups, but all trunks in the group must have the same configuration. When a group is first created, all ports follow the parameters set for the first port to be added to the group. If you change the configuration of one of these parameters, the device propagates that setting to all ports in the group, such as the allowed VLANs and the trunk status. For example, if one port in a port group ceases to be a trunk, all ports cease to be trunks.

- If you try to enable 802.1X on a trunk port, an error message appears, and 802.1X is not enabled.
- If you try to change the mode of an 802.1X-enabled port to trunk, the port mode is not changed.

Default Settings

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

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

Configuring Access and Trunk Interfaces

This section includes the following topics:

Note Be aware that the Cisco Nexus 1000V commands may differ from the Cisco IOS commands.

Configuring a LAN Interface as a Layer 2 Access Port

You can use this procedure to configure a Layer 2 port as an access port.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- The interface can be either Ethernet or vEthernet.
- An access port transmits packets on only one, untagged VLAN. You specify which VLAN traffic that the interface carries, which becomes the access VLAN. If you do not specify a VLAN for an access port, that interface carries traffic only on the default VLAN. The default VLAN is VLAN1.
- The VLAN must exist before you can specify that VLAN as an access VLAN. The system shuts down an access port that is assigned to an access VLAN that does not exist.

SUMMARY STEPS

1. **config t**
2. **interface** *interface*
3. **switchport mode access**
4. **switchport access vlan** *vlan-id*
5. **show interface**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters the global configuration mode.

Step 2	**interface** *interface*
	
	Example 1:
	`n1000v(config)# interface ethernet 3/1`
	`n1000v(config-if)#`
		Specifies the interface that you are configuring and places you in interface configuration mode.
		• For an Ethernet port, use **ethernet** *slot/port*, where *slot* is the module slot number and *port* is the port number.
		• For a vEthernet port, use **vethernet** *interface-number*, where *interface-number* is a number from 1 to 1048575.

Step 3	**switchport mode access** **Example:** `n1000v(config-if)# switchport mode access`	Sets the interface as a nontrunking nontagged, single-VLAN Layer 2 interface in the running configuration.
Step 4	**switchport access vlan** *vlan-id* **Example:** `n1000v(config-if)# switchport access vlan 5`	(Optional) Specifies the VLAN for which this access port will carry traffic and saves the change in the running configuration. If you do not enter this command, the access port carries traffic on VLAN1 only; use this command to change the VLAN for which the access port carries traffic.
Step 5	**show interface** **Example:** `n1000v(config)# show interface`	(Optional) Displays the interface status and information.
Step 6	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

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

```
n1000v# config t
n1000v(config)# interface ethernet 3/1
n1000v(config-if)# switchport mode access
n1000v(config-if)# switchport access vlan 5
n1000v(config-if)#
```

Configuring Trunk Ports

You can use this procedure to configure a Layer 2 port as a trunk port.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- Before you configure a trunk port, ensure that you are configuring a Layer 2 interface.
- The interface can be either Ethernet or vEthernet.
- A trunk port transmits untagged packets for one VLAN plus encapsulated, tagged, packets for multiple VLANs. (See the "IEEE 802.1Q Encapsulation" section on page 3-2 for information about encapsulation.)
- The device supports 802.1Q encapsulation only.

SUMMARY STEPS

1. **config t**
2. **interface** *interface*
3. **switchport mode trunk**
4. **show interface**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters the global configuration mode.

Step 2	**interface** *interface*
	
	Example:
	`n1000v(config)# interface ethernet 3/1`
	`n1000v(config-if)#`
		Specifies the interface that you are configuring and places you in interface configuration mode.
		• For an Ethernet port, use **ethernet** *slot/port*, where *slot* is the module slot number and *port* is the port number.
		• For a vEthernet port, use **vethernet** *interface-number*, where *interface-number* is a number from 1 to 1048575.

	Command	Purpose
Step 3	**switchport mode trunk** **Example:** n1000v(config-if)# switchport mode trunk	Sets the interface as a Layer 2 trunk port in the running configuration. A trunk port can carry traffic in one or more VLANs on the same physical link (VLANs are based on the trunk-allowed VLANs list). By default, a trunk interface can carry traffic for all VLANs. To specify that only certain VLANs are allowed on the specified trunk, use the **switchport trunk allowed vlan** command.
Step 4	**show interface** **Example:** n1000v(config)# show interface	(Optional) Displays the interface status and information.
Step 5	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

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

```
n1000v# config t
n1000v(config)# interface ethernet 3/1
n1000v(config-if)# switchport mode trunk
n1000v(config-if)#
```

Configuring the Native VLAN for 802.1Q Trunking Ports

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

SUMMARY STEPS

1. **config t**
2. **interface** *interface*
3. **switchport trunk native vlan** *vlan-id*
4. **show vlan**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters the global configuration mode.
Step 2	**interface** *interface* **Example:** n1000v(config)# interface ethernet 3/1 n1000v(config-if)#	Specifies the interface that you are configuring and places you in interface configuration mode. • For an Ethernet port, use **ethernet** *slot/port*, where *slot* is the module slot number and *port* is the port number. • For a vEthernet port, use **vethernet** *interface-number*, where *interface-number* is a number from 1 to 1048575.
Step 3	**switchport trunk native vlan** *vlan-id* **Example:** n1000v(config-if)# switchport trunk native vlan 5	Designates the native VLAN for the 802.1Q trunk in the running configuration. Valid values are from 1 to 4094, except those VLANs reserved for internal use. The default value is VLAN1.
Step 4	**show vlan** **Example:** n1000v(config)# show vlan	(Optional) Displays the status and information of VLANs.
Step 5	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

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

```
n1000v# config t
n1000v(config)# interface ethernet 3/1
n1000v(config-if)# switchport trunk native vlan 5
n1000v(config-if)#
```

Configuring the Allowed VLANs for Trunking Ports

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

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- 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.

SUMMARY STEPS

1. **config t**
2. **interface** *interface*
3. **switchport trunk allowed vlan** {*vlan-list* | **all** | **none** | [**add** | **except** | | **remove** {*vlan-list*}]}
4. **show vlan**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters the global configuration mode.

Step 2	**interface** *interface*
	
	Example:
	`n1000v(config)# interface ethernet 3/1`
	`n1000v(config-if)#`
		Specifies the interface that you are configuring and places you in interface configuration mode.
		• For an Ethernet port, use **ethernet** *slot/port*, where *slot* is the module slot number and *port* is the port number.
		• For a vEthernet port, use **vethernet** *interface-number*, where *interface-number* is a number from 1 to 1048575.

Step 3	**switchport trunk allowed vlan** {*vlan-list* **all** | **none** [**add** |**except** | **none** | **remove** {*vlan-list*}]} **Example:** `n1000v(config-if)# switchport trunk allowed vlan add 15-20#`	Sets the allowed VLANs for the trunk interface in the running configuration. The default is to allow all VLANs on the trunk interface. The range is from 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.
Step 4	**show vlan** **Example:** `n1000v# show vlan`	(Optional) Displays the status and information for VLANs.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

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

```
n1000v# config t
n1000v(config)# interface ethernet 3/1
n1000v(config-if)# switchport trunk allowed vlan 15-20
n1000v(config-if)#
```

Configuring the Device to Tag Native VLAN Traffic

When working with 802.1Q trunked interfaces, you can maintain the tagging for all packets that enter with a tag that matches the native VLAN ID. Untagged traffic is dropped (you will still carry control traffic on that interface).

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- The **vlan dot1q tag native** global command changes the behavior of all native VLAN ID interfaces on all trunks on the device.
- This feature applies to the entire device; you cannot apply it to selected VLANs on a device.

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

SUMMARY STEPS

1. **config t**
2. **vlan dot1q tag native**
3. **show vlan**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters the global configuration mode.

Step 2	**vlan dot1q tag native** **Example:** `n1000v(config)# vlan dot1q tag native`	Modifies the behavior of a 802.1Q trunked native VLAN ID interface in the running configuration. The interface maintains the taggings for all packets that enter with a tag that matches the value of the native VLAN ID and drops all untagged traffic. The control traffic is still carried on the native VLAN. The default is disabled.

	Command	Purpose
Step 3	**show vlan** **Example:** n1000v# show vlan	(Optional) Displays the status and information for VLANs.
Step 4	**copy running-config startup-config** **Example:** n1000v# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

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

```
n1000v# config t
n1000v(config)# vlan dot1q tag native
n1000v#
```

Verifying the Interface Configuration

You can display access and trunk interface configuration information.

Command	Purpose
show interface ethernet *slot/port* [**brief** | **capabilities** | **counters** | **mac-address** | **status** | **switchport** | **trunk**]	Displays the interface configuration
show interface ethernet *slot/port* **counters** [**brief** | **detailed** | **errors** | **snmp** | **storm-control** | **trunk**]	Displays the counters for a specified Ethernet interface.
show interface ethernet *slot/port* **status** [**err-disable**]	Displays the status for a specified Ethernet interface.
show interface brief	Displays interface configuration information, including the mode.
show interface switchport	Displays information, including access and trunk interface, information for all Layer 2 interfaces.
show interface trunk [**module** *module-number* | **vlan** *vlan-id*]	Displays trunk configuration information.
show interface capabilities	Displays information on the capabilities of the interfaces.
show running-config interface ethernet *slot/port*	Displays configuration information about the specified interface.

Monitoring the Interface Configuration

You can display access and trunk interface configuration information.

Command	Purpose
clear counters [*interface*]	Clears the counters.
show interface counters [**module** *module*]	Displays input and output octets unicast packets, multicast packets, and broadcast packets.
show interface counters detailed [**all**]	Displays input packets, bytes, and multicast as well as output packets and bytes.
show interface counters errors [**module** *module*]	Displays information on the number of error packets.

Configuration Examples for Access and Trunk Port Mode

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

```
n1000v# configure terminal
n1000v(config)# interface ethernet 2/30
n1000v(config-if)# switchport
n1000v(config-if)# switchport mode access
n1000v(config-if)# switchport access vlan 5
n1000v(config-if)#
```

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

```
n1000v# configure terminal
n1000v(config)# interface ethernet 2/35
n1000v(config-if)# switchport
n1000v(config-if)# switchport mode trunk
n1000v(config-if)# switchport trunk native vlan 10
n1000v(config-if)# switchport trunk allowed vlan 5, 10
n1000v(config-if)# exit
n1000v(config)# vlan dot1q tag native
n1000v(config)#
```

Additional References

For additional information related to implementing access and trunk port modes, see the following sections:

- Related Documents, page 3-13
- Standards, page 3-13

Related Documents

Related Topic	Document Title
Complete command syntax, command modes, command history, defaults, usage guidelines, and examples for all Cisco Nexus 1000V commands.	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(5.1)*
Port channels	Chapter 5, "Configuring Port Channels"
VLANs, private VLANs, and STP	*Cisco Nexus 1000V Layer 2 Switching Configuration Guide, Release 4.2(1)SV1(5.1)*
System management	*Cisco Nexus 1000V System Management Configuration Guide, Release 4.2(1)SV1(5.1)*
Release Notes	*Cisco Nexus 1000V Release Notes, Release 4.2(1)SV1(5.1)*

Standards

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

Feature History for Layer 2 Interface Parameters

This section provides the feature history for Layer 2 interface parameters.

Feature Name	Releases	Feature Information
Layer 2 interface parameters	4.0(4)SV1(1)	This feature was introduced.

CHAPTER 4

Configuring Virtual Ethernet Interfaces

This chapter describes how to configure virtual Ethernet (vEthernet or vEth) interfaces.

This chapter includes the following sections:

Information About vEthernet Interfaces

Virtual Ethernet (vEthernet or vEth) interfaces are logical interfaces. Each vEthernet interface corresponds to a switch interface that is connected to a virtual port. The interface types are as follows:

- VM (interfaces connected to VM NICs)
- Service console
- vmkernel

vEthernet interfaces are created on the Cisco Nexus 1000V to represent virtual ports in use on the distributed virtual switch.

vEthernet interfaces are mapped to connected ports by MAC address as well as DVPort number. When a server administrator changes the port profile assignment on a vNIC or hypervisor port, the same vEthernet interface is reused. This is a change in Release 4.2(1)SV1(4). In previous releases, the VSM assigned a new vEthernet interface.

When bringing up a vEthernet interface where a change in the port profile assignment is detected, the VSM automatically purges any manual configuration present on the interface. You can use the following command to prevent purging of the manual configuration:

no svs veth auto-config-purge

Guidelines and Limitations

vEthernet interface configuration has the following guideline and limitation:

- MTU cannot be configured on a vEthernet interface.

Default Settings

The following table lists the default settings for vEthernet interface configuration.

Parameters	Default
Switchport mode	Access
Allowed VLANs	1 to 4094
Access VLAN ID	VLAN1
Native VLAN ID	VLAN1
Native VLAN ID tagging	Disabled
Administrative state	Shut
Automatic deletion of vEthernet interfaces	Enabled
Automatic purge of manual configuration on vEthernet interfaces	Enabled
Automatic creation of vEthernet interfaces	Enabled

Configuring vEthernet Interfaces

This section includes the following topics:

Configuring Global vEthernet Properties

You can use this procedure to enable or disable the following automatic controls for vEthernet interfaces:

- Deleting unused vEthernet interfaces
- Purging of manual vEthernet configurations
- Creating vEthernet interfaces

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **config t**
2. (Optional) [**no**] **svs veth auto-delete**
3. (Optional) [**no**] **svs veth auto-config-purge**
4. (Optional) [**no**] **svs veth auto-setup**
5. **show running-config all | grep "svs-veth"**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** n1000v# config t n1000v(config)#	Enters the global configuration mode.
Step 2	[**no**] **svs veth auto-delete** **Example:** n1000v(config)# svs veth auto-delete n1000v(config)#	(Optional) Enables the VSM to automatically delete DVPorts no longer used by a vNIC or hypervisor port. The default setting = enabled The no form of this command prevents the VSM from deleting unused DVPorts.
Step 3	[**no**] **svs veth auto-config-purge** **Example:** n1000v(config)# svs veth auto-config-purge n1000v(config)#	(Optional) Enables the VSM to remove all manual configuration on a vEthernet interface when the system administrator changes a port profile on the interface. The default setting = enabled The no form of this command prevents the manual configuration from being deleted in this situation. **Note** Port profiles with ephemeral bindings are purged regardless of this setting.
Step 4	[**no**] **svs veth auto-setup** **Example:** n1000v(config)# svs veth auto-setup n1000v(config)#	(Optional) Enables the VSM to automatically create a vEthernet interface when a new port is activated on a host. The no form of this command disables the automatic creation of vEthernet interfaces in this situation. **Note** You can use no form of the command to temporary block automatic creation of vEthernet interfaces.

	Command	Purpose
Step 5	**show running-config all | grep "svs-veth"** **Example:** n1000v(config)# show running-config all | grep "svs veth" svs veth auto-setup svs veth auto-delete svs veth auto-config-purge n1000v(config)# **Example:** n1000v(config)# show running-config all | grep "svs veth" n1000v(config)#	(Optional) Displays the default global vEthernet settings that are in effect on the VSM for verification. If a setting is disabled, it does not display in the show command output.
Step 6	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Configuring a vEthernet Access Interface

You can use this procedure to configure a vEthernet interface for use as an access interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged into the CLI in EXEC mode.
- If you do not add a description to the vEthernet interface, then one of the following descriptions is added at attach time. If you add a description and then remove it using the **no description** command, then one of the following descriptions is added to the interface:
  - For a VM—*VM-Name, Network Adapter number*
  - For a VMK—*VMware VMkernel, vmk number*
  - For a VSWIF—*VMware Service Console, vswif number*

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *interface-number*
3. (Optional) **description** *string*
4. **switchport access vlan** *vlan-id*
5. **switchport mode access**
6. **show interface** *interface-number*
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters the global configuration mode.

Step 2	**interface vethernet** *interface-number*
	
	Example:
	`n1000v(config)# interface vethernet 100`
	`n1000v(config-if)#`
		Enters the interface configuration mode for the specified vEthernet interface (from 1 to 1048575).

Step 3	**description** *string* **Example:** `n1000v(config-if)# description accessvlan`	(Optional) Adds a description of up to 80 alphanumeric characters to the interface in the running configuration. **Note** If you do not add a description, the default description is added. **Note** You do not need to use quotations around descriptions that include spaces.
Step 4	**switchport access vlan** *vlanid* **Example:** `n1000v(config-if)# switchport access vlan 5`	Configures the vEthernet interface as an access interface and specifies the VLAN ID (1 to 4094) in the running configuration.

Step 5	**switchport mode access**
	
	Example:
	`n1000v(config-if)# switchport mode access`
	`n1000v(config-if)#`
		Configures the vEthernet interface for use as an access interface in the running configuration.

Step 6	**show interface vethernet** *interface-number* **Example:** `n1000v(config-if)# show interface vethernet1`	(Optional) Displays the specified interface for verification.
Step 7	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Configuring a Private VLAN on a vEthernet Interface

You can use this procedure to configure a private VLAN (PVLAN) on a vEthernet interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged into the CLI in EXEC mode.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *interface-number*
3. (Optional) **description** *string*
4. **switchport access vlan** *vlan-id*
5. **switchport mode private-vlan host**
6. **switchport private-vlan host-association** *primary-vlan-id*
7. **show interface**
8. **copy running-config startup-config**

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters the global configuration mode.

Step 2	**interface vethernet** *interface-number*
	
	Example:
	`n1000v(config)# interface vethernet 1`
	`n1000v(config-if)#`
		Enters the interface configuration mode for the specified vEthernet interface (from 1 to 1048575).

Step 3	**description** *string* **Example:** `n1000v(config-if)# description isp_pvlan1`	(Optional) Adds a description of up to 80 alphanumeric characters to the interface in the running configuration. **Note** If you do not add a description, the default description is added. **Note** You do not need to use quotations around descriptions that include spaces.
Step 4	**switchport access vlan** *vlan-id* **Example:** `n1000v(config-if)# switchport access vlan 5`	Configures the vEthernet interface as an access interface and specifies the VLAN ID (from 1 to 4094) in the running configuration.
Step 5	**switchport mode private-vlan host** **Example:** `n1000v(config-if)# switchport mode private-vlan host`	Configures the vEthernet interface for a PVLAN host in the running configuration.
Step 6	**switchport private-vlan host-association** *primary-vlanid* **Example:** `n1000v(config-if)# switchport private-vlan host-association 5`	Configures the vEthernet interface for a host association with a specific primary VLAN ID (from 1 to 4094) in the running configuration.

	Command	Purpose
Step 7	**show interface** **Example:** n1000v# show interface	(Optional) Displays the interface status and information.
Step 8	**copy running-config startup-config** **Example:** n1000v(config)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to configure a vEthernet interface to use in a private vlan:

```
n1000v# config t
n1000v(config)# interface vethernet 1
n1000v(config-if)# switchport access vlan 5
n1000v(config-if)# switchport mode private-vlan host
n1000v(config-if)# switchport private-vlan host-association 5
n1000v(config-if)# show interface vethernet 1
Vethernet1 is up
    Port description is gentoo, Network Adapter 1
    Hardware is Virtual, address is 0050.5687.3bac
    Owner is VM "gentoo", adapter is Network Adapter 1
    Active on module 4
    VMware DVS port 1
    Port-Profile is vm
    Port mode is access
    5 minute input rate 1 bytes/second, 0 packets/second
    5 minute output rate 94 bytes/second, 1 packets/second
    Rx
    655 Input Packets 594 Unicast Packets
    0 Multicast Packets 61 Broadcast Packets
    114988 Bytes
    Tx
    98875 Output Packets 1759 Unicast Packets
    80410 Multicast Packets 16706 Broadcast Packets 0 Flood Packets
    6368452 Bytes
    0 Input Packet Drops 0 Output Packet Drops
```

Enabling or Disabling a vEthernet Interface

You can use this procedure to enable or disable a vEthernet interface.

SUMMARY STEPS

1. **config t**
2. **interface vethernet** *interface-number*
3. **[no] shutdown**
4. **show interface**
5. **copy running-config startup-config**

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged into the CLI in EXEC mode.

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`n1000v# config t`
	`n1000v(config)#`
		Enters the global configuration mode.

Step 2	**interface vethernet** *interface-number*
	
	Example:
	`n1000v(config)# interface vethernet 100`
	`n1000v(config-if)#`
		Enters the interface configuration mode for the specified vEthernet interface (from 1 to 1048575).

Step 3	[**no**] **shutdown**
	
	Example:
	`n1000v(config-if)# no shutdown`
	`n1000v(config-if)#`
		Enables or disables the vEthernet interface in the running configuration:
		• **shutdown**—Disables the vEthernet interface.
		• **no shutdown**—Enables the vEthernet interface.

Step 4	**show interface** **Example:** `n1000v# show interface`	(Optional) Displays the interface status and information.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to enable a vEthernet interface:

```
n1000v# config t
n1000v(config)# interface vethernet 100
n1000v(config)# no shutdown
n1000v(config-if)# show interface veth100 status

--------------------------------------------------------------------------------
Port           Name               Status    Vlan      Duplex  Speed   Type
--------------------------------------------------------------------------------
Veth100        --                       up          1       auto        auto    --
n1000v(config-if)#
```

Verifying the vEthernet Interface Configuration

You can use the following commands to display the vEthernet interface configuration:

Command	Purpose
show interface vethernet *interface-number* **[brief | counters [detailed [all] | errors] | description | mac-address | status [down | err-disabled | inactive | module** *num* **| up] | switchport]**	Displays the vEthernet interface configuration.
show interface [**vethernet** *interface-number*]	Displays the complete interface configuration.
show interface [**vethernet** *interface-number*] **brief**	Displays abbreviated interface configuration.
show interface [**vethernet** *interface-number*] **description**	Displays the interface description.
show interface [**vethernet** *interface-number*] **mac-address**	Displays the interface MAC address. **Note** For vEth interfaces this shows the MAC address of the connected device.
show interface [**vethernet** *interface-number*] **status** [**down** | **err-disabled** | **inactive** | **module** *num* | **up**]	Displays interface line status.
show interface [**vethernet** *interface-number*] **switchport**	Displays interface switchport information.
show interface virtual [**vm** [*vm_name*] | **vmk** | **vswif**] [**module** *mod_no*]	Displays virtual interfaces only.
show interface virtual port-mapping [**vm** [*name*] | **vmk** | **vswif** | **description**] [*module_num*]	Displays mappings between veth and VMware DVPort.

The following example shows how to display vEthernet 1:

```
n1000v# show interface veth1
Vethernet1 is up
    Port description is gentoo1, Network Adapter 1
    Hardware is Virtual, address is 0050.56bd.42f6
    Owner is VM "gentoo1", adapter is Network Adapter 1
    Active on module 33
    VMware DVS port 100
    Port-Profile is vlan48
    Port mode is access
    Rx
    491242 Input Packets 491180 Unicast Packets
    7 Multicast Packets 55 Broadcast Packets
    29488527 Bytes
    Tx
    504958 Output Packets 491181 Unicast Packets
    1 Multicast Packets 13776 Broadcast Packets 941 Flood Packets
    714925076 Bytes
    11 Input Packet Drops 0 Output Packet Drops
n1000v#
```

The following example shows how to display information for all vEthernet interfaces:

```
n1000v# show interface virtual

--------------------------------------------------------------------------------
Port        Adapter        Owner                    Mod Host
--------------------------------------------------------------------------------
```

```
Veth1                     Vm1-kl61                2
Veth2                     VM1-kl65                5
Veth3                     VM2-kl61                2
Veth1       Net Adapter 1 austen-gentoo1         33  austen-strider.austen.
Veth2       Net Adapter 2 austen-gentoo1         33  austen-strider.austen.
n1000v#
```

The following example shows how to display the descriptions for all vEthernet interfaces:

```
n1000v# show interface virtual description

--------------------------------------------------------------------------------
Interface   Description
--------------------------------------------------------------------------------
Veth1       gentoo1, Network Adapter 1
Veth2       gentoo1, Network Adapter 2
Veth3       VMware VMkernel, vmk1
Veth4       VMware Service Console, vswif1
```

The following example shows how to display the virtual port mapping for all vEthernet interfaces:

```
n1000v# show interface virtual port-mapping

--------------------------------------------------------------------------------
Port    Hypervisor Port    Binding Type    Status    Reason
--------------------------------------------------------------------------------
Veth1   DVPort5747         static          up        none
Veth2   DVPort3361         static          up        none
```

The following example shows how to display the running configuration information for all vEthernet interfaces:

```
n1000v# show running-config interface veth1
version 4.2(1)SV1(4)

interface Vethernet1
  inherit port-profile vlan48
  description gentoo1, Network Adapter 1
  vmware dvport 2968 dvswitch uuid "d4 02 20 50 16 4b 36 97-46 09 dc d8 5b c6 1e c1"
  vmware vm mac 0050.56A0.0000
```

Monitoring the vEthernet Interface Configuration

You can use the following commands to monitor the vEthernet interface configuration:

Command	Purpose
show interface [**vethernet** *interface-number*] **counters**	Displays the interface incoming and outgoing counters.
show interface [**vethernet** *interface-number*] **counters detailed** [**all**]	Displays detailed information for all counters. **Note** If 'all' is not specified then only non-zero counters are shown.
show interface [**vethernet** *interface-number*] **counters errors**	Displays the interface error counters.

The following example shows how to display the counters for all vEthernet interfaces:

```
n1000v# show interface counters

--------------------------------------------------------------------------------
Port              InOctets    InUcastPkts    InMcastPkts    InBcastPkts
--------------------------------------------------------------------------------
mgmt0                42754             --              0             --
Eth2/2            41423421         112708         125997         180167
Eth5/2            39686276         119152          93284         180100
Eth5/6             4216279           9530          31268             40
Veth1                    0              0              0              0
Veth2                    0              0              0              0
Veth3                    0              0              0              0
Veth4                    0              0              0              0
Veth5                    0              0              0              0
Veth6                    0              0              0              0
Veth7                    0              0              0              0
Veth100                  0              0              0              0

--------------------------------------------------------------------------------
Port             OutOctets  OutUcastPkts  OutMcastPkts  OutBcastPkts
--------------------------------------------------------------------------------
mgmt0                 3358             --            --            --
Eth2/2            23964739         116150           516         52768
Eth5/2            26419473         111598           571         52420
Eth5/6             1042930           9548           536            14
Veth1               393589              0          6150             0
Veth2               393600              0          6150             0
Veth3               393600              0          6150             0
Veth4                    0              0             0             0
Veth5                    0              0             0             0
Veth6                    0              0             0             0
Veth7                    0              0             0             0
Veth100                  0              0             0             0

n1000v#
```

Configuration Examples for vEthernet Interfaces

The following example shows how to configure a vEthernet access interface and assign the access VLAN for that interface:

```
n1000v# configure terminal
n1000v(config)# interface vethernet 100
n1000v(config-if)# switchport
n1000v(config-if)# switchport mode access
n1000v(config-if)# switchport access vlan 5
n1000v(config-if)#
```

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

```
n1000v# configure terminal
n1000v(config)# interface vethernet 1
n1000v(config-if)# switchport
n1000v(config-if)# switchport mode trunk
n1000v(config-if)# switchport trunk native vlan 10
n1000v(config-if)# switchport trunk allowed vlan 5, 10
n1000v(config-if)#
```

Additional References

For additional information related to implementing access and trunk port modes, see the following sections:

- Related Documents, page 4-12
- Standards, page 4-12

Related Documents

Related Topic	Document Title
Complete command syntax, command modes, command history, defaults, usage guidelines, and examples for all Cisco Nexus 1000V commands.	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(5.1)*
Port Profiles	*Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(5.1)*
VLANs and private VLANs	*Cisco Nexus 1000V Layer 2 Switching Configuration Guide, Release 4.2(1)SV1(5.1)*
System management	*Cisco Nexus 1000V System Management Configuration Guide, Release 4.2(1)SV1(5.1)*
Release Notes	*Cisco Nexus 1000V Release Notes, Release 4.2(1)SV1(5.1)*

Standards

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

Feature History for vEthernet Interfaces

This section provides the feature history for vEthernet interfaces.

Feature Name	Releases	Feature Information
Global vEthernet interface controls	4.2(1)SV1(4)	You can enable or disable the following automatic vEthernet interface controls: • Deleting unused vEthernet interfaces • Purging of manual vEthernet configurations • Creating vEthernet interfaces
vEthernet interface parameters	4.0(4)SV1(1)	This feature was introduced.

CHAPTER 5

Configuring Port Channels

This chapter describes how to configure port channels and includes the following topics:

Information About Port Channels

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

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

This section includes the following topics:

- Network State Tracking for VPC-HM, page 5-11

Port Channels

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

You can bundle up to eight ports into a static port channel without using any aggregation protocol.

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

Each port can be in only one port channel. All the ports in a port channel must be compatible; they must use the same speed and duplex mode (see the "Compatibility Checks" section on page 5-2). When you run static port channels with no aggregation protocol, the physical links are all in the **on** channel mode.

You can create port channels directly by creating the port channel interface, or you can create a channel group that acts to aggregate individual ports into a bundle. When you associate an interface with a channel group, the software creates a matching port channel automatically if the port channel does not already exist. In this instance, the port channel assumes the Layer 2 configuration of the first interface. You can also create the port channel first. In this instance, the Cisco Nexus 1000V creates an empty channel group with the same channel number as the port channel and takes the default Layer 2 configuration, as well as the compatibility configuration (see the "Compatibility Checks" section on page 5-2).

Note The port channel is operationally up when at least one of the member ports is up and is in the channeling state. The port channel is operationally down when all member ports are operationally down.

Compatibility Checks

When you add an interface to a port channel group, the following compatibility checks are made before allowing the interface to participate in the port channel:

- Network layer
- (Link) speed capability
- Speed configuration
- Duplex capability
- Duplex configuration
- Port mode
- Access VLAN
- Trunk native VLAN
- Tagged or untagged
- Allowed VLAN list
- MTU size

- SPAN—cannot be a SPAN source or a destination port
- Storm control

To view the full list of compatibility checks performed by the Cisco Nexus 1000V, use the **show port-channel compatibility-parameters**.

You can only add interfaces configured with the channel mode set to **on** to static port channels. You can configure these attributes on an individual member port. If you configure a member port with an incompatible attribute, the Cisco Nexus 1000V suspends that port in the port channel.

Alternatively, you can force ports with incompatible parameters to join the port channel if the following parameters are the same:

- (Link) speed capability
- Speed configuration
- Duplex capability
- Duplex configuration

When the interface joins a port channel, some of its individual parameters are removed and replaced with the values on the port channel as follows:

- Bandwidth
- Delay
- Extended Authentication Protocol over UDP
- VRF
- IP address (v4 and v6)
- MAC address
- Spanning Tree Protocol
- NAC
- Service policy
- Quality of Service (QoS)
- Access control lists (ACLs)

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

- Description
- CDP
- MDIX
- Rate mode
- Shutdown
- SNMP trap

Note When you delete the port channel, the software sets all member interfaces as if they were removed from the port channel.

Load Balancing Using Port Channels

The Cisco Nexus 1000V load balances traffic across all operational interfaces in a port channel by hashing the addresses in the frame to a numerical value that selects one of the links in the channel. Port channels provide load balancing by default. Port channel load balancing uses MAC addresses, IP addresses, or Layer 4 port numbers to select the link. Port channel load balancing uses either source or destination addresses or ports, or both source and destination addresses or ports.

You can configure the load balancing mode to apply to all port channels that are configured on the entire device or on specified modules. The per-module configuration takes precedence over the load-balancing configuration for the entire device. You can configure one load balancing mode for the entire device, a different mode for specified modules, and another mode for the other specified modules. You cannot configure the load balancing method per port channel.

You can configure the type of load balancing algorithm used. You can choose the load balancing algorithm that determines which member port to select for egress traffic by looking at the fields in the frame.

Note The default load balancing method uses source MAC addresses.

You can configure one of the following methods to load balance across the port channel:

- Destination MAC address
- Source MAC address
- Source and Destination MAC address
- Destination IP address and VLAN
- Source IP address and VLAN
- Source and destination IP address and VLAN
- Destination TCP/UDP port number
- Source TCP/UDP port number
- Source and destination TCP/UDP port number
- Destination IP address and TCP/UDP port number
- Source IP address and TCP/UDP port number
- Source and destination IP address and TCP/UDP port number
- Destination IP address, TCP/UDP port number, and VLAN
- Source IP address, TCP/UDP port number, and VLAN
- Source and destination IP address, TCP/UDP port number, and VLAN
- Destination IP address
- Source IP address
- Source and Destination IP address
- VLAN only
- Source Virtual Port ID

When you configure source IP address load balancing, the source MAC address is used to balance the traffic load. When you configure the destination MAC address load balancing method, the traffic load is balanced using the destination MAC address.

The load balancing methods that use port channels do not apply to multicast traffic. Regardless of the method configured, multicast traffic uses the following methods for load balancing with port channels:

- Multicast traffic with Layer 4 information—Source IP address, source port, destination IP address, and destination port
- Multicast traffic without Layer 4 information—Source IP address and destination IP address
- Non-IP multicast traffic—Source MAC address and destination MAC address

To configure port channel load balancing, see the "Configuring Port Channel Load Balancing" procedure on page 5-38.

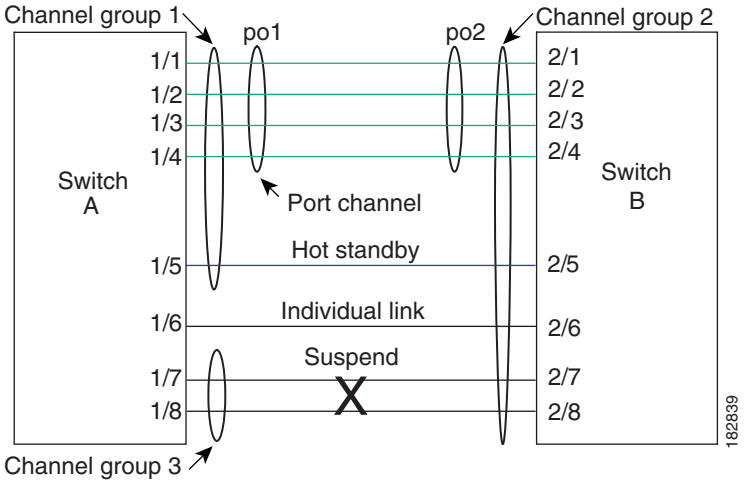
LACP

Link Aggregation Control Protocol (LACP) lets you configure up to 16 interfaces into a port channel. A maximum of eight interfaces can be active, and a maximum of eight interfaces can be placed in a standby state. Figure 5-1 shows how individual links can be combined into LACP port channels and channel groups as well as function as individual links.

Note When you delete the port channel, the associated channel group is automatically deleted. All member interfaces revert to their original configuration.

This section includes the following topics:

Figure 5-1 Individual Links Combined into a Port Channel



VEM Management of LACP

You can offload operation of the LACP protocol from the VSM to the VEMs. This prevents a situation where the VSM cannot negotiate LACP with the upstream switch when the VEM is disconnected from the VSM (referred to as headless mode). VEM management of LACP allows it to re-establish port channels after the reboot of a headless VEM.

Port Channel Modes

Individual interfaces in port channels are configured with channel modes. When you run static port channels with no aggregation protocol, the channel mode is always set to **on**.

You enable LACP for each channel by setting the channel mode for each interface to **active** or **passive**. You can configure either channel mode for individual links in the LACP channel group when you are adding the links to the channel group.

Table 5-1 describes the channel modes.

Table 5-1 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 are not running LACP) remain in this mode. If you attempt to change the channel mode to active or passive before enabling LACP, the device displays an error message. You enable LACP on each channel by configuring the interface in that channel for the channel mode as either **active** or **passive**. When an LACP attempts to negotiate with an interface in the **on** state, it does not receive any LACP packets and becomes an individual link with that interface; it does not join the LACP channel group. The default port channel mode is **on**.

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

Ports can form an LACP port channel when they are in different LACP modes if the modes are compatible as 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 and cannot form a port channel with another port that is in **active** or **passive** mode.

LACP ID Parameters

This section describes the LACP parameters in the following topics:

LACP System Priority

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

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

LACP Port Priority

Each port that is configured to use LACP has an LACP port priority. You can accept the default value of 32768 for the LACP port priority, or you can configure a value between 1 and 65535. LACP uses the port priority with the port number to form the port identifier.

LACP uses the port priority to decide which ports should be put in standby mode when there is a limitation that prevents all compatible ports from aggregating and which ports should be put into active mode. A higher port priority value means a lower priority for LACP. You can configure the port priority so that specified ports have a lower priority for LACP and are most likely to be chosen as active links, rather than hot-standby links.

LACP Administrative Key

LACP automatically configures an administrative key value that is equal to the channel-group number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port's ability to aggregate with other ports is determined by these factors:

- Port physical characteristics, such as the data rate and the duplex capability
- Configuration restrictions that you establish

LACP Marker Responders

You can dynamically redistribute the data traffic by using port channels. This redistribution may result from a removed or added link or a change in the load-balancing scheme. Traffic redistribution that occurs in the middle of a traffic flow can cause misordered frames.

LACP uses the Marker Protocol to ensure that frames are not duplicated or reordered due to this redistribution. The Marker Protocol detects when all the frames of a given traffic flow are successfully received at the remote end. LACP sends Marker PDUs on each of the port-channel links. The remote system responds to the Marker PDU once it receives all the frames received on this link prior to the Marker PDU. The remote system then sends a Marker Responder. Once the Marker Responders are received by the local system on all member links of the port channel, the local system can redistribute the frames in the traffic flow with no chance of misordering. The software supports only Marker Responders.

LACP-Enabled and Static Port Channels Differences

Table 5-2 summarizes the major differences between port channels with LACP enabled and static port channels.

Table 5-2 Port Channels with LACP Enabled and Static Port Channels

Configurations	Port Channels with LACP Enabled	Static Port Channels
Protocol applied	Enable globally	Not applicable
Channel mode of links	Can be either: • Active • Passive	Can only be On
Maximum number of links in channel	16	8

vPC Host Mode

vPC-HM is a way of creating a port channel when connecting to multiple upstream switches that are not clustered. In the Cisco Nexus 1000V, the port channel is divided into subgroups or logical smaller port channels, each representing one or more uplinks to one upstream physical switch.

Links that connect to the same physical switch are bundled in the same subgroup automatically by using information gathered from the Cisco Discovery Protocol packets from the upstream switch. Interfaces can also be manually assigned a specific subgroup. For more information, see the following procedures:

- Pinning a vEthernet Interface to a Subgroup, page 5-24 (configured on the port profile)
- Configuring Static Pinning for an Interface, page 5-32 (configured on the interface)

When vPC-HM is used, each vEthernet interface on the VEM is mapped to one of two subgroups in a round-robin method. All traffic from the vEthernet interface uses the assigned subgroup unless it is unavailable, in which case the vEthernet interface fails over to the remaining subgroup. When the original subgroup becomes available again, traffic shifts back to it. Traffic from each vEthernet interface is then balanced based on the configured hashing algorithm.

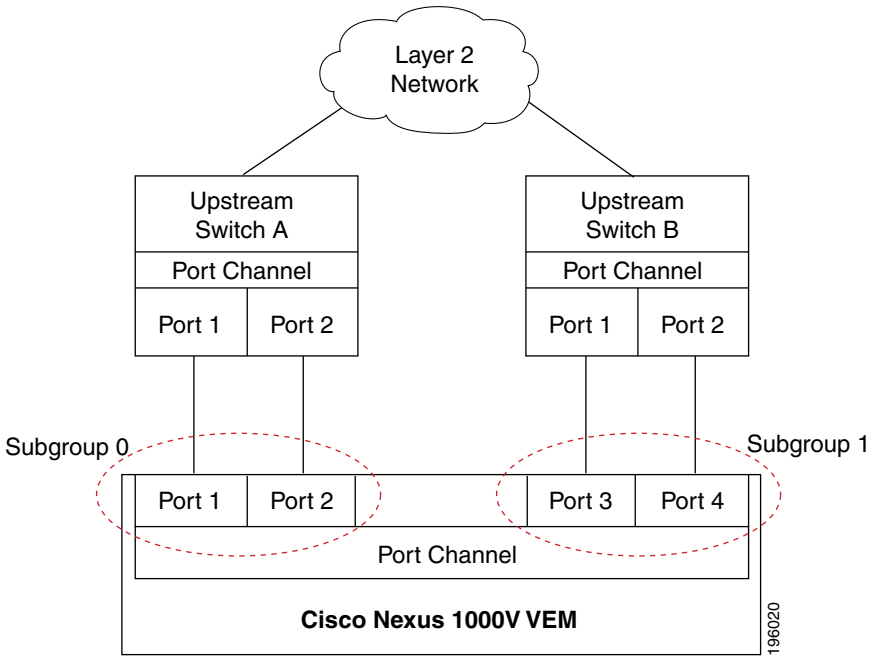
When multiple uplinks are attached to the same subgroup, the upstream switch must be configured in a port channel, the links bundled together. The port channel must also be configured with the **channel-group auto mode on** (active and passive modes use LACP).

If the upstream switches do not support port channels, you can use MAC pinning to assign each Ethernet port member to a particular port channel subgroup. For more information, see the "MAC Pinning" section on page 5-10.

Note Do not configure vPC-HM on the Cisco Nexus 1000V when the upstream switch ports that connect to the VEMs have vPC configured. In this case, the connection can be interrupted or disabled.

Figure 5-2 shows traffic separation using vPC-HM by assigning member ports 1 and 2 to subgroup ID 0 and member ports 3 and 4 to subgroup ID 1.

Figure 5-2 Using vPC-HM to Connect a Port Channel to Multiple Upstream Switches



To configure a port profile in vPC-HM, see the "Connecting to Multiple Upstream Switches" procedure on page 5-17.

Subgroup Creation

If Cisco Discovery Protocol (CDP) is enabled on the upstream switches, then subgroups are automatically created using information gathered from the Cisco Discovery Protocol packets. If not, then you must use the "Manually Configuring Interface Subgroups" procedure on page 5-22.

Static Pinning

Static pinning allows you to pin the virtual ports behind a VEM to a particular subgroup within the channel. Instead of allowing round robin dynamic assignment between the subgroups, you can assign (or pin) a static vEthernet interface, control VLAN, or packet VLAN to a specific port channel subgroup. With static pinning, traffic is forwarded only through the member ports in the specified subgroup.

You can use the following procedures to designate the subgroup to communicate with the network.

- "Pinning a vEthernet Interface to a Subgroup" section on page 5-24
- "Pinning a Control or Packet VLAN to a Subgroup" section on page 5-26

You can also pin vEthernet interfaces to subgroups in interface configuration mode using the "Configuring Static Pinning for an Interface" procedure on page 5-32.

MAC Pinning

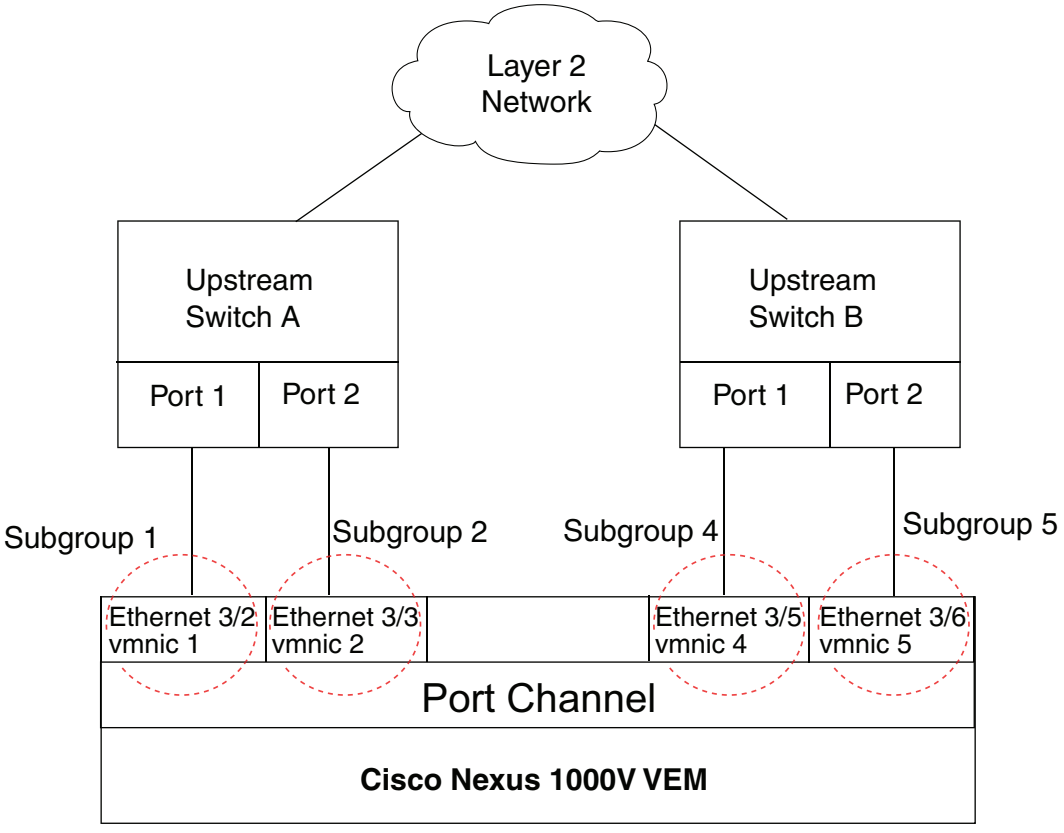
If you are connecting to multiple upstream switches that do not support port channels, then MAC pinning is the preferred configuration. MAC pinning divides the uplinks from your server into standalone links and pins the MAC addresses to those links in a round-robin method. This ensures that the MAC address of a virtual machine is never seen on multiple upstream switch interfaces. Therefore no upstream configuration is required to connect the VEM to upstream switches.

MAC pinning does not rely on any protocol to distinguish upstream switches so the configuration is independent of upstream hardware or design.

In case of a failure, the Cisco Nexus 1000V first sends a gratuitous ARP packet to the upstream switch indicating that the VEM MAC address will now be learned on a different link. It also allows for sub-second failover time.

Figure 5-3 shows each member port that is assigned to a specific port channel subgroup using MAC pinning.

Figure 5-3 Using MAC Pinning to Connect a Port Channel to Multiple Upstream Switches



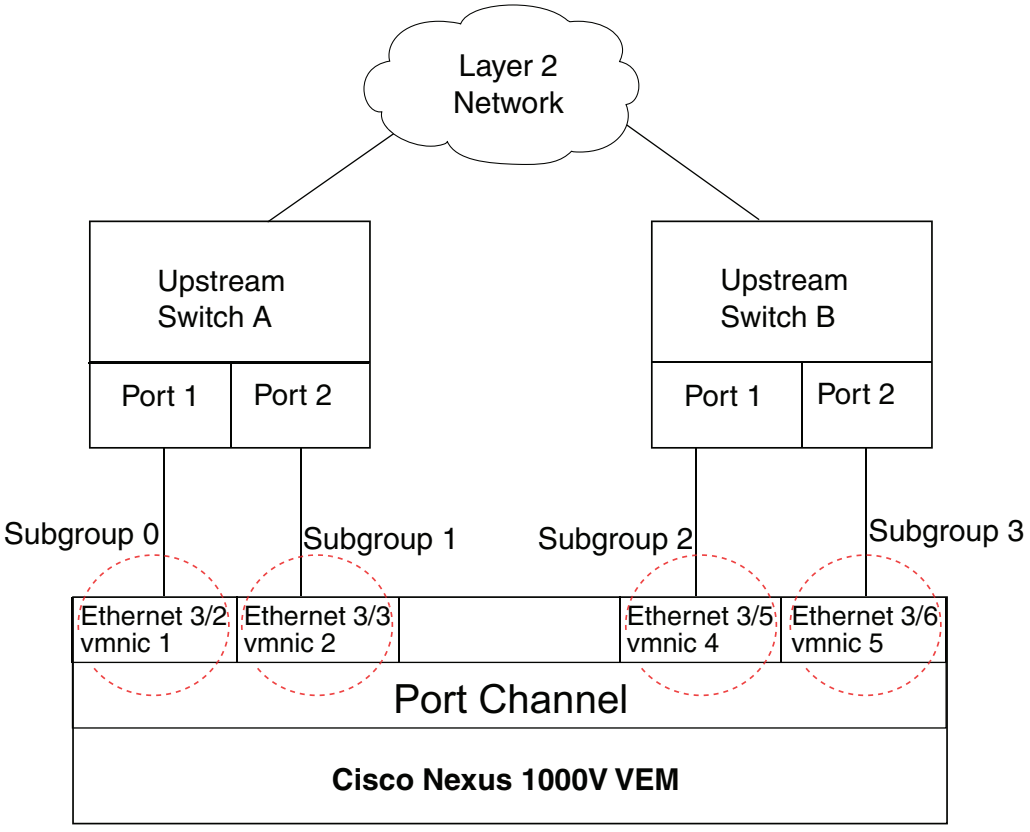
MAC Pinning Relative

This feature modifies the existing algorithm for MAC pinning where the port-channel uses the port number (vmnic number) as the subgroup ID for an Ethernet member port.

The new algorithm assigns zero-based logical subgroup IDs to Ethernet member ports. The member port having the lowest port number (vmnic number) is assigned subgroup ID 0.

Figure 5-4 shows each member port that is assigned to a specific port channel subgroup using MAC pinning relative.

Figure 5-4 Using MAC Pinning Relative to Connect a Port Channel to Multiple Upstream Switches



Network State Tracking for VPC-HM

Network state tracking for VPC-HM identifies link failures where other detection methods fail, and verifies Layer 2 connectivity between vPC-HM channel sub groups. It is not intended to detect network configuration problems.

Network state tracking selects one uplink interface in each sub group for broadcasting packets to a tracking VLAN. The tracking VLAN is usually the lowest forwarding VLAN for trunk ports and the primary VLAN for promiscuous access ports. Packets received back from the network on each sub group are tracked as are the number of consecutively missed broadcasts. If the missed broadcasts for a sub group exceed the threshold, the port channel is considered to be in split mode. When in split mode, the interfaces are marked as inactive, and traffic is pinned to active interfaces.

System messages indicate when a port channel enters or recovers from split mode; and interfaces are marked active or inactive.

For more information, see the "Configuring Network State Tracking for vPC-HM" procedure on page 5-30.

High Availability

Port channels provide high availability by load balancing traffic across multiple ports. If a physical port fails, the port channel is still operational if there is an active member in the port channel.

Port channels support stateful and stateless restarts. A stateful restart occurs on a supervisor switchover. After the switchover, the Cisco Nexus 1000V applies the runtime configuration after the switchover.

Prerequisites for Port Channels

Port channeling has the following prerequisites:

- You are logged into the Cisco Nexus 1000V in EXEC mode.
- All ports for a single port channel must meet the compatibility requirements. See the "Compatibility Checks" section on page 5-2 for more information about the compatibility requirements.
- You can use virtual vPC-HM to configure a port channel even when the physical ports are connected to two different switches.

Guidelines and Limitations

Port channeling has the following guidelines and restrictions:

- All ports in the port channel must be in the same Cisco Nexus 1000V module; you cannot configure port channels across Cisco Nexus 1000V modules.
- Port channels can be formed with multiple upstream links only when they satisfy the compatibility requirements and under the following conditions:
  - The uplinks from the host are going to the same upstream switch.
  - The uplinks from the host going to multiple upstream switches are configured with vPC-HM.
- You can configure multiple port channels on a device.
- After you configure a port channel, the configuration that you apply to the port channel interface affects the port channel member ports. The configuration that you apply to the member ports affects only the member port where you apply the configuration.
- You must remove the port security information from a port before you can add that port to a port channel. Similarly, you cannot apply the port security configuration to a port that is a member of a channel group.
- You can configure ports that belong to a port channel group as PVLAN ports.
- Any configuration changes that you apply to the port channel is applied to every member interface of that port channel.
- Channel member ports cannot be a source or destination SPAN port.
- In order to support LACP when inband/AIPC are also carried over the link, you must configure the following commands on the ports connected to the ESX host:
  - **spanning-tree portfast trunk**
  - **spanning-tree bpdufilter enable**

Note If you have a separate dedicated NIC for control traffic, these settings are not required.

- There should be at least two links that connect two switches when inband/AIPC are also carried over the LACP channel.
- If you configure LACP and your upstream switch uses the LACP suspend feature, make sure this feature is disabled. For more information, see the documentation for your upstream switch, such as:

  Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 5.x
- If you are connecting to an upstream switch or switches that do not support port channels, then MAC pinning is the preferred configuration. MAC pinning divides the uplinks from your server into standalone links and pins the MAC addresses to those links in a round-robin method. The drawback is that you cannot leverage the load sharing performance that LACP provides.
- Once a port profile is created, you cannot change its type (Ethernet or vEthernet).
- The server administrator should not assign more than one uplink on the same VLAN without port channels. It is not supported to assign more than one uplink on the same host to a profile without port channels or port profiles that share one or more VLANs.

Caution Disruption of connectivity may result if you configure vPC-HM on the Cisco Nexus 1000V when vPC is also configured on the ports of upstream switches that connect to its VEMs.

- You must have already configured the Cisco Nexus 1000V software using the setup routine. For information, see the *Cisco Nexus 1000V Getting Started Guide, Release 4.2(1)SV1(5.1)*.
- The Cisco Nexus 1000V must be connected to the vCenter Server.
- You are logged in to the CLI in EXEC mode.
- When you create a port channel, an associated channel group is automatically created.
- If LACP support is required for the port channel, then the LACP feature must be enabled before you can configure it.
- Network State Tracking is only supported with HP Virtual Connect where one physical link from the Flex-10 fabric appears as four Flex-10 NICs (physical NICs) to the VMkernel. For more information, see the "Network State Tracking for VPC-HM" section on page 5-11.

Default Settings

The following table lists the default settings for port channels.

Parameters	Default
Port profile type	vEthernet
Port profile administrative state	all ports disabled
Port channel	Admin up
LACP	Disabled **Note** If upgrading to Release 4.2(1)SV1(5.1) from a previous release, LACP is enabled by default.

Parameters	Default
Load balancing method for Layer 2 interfaces	Source and destination MAC address
Load balancing per module	Disabled
Channel mode	on
LACP offload (Offloading LACP management to VEMs)	Enabled **Note** If upgrading to Release 4.2(1)SV1(5.1) from a previous release, LACP offload is disabled by default.
Network State Tracking:	
Broadcast interval	5 seconds
Split-network mode action	repin
Maximum threshold miss count	5 seconds
State	Disabled

Configuring Port Channels

This section includes the following topics:

Note Be aware that the Cisco Nexus 1000V commands may differ from the Cisco IOS commands.

Creating a Port Profile for a Port Channel

You can use the procedures in this section to define a port channel in a port profile and, if needed, to configure and pin interface or VLAN subgroups.

Connecting to a Single Upstream Switch

You can configure a port channel whose ports are connected to the same upstream switch.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- If the ports are connected to multiple upstream switches, see the "Connecting to Multiple Upstream Switches" section on page 5-17.
- The channel group number assignment is made automatically when the port profile is assigned to the first interface.

SUMMARY STEPS

1. **configure terminal**
2. **port-profile** [**type** {**ethernet** | **vethernet**}] *name*
3. **channel-group auto** [**mode** {**on** | **active** | **passive**} [**sub-group** {**cdp** | **manual**}] [**mac-pinning** [**relative**]]
4. **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Description
Step 1	**configure terminal** **Example:** n1000v# configure terminal n1000v(config)#	Enters global configuration mode.
Step 2	**port-profile** [**type** {**ethernet** | **vethernet**}] *name* **Example:** n1000v(config)# port-profile AccessProf n1000v(config-port-prof)#	Enters port profile configuration mode for the named port profile. • *name*—Specifies the port profile name, which can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V. • **type**—(Optional) Specifies the port profile as an Ethernet or vEthernet type. Once configured, this setting cannot be changed. The default is the vEthernet type. For configuring port channels, specify the port profile as an Ethernet type. Defining a port profile as an Ethernet type allows the port profile to be used for physical (Ethernet) ports. In the vCenter Server, the corresponding port group can be selected and assigned to physical ports (PNICs). **Note** If a port profile is configured as an Ethernet type, then it cannot be used to configure VMware virtual ports.
Step 3	**channel-group auto** [**mode** {**on** | **active** | **passive**}] [**mac-pinning** [**relative**]] **Example:** n1000v(config-port-prof)# channel-group auto mode on n1000v(config-port-prof)# **Example:** n1000v(config-port-prof)# channel-group auto mode on mac-pinning n1000v(config-port-prof)# **Example:** n1000v(config-port-prof)# channel-group auto mode on mac-pinning relative n1000v(config-port-prof)#	Defines a port channel group in which a unique port channel is created and automatically assigned when the port profile is assigned to the first interface. Each additional interface that belongs to the same module is added to the same port channel. In VMware environments, a different port channel is created for each module. • **mode—**Sets the port channel mode to **on**, **active**, or **passive** (active and passive use LACP). • **mac-pinning**—If the upstream switch does not support port channels, this designates that one subgroup per Ethernet member port must be automatically assigned, – **relative** - The subgroup numbering begins at zero and continues numbering the subgroups consecutively.

	Command	Description
Step 4	**show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*] **Example:** n1000v(config-port-prof)# show port-profile name AccessProf	(Optional) Displays the configuration for verification.
Step 5	**copy running-config startup-config** **Example:** n1000v(config-port-prof)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

This example shows how to configure a port channel that connects to one upstream switch:

```
Example:
n1000v  configure terminal
n1000v(config)# port-profile AccessProf
n1000v(config-port-prof)# channel-group auto mode on
n1000v(config-port-prof)# show port-profile name AccessProf
port-profile AccessProf
  description: allaccess4
  status: disabled
capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    channel-group auto mode on
  evaluated config attributes:
    channel-group auto mode on
  assigned interfaces:
n1000v(config-port-prof)#
```

Connecting to Multiple Upstream Switches

You can create a port channel that connects to multiple upstream switches,.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- If the ports are connected to a single upstream switch, see the "Connecting to a Single Upstream Switch" procedure on page 5-15.
- You can configure an uplink port profile to be used by the physical NICs in the VEM in virtual port channel-host mode (vPC-HM) when the ports connect to multiple upstream switches.
- If you are connecting to multiple upstream switches that do not support port channels, then MAC pinning is the preferred configuration. You can configure MAC pinning using this procedure. For more information about the feature, see the "MAC Pinning" section on page 5-10.
- The channel group mode must be set to **on** (active and passive modes use LACP).

- You need to know whether CDP is configured in the upstream switches.
  - If configured, then CDP packets from the upstream switch are used to automatically create a subgroup for each upstream switch to manage its traffic separately.
  - If not configured, then, after completing this procedure, you must manually configure subgroups to manage the traffic flow on the separate switches. See the "Manually Configuring Interface Subgroups" procedure on page 5-22.

Caution Connectivity may be disrupted for up to 60 seconds if the CDP timer is set to 60 seconds (the default).

Caution The VMs behind the Cisco Nexus 1000V receive duplicate packets from the network for unknown unicasts, multicast floods, and broadcasts if vPC-HM is not configured when port channels connect to two different upstream switches.

SUMMARY STEPS

1. **configure terminal**
2. **port-profile** [**type** {**ethernet** | **vethernet**}] *name*
3. **channel-group auto mode on** [**sub-group** {**cdp** | **manual**}] [**mac-pinning** [**relative**]]
4. **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Description

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**port-profile** [**type** {**ethernet** | **vethernet**}] *name*
	
	Example:
	`n1000v(config)# port-profile uplinkProf`
	`n1000v(config-port-prof)#`
		Creates an Ethernet type port profile (the default) and enters port profile configuration mode for that port profile.
		• *name*—Specifies the port profile name, which can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V.
		• **type**—(Optional) Specifies the port profile as an Ethernet or vEthernet type. Once configured, this setting cannot be changed. The default is the vEthernet type.
		For configuring port channels, specify the port profile as an Ethernet type.
		Defining a port profile as an Ethernet type allows the port profile to be used for physical (Ethernet) ports. In the vCenter Server, the corresponding port group can be selected and assigned to physical ports (PNICs).
		Note If a port profile is configured as an Ethernet type, then it cannot be used to configure VMware virtual ports.

	Command	Description
Step 3	**channel-group auto mode on** [**sub-group** {**cdp** | **manual**}] [**mac-pinning** [**relative**]] **Example—CDP is configured on the upstream switches:** n1000v(config-port-prof)# channel-group auto mode on sub-group cdp n1000v(config-port-prof)# **Example—CDP is not configured on the upstream switches:** n1000v(config-port-prof)# channel-group auto mode on manual n1000v(config-port-prof)# **Example—Upstream switches do not support port channels:** n1000v(config-port-prof)# channel-group auto mode on mac-pinning n1000v(config-port-prof)# **Example—MAC pinning relative:** n1000v(config-port-prof)# channel-group auto mode on mac-pinning relative n1000v(config-port-prof)#	Creates a unique asymmetric port channel (also known as vPC-HM) and automatically assigns it when the port profile is assigned to the first interface. Each additional interface that belongs to the same module is added to the same port channel. In VMware environments, a different port channel is created for each module. The following options are also defined: • **mode—**Sets the port channel mode to **on** (active and passive use LACP). • **sub-group**—Identifies this channel group as asymmetric, or connected to more than one switch. – **cdp**—Specifies that CDP information is used to automatically create subgroups for managing the traffic flow. – **manual**—Specifies that subgroups are configured manually. This option is used if CDP is not configured on the upstream switches. To configure subgroups, see the "Manually Configuring Interface Subgroups" procedure on page 5-22. • **mac-pinning**—Specifies that Ethernet member ports are assigned to subgroups automatically, one subgroup per member port. This option is used if the upstream switch does not support port channels. – **relative** - The subgroup numbering begins at zero and continues numbering the subgroups consecutively.
Step 4	**show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*] **Example:** n1000v(config-port-prof)# show port-profile name AccessProf	(Optional) Displays the configuration for verification.
Step 5	**copy running-config startup-config** **Example:** n1000v(config-port-prof)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

This example shows how to create a port channel that connects to multiple upstream switches that support CDP:

```
n1000v(config)# port-profile UpLinkProfile2
n1000v(config-port-prof)# channel-group auto mode on sub-group cdp
n1000v(config-port-prof)# show port-profile name UpLinkProfile2
port-profile UpLinkProfile2
  description:
  type: ethernet
  status: disabled
```

```
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    channel-group auto mode on sub-group cdp
  evaluated config attributes:
    channel-group auto mode on sub-group cdp
  assigned interfaces:
n1000v(config-port-prof)# copy running-config startup-config
```

This example shows how to create a port channel that connects to multiple upstream switches that do not support CDP:

```
n1000v(config)# port-profile UpLinkProfile3
n1000v(config-port-prof)# channel-group auto mode on sub-group manual
n1000v(config-port-prof)# exit
n1000v(config)# interface ethernet3/2-3
n1000v(config-if)# sub-group-id 0
n1000v(config-port-prof)# show port-profile name
n1000v(config-port-prof)# show port-profile name UplinkProfile3
port-profile UplinkProfile3
  description:
  type: ethernet
  status: enabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group: UplinkProfile3
  max ports: -
  inherit:
  config attributes:
    channel-group auto mode on sub-group manual
  evaluated config attributes:
    channel-group auto mode on sub-group manual
  assigned interfaces:
n1000v(config-port-prof)# copy running-config startup-config
```

This example shows how to create a port channel that connects to multiple upstream switches that do not support port channels:

```
n1000v(config)# port-profile UpLinkProfile1
n1000v(config-port-prof)# channel-group auto mode on mac-pinning
n1000v(config-port-prof)# show port-profile name UpLinkProfile1
port-profile UpLinkProfile1
  description:
  type: ethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    channel-group auto mode on mac-pinning
  evaluated config attributes:
    channel-group auto mode on mac-pinning
  assigned interfaces:
n1000v(config-port-prof)# copy running-config startup-config
```

Manually Configuring Interface Subgroups

You can manually configure port channel subgroups to manage the traffic flow on multiple upstream switches. This is required for a port channel that connects to multiple upstream switches where CDP is not configured.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You have already configured the port profile for the port channel using the "Connecting to Multiple Upstream Switches" procedure on page 5-17.
- You know the interface range and the subgroup IDs (0-31) for traffic to the upstream switches.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *range*
3. **sub-group-id** *number*
4. Repeat step 2 and 3 for each port connected to an upstream switch where CDP is not configured.
5. **show interface ethernet** *range*
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Description
Step 1	**configure terminal** **Example:** n1000v# configure terminal n1000v(config)#	Enters global configuration mode.
Step 2	**interface ethernet** *range* **Example:** n1000v(config)# interface ethernet3/2-3 n1000v(config-if)#	Enters interface configuration mode for the specified interface range.
Step 3	**sub-group id** *number* **Example:** n1000v(config-if)# sub-group-id 0 n1000v(config-if)#	Manually configures a subgroup to manage traffic for the upstream switch. Allowable subgroup numbers are from 0 to 31.
Step 4	Repeat Step 2 and Step 3 for each port connected to an upstream switch where CDP is not configured.	
Step 5	**show interface ethernet** *range* **Example:** n1000v(config-if)# show interface ethernet 3/2-3	(Optional) Displays the configuration for verification.
Step 6	**copy running-config startup-config** **Example:** n1000v(config-if)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

This example shows how to manually configure port channel subgroups for a host in module 3 which has four physical ports. The upstream switches do not support CDP. Ethernet ports 3/2 and 3/3 connect to one upstream switch and the Ethernet ports 3/4 and 3/5 connect to another.

```
n1000v# conf t
n1000v(config)# int eth3/2
n1000v(config-if)# sub-group-id 0
n1000v(config-if)# int eth3/3
n1000v(config-if)# sub-group-id 0
n1000v(config-if)# int eth3/4
n1000v(config-if)# sub-group-id 1
n1000v(config-if)# int eth3/5
n1000v(config-if)# sub-group-id 1
n1000v(config-if)# show running-config interface
. . .
interface Ethernet3/2
  inherit port-profile system-uplink-pvlan
  sub-group-id 0
interface Ethernet3/3
  inherit port-profile system-uplink-pvlan
  sub-group-id 0
interface Ethernet3/4
  inherit port-profile system-uplink-pvlan
  sub-group-id 1
interface Ethernet3/5
  inherit port-profile system-uplink-pvlan
  sub-group-id 1
```

Pinning a vEthernet Interface to a Subgroup

You can pin a vEthernet interface to a specific port channel subgroup in the port profile configuration.

Note You can also pin a subgroup to a vEthernet interface in the interface configuration. For information, see the "Configuring Static Pinning for an Interface" procedure on page 5-32.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You know the subgroup ID (0-31) for the vEthernet interface.

SUMMARY STEPS

1. **configure terminal**
2. **port-profile type vethernet** *name*
3. **pinning id** *subgroup_id* [**backup** *subgroup_id1...subgroup_id7*]
4. **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Description

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**port-profile type vethernet** *name*
	
	Example:
	`n1000v(config)# port-profile type vethernet PortProfile1`
	`n1000v(config-port-prof)#`
		Enters port profile configuration mode for the named profile.

Step 3	**pinning id** *subgroup_id* [**backup** *subgroup_id1...subgroup_id7*] **Example:** `n1000v(config-port-prof)# pinning id 3 backup 4`	For the named port profile, assigns (or pins) a vEthernet interface to a port channel subgroup (0–31). **backup** - Optionally specify an ordered list of backup sub-groups for pinning to be used if the primary sub-group is not available.

	Command	Description
Step 4	**show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*] **Example:** n1000v(config-port-prof)# show port-profile PortProfile1	(Optional) Displays the configuration for verification.
Step 5	**copy running-config startup-config** **Example:** n1000v(config-port-prof)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

This example shows how to create a vEthernet port profile and pin it to port channel subgroup 3:

```
n1000v# configure terminal
n1000v(config)# port-profile type vethernet PortProfile1
n1000v(config-port-prof)# pinning id 3
n1000v(config-port-prof)# show port-profile name PortProfile1
port-profile PortProfile1
  description:
  type: vethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    pinning id 3
  evaluated config attributes:
    pinning id 3
  assigned interfaces:
n1000v(config-port-prof)# copy running-config startup-config
```

This example shows how to create a vEthernet port profile and pin it to port channel subgroup 3 and backup subgroups 4 and 6.:

```
n1000v# configure terminal
n1000v(config)# port-profile type vethernet PortProfile1
n1000v(config-port-prof)# pinning id 3 backup 4 6
n1000v(config-port-prof)# show port-profile name PortProfile1
port-profile PortProfile1
  description:
  type: vethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
```

```
    pinning id 3 backup 4 6
  evaluated config attributes:
    pinning id 3
  assigned interfaces:
n1000v(config-port-prof)# copy running-config startup-config
```

Pinning a Control or Packet VLAN to a Subgroup

You can pin a control or packet VLAN to a specific subgroup.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- The existing port profile must be a system port profile.
- The port profile must be an Ethernet type.
- If you are pinning a control or packet VLAN, it must already be in the port profile.
  - If you are pinning a control VLAN, the control VLAN must already be one of the system VLANs in the port profile.

SUMMARY STEPS

1. **configure terminal**
2. **port-profile** *name*
3. **pinning** {**control-vlan** | **packet-vlan**} *subgroup_id*
4. **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Description

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**port-profile** *name*
	
	Example:
	`n1000v(config)# port-profile SystemProfile1`
	`n1000v(config-port-prof)#`
		Enters port profile configuration mode for the named port profile.

Step 3	**pinning** {**control-vlan** | **packet-vlan**} *subgroup_id*
	
	Example:
	`n1000v(config-port-prof)# pinning control-vlan 3`
	`n1000v(config-port-prof)#`
		Assigns (or pins) a control VLAN or packet VLAN to a port channel subgroup (0–31).

	Command	Description
Step 4	**show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*] **Example:** n1000v(config-port-prof)# show port-profile SystemProfile1	(Optional) Displays the configuration for verification.
Step 5	**copy running-config startup-config** **Example:** n1000v(config-port-prof)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

This example shows how to configure static pinning on a control VLAN:

```
n1000v# configure terminal
n1000v(config)# port-profile SystemProfile1
n1000v(config-port-prof)# pinning control-vlan 3
n1000v(config-port-prof)# show port-profile SystemProfile1
port-profile SystemProfile1
  description:
  type: ethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: 3
  pinning packet-vlan: -
  system vlans: 1
  port-group: SystemProfile1
  max ports: -
  inherit:
  config attributes:
    switchport mode trunk
    switchport trunk allowed vlan 1-5
    no shutdown
  evaluated config attributes:
    switchport mode trunk
    switchport trunk allowed vlan 1-5
    no shutdown
  assigned interfaces:
n1000v(config-port-prof)# copy running-config startup-config
```

This example shows how to configure static pinning on a packet VLAN:

```
n1000v# configure terminal
n1000v(config)# port-profile SystemProfile1
n1000v(config-port-prof)# pinning packet-vlan 0
n1000v(config-port-prof)# show port-profile name SystemProfile1
port-profile SystemProfile1
  description:
  type: ethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: 0
  system vlans: 1
  port-group:
  max ports: -
  inherit:
  config attributes:
```

```
    switchport mode access
    switchport access vlan 1
    switchport trunk native vlan 1
    no shutdown
  evaluated config attributes:
    switchport mode access
    switchport access vlan 1
    switchport trunk native vlan 1
    no shutdown
  assigned interfaces:
n1000v(config-port-prof)# copy running-config startup-config
```

Migrating a Channel Group to a Port Profile

You can migrate a channel group to a port profile.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You are logged into vCenter server on the host server.

Step 1 Place the host in maintenance mode.

Step 2 Do one of the following:

- If distributed resource scheduling (DRS) is enabled, make sure to wait until the virtual machines are migrated to other host(s).
- Otherwise, manually migrate the virtual machines.

Step 3 When all the virtual machines are successfully migrated, from the Cisco Nexus 1000V CLI, create a new Ethernet type port profile for the uplink ports on this host with the needed parameters including the following.

- One of the following:
  - **channel-group auto mode active/passive**
  - **channel-group auto mode on mac-pinning**.
- CLI overrides on the existing port channels.

Step 4 Remove the port channel configuration from the uplink switches.

Step 5 From vCenter on the host, move the port(s) to the new port profile.

Step 6 Verify that the port(s) are successfully bundled into the new port channel.

Note The new port channel has a new port channel ID.

Step 7 When all the port(s) are moved from the old port profile, use the following command from the Cisco Nexus 1000V CLI to delete the port channels with zero members:

no interface port-channel *id*

Step 8 Bring the host out of maintenance mode.

Step 9 Migrate the virtual machines back to this host.

Step 10 Use the following command from the Cisco Nexus 1000V to save the running configuration persistently through reboots and restarts by copying it to the startup configuration.

copy running-config startup-config

Step 11 Create the port channel type in the upstream switch. For more information, see Creating a Port Profile for a Port Channel, page 5-14.

Migrating Port Profile Types in a Port Profile

To move port profile types in a port profile, you tear down the existing port channel then recreate the port channel. These steps use procedures documented in other sections of this chapter.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

Step 1 Place the host in maintenance mode.

Step 2 Do one of the following:

- If distributed resource scheduling (DRS) is enabled, make sure to wait until the virtual machines are migrated to other host(s).
- Otherwise, manually migrate the virtual machines.

Step 3 When all the virtual machines are successfully migrated, from the Cisco Nexus 1000V CLI, create a new Ethernet type port profile for the uplink ports on this host with the needed parameters including the following.

- One of the following:
  - **channel-group auto mode active/passive**
  - **channel-group auto mode on mac-pinning**.
- CLI overrides on the existing port channels.

Step 4 Remove the port channel you want to migrate in the upstream switch. For more information, see Removing a Port Channel Group from a Port Profile, page 5-34.

Step 5 Remove the port channel in the upstream switch.

Step 6 Manually configure subgroup IDs in the Nexus 1000V Ethernet interface. For more information, see Manually Configuring Interface Subgroups, page 5-22.

Note Follow this step if you want the to use the port channel in manual mode.

Step 7 Change the port channel type in the Nexus 1000v port profile. For more information, see Migrating a Channel Group to a Port Profile, page 5-28.

Step 8 Change the port channel type in the Nexus 1000v port profile. For more information, see Connecting to a Single Upstream Switch, page 5-15.

Step 9 Bring the host out of maintenance mode.

Step 10 Migrate the virtual machines back to this host.

Step 11 Use the following command from the Cisco Nexus 1000V to save the running configuration persistently through reboots and restarts by copying it to the startup configuration.

copy running-config startup-config

Step 12 Create the port channel type you want in the upstream switch. For more information, see Creating a Port Profile for a Port Channel, page 5-14.

Configuring Network State Tracking for vPC-HM

You can configure Network State Tracking to pinpoint link failures on port channels configured for vPC-HM.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- Once enabled, Network State Tracking is used on every VEM that is configured with a vPC-HM port profile.
- If you specify repinning (the default) and a split network is detected, then Ethernet interfaces are inactivated, and the vEths are redistributed among all interfaces including the reactivated Ethernet interfaces. Restoration to the earlier pinned state is not guaranteed.
- For more information about Network State Tracking, see the "Network State Tracking for VPC-HM" section on page 5-11.

SUMMARY STEPS

1. **configure terminal**
2. **track network-state enable**
3. **(Optional) track network-state interval** *seconds*
4. **(Optional) track network-state split action [repin | log-only]**
5. **(Optional) track network-state threshold miss-count** *count*
6. **show network-state tracking config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**track network-state enable**
	
	Example:
	`n1000v(config)# track network-state enable`
	`n1000v(config)#`
		Enables Network State Tracking on all interfaces in vPC-HM port-channels.

Step 3	**track network-state interval** *seconds*
	
	Example:
	`n1000v(config)# track network-state interval 8`
	`n1000v(config)#`
		(Optional) Specifies the interval of time, from 1 to 10 seconds, between which tracking broadcasts are sent; and the interval for tracking packets. The default interval is 5 seconds between broadcasts.

Step 4	**track network-state split action [repin | log-only]**
	
	Example:
	`n1000v(config)# track network-state split action repin`
	`n1000v(config)#`
		(Optional) Specifies the action to be taken if a split network is detected.
		• **repin**: pins traffic to another uplink. (the default)
		• **no repin**: leaves vEths where they are.

Step 5	**track network-state threshold miss-count** *count*
	
	Example:
	`n1000v(config)# track network-state threshold miss-count 7`
	`n1000v(config)#`
		(Optional) Specifies the maximum number of broadcasts that can be missed successively (from 3 to 7) before a split network is declared. The default is 5 missed broadcasts.

Step 6	**show network-state tracking config**
	
	Example:
	`n1000v(config)# show network-state tracking config`
	`Tracking mode : disabled`
	`Tracking Interval : 8 sec`
	`Miss count threshold : 7 pkts`
	`Split-network action : repin`
	`n1000v(config)#`
		(Optional) Displays the Network State Tracking configuration for verification.

Step 7	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to configure Network State Tracking with an 8 second interval between each sent broadcast, repinning traffic to another uplink if a split network is detected, and a maximum of 7 missed broadcasts before declaring a split network:

```
configure terminal
```

```
track network-state enable
track network-state interval 8
track network-state split action repin
track network-state threshold miss-count 7
show network-state tracking config
Tracking mode        : enabled
Tracking Interval    : 8 sec
Miss count threshold : 7 pkts
Split-network action : repin
n1000v(config)#
```

Configuring Static Pinning for an Interface

You can configure static pinning on a vEthernet interface.

Note You can also pin a subgroup to a vEthernet interface in the port profile configuration. For information, see the "Pinning a vEthernet Interface to a Subgroup" procedure on page 5-24.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface vethernet** *interface-number*
3. **pinning id** *subgroup_id* [**backup** *subgroup_id1...subgroup_id7*]
4. **show running-config interface vethernet** *interface-number*
5. **module vem** *module_number* **execute vemcmd show pinning**
6. **module vem** *module_number* **execute vemcmd show static pinning config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Description

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**interface vethernet** *interface-number*
	
	Example:
	`n1000v(config)# interface vethernet 1`
	`n1000v(config-if)#`
		Enters interface configuration mode for the specified interface (from 1 to 1048575).

	Command	Description
Step 3	**pinning id** *subgroup_id* [**backup** *subgroup_id1...subgroup_id7*] **Example:** n1000v(config-if)# pinning id 0 backup 1 2	Assigns (or pins) a vEthernet interface to a specific port channel subgroup (from 0 to 31). **backup** - Optionally specify an ordered list of backup sub-groups for pinning to be used if the primary sub-group is not available.
Step 4	**show running-config interface vethernet** *interface-number* **Example:** n1000v(config-if)# show running-config interface vethernet 1	(Optional) Displays the pinning configuration of the specified interface.
Step 5	**module vem** *module_number* **execute vemcmd show pinning** **Example:** n1000v(config-if)# module vem 3 execute vemcmd show pinning	(Optional) Displays the pinning configuration on the specified VEM.
Step 6	**module vem** *module_number* **execute vemcmd show static pinning config** **Example:** n1000v(config-if) module vem 3 execute vemcmd show static pinning config	(Optional) Displays the VSM configured pinning subgroups.
Step 7	**copy running-config startup-config** **Example:** n1000v(config-if)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to pin subgroup ID 0 to vEthernet interface 1:

```
n1000v(config)# configure terminal
n1000v(config)# interface vethernet 1
n1000v(config-if)# pinning id 0
n1000v(config-if)# show running-config interface vethernet 1
version 4.0(4)SV1(2)

interface Vethernet3
  service-policy type qos input policy1
  pinning id 0

n1000v(config-if)# exit
n1000v(config)# exit
n1000v# module vem 3 execute vemcmd show pinning
  LTL    IfIndex  PC_LTL  VSM_SGID  VEM_SGID  Eff_SGID
   48   1b040000     304         0         0         0
```

The following example shows the output after configuring backup subgroups for pinning:

```
n1000v(config-if)# module vem 4 execute vemcmd show static pinning config
  LTL    IfIndex   VSM_SGID  Backup_SGID
   48   1c0000a0      0,          1,2
   50   1c000100      0,          1

n1000v(config-if)# copy running-config startup-config
```

Removing a Port Channel Group from a Port Profile

You can remove a port channel group from a port profile.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **configure terminal**
2. **port-profile** *name*
3. **no channel-group auto**
4. **show**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**port-profile** *name*
	
	Example:
	`n1000v(config)# port-profile testProf`
	`n1000v(config-port-prof)#`
		Specifies the port profile from which the port channel will be removed.

Step 3	**no channel-group auto**
	
	Example:
	`n1000v(config-port-prof)# no channel-group auto`
	`n1000v(config-port-prof)#`
		Removes the channel group configuration from all member interfaces in the specified port profile.

Step 4	**show port-profile** *name* **Example:** `n1000v(config)# show port-profile testProf`	Displays the configuration for verification.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Shutting Down and Restarting a Port Channel Interface

You can shut down and restart a port channel interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- When you shut down a port channel interface, no traffic passes, and the interface is administratively down.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel-number*
3. **shutdown | no shutdown**
4. **show interface port-channel** *channel-number*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**interface port-channel** *channel-number*
	
	Example:
	`n1000v(config)# interface port-channel 2`
	`n1000v(config-if)`
		Enters interface configuration mode for the specified port channel interface.

Step 3	**shutdown** **Example:** `n1000v(config-if)# shutdown`	Shuts down the interface. No traffic passes and the interface displays as administratively down. The default is **no shutdown**.
	no shutdown **Example:** `n1000v(config-if)# no shutdown`	Brings the interface back up. The interface displays as administratively up. If there are no operational problems, traffic passes. The default is **no shutdown**.
Step 4	**show interface port-channel** *channel-number* **Example**: `n1000v(config-if)# show interface port-channel 2`	(Optional) Displays interface information for the specified port channel.
Step 5	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to bring up the interface for port channel 2:

```
n1000v# configure terminal
n1000v(config)# interface port-channel 2
n1000v(config-if)# no shutdown
```

Adding a Description to a Port Channel Interface

You can add a description to a port channel interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel-number*
3. **description** *string*
4. **show interface port-channel** *channel-number*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**interface port-channel** *channel-number*
	
	Example:
	`n1000v(config)# interface port-channel 2`
	`n1000v(config-if)`
		Places you into interface configuration mode for the specified port channel interface.
		
		For *channel number*, the range is from 1 to 4096. The port channel associated with this channel group is automatically created if the port channel does not already exist.

Step 3	**description** *string* **Example:** `n1000v(config-if)# description engineering`	Adds a description to the port channel interface. For *string*, the description can be up to 80 alphanumeric characters. **Note** You do not need to use quotations around descriptions that include spaces.

	Command	Purpose
Step 4	**show interface port-channel** *channel-number* **Example:** n1000v(config-if)# show interface port-channel 2	(Optional) Displays interface information for the specified port channel.
Step 5	**copy running-config startup-config** **Example:** n1000v(config-if)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to add a description to port channel 2:

```
n1000v# configure terminal
n1000v(config)# interface port-channel 2
n1000v(config-if)# description engineering
```

Configuring the Speed and Duplex Settings for a Port Channel Interface

You can configure the speed and duplex settings for a port channel interface.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel-number*
3. **speed** {**10** | **100** | **1000** | **auto**}
4. **duplex** {**auto** | **full** | **half**}
5. **show interface port-channel** *channel-number*
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** n1000v# configure terminal n1000v(config)#	Enters global configuration mode.

Step 2	**interface port-channel** *channel-number*
	
	Example:
	n1000v(config)# interface port-channel 2
	n1000v(config-if)
		Specifies the port channel interface that you want to configure and enters the interface mode.
		Allowable channel numbers are from 1 to 4096.

Step 3	**speed** {**10** | **100** | **1000** | **auto**} **Example:** n1000v(config-if)# speed auto	Sets the speed for the port channel interface. The default is **auto** for autonegotiation.
Step 4	**duplex** {**auto** | **full** | **half**} **Example:** n1000v(config-if)# speed auto	Sets the duplex mode for the port channel interface. The default is **auto** for autonegotiation.
Step 5	**show interface port-channel** *channel-number* **Example:** n1000v(config-if)# show interface port-channel 2	(Optional) Displays interface information for the specified port channel.
Step 6	**copy running-config startup-config** **Example:** n1000v(config-if)# copy running-config startup-config	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to set port channel 2 to 100 Mbps:

```
n1000v# configure terminal
n1000v(config)# interface port channel 2
n1000v(config-if)# speed 100
```

Configuring Port Channel Load Balancing

You can configure port channel load balancing.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You can configure port channel load balancing for the entire device or for a single module.
- Module-based load balancing takes precedence over device-based load balancing.
- The default load balancing method is the source MAC address.

- For more information about port channel load balance, see the "Load Balancing Using Port Channels" section on page 5-4.

SUMMARY STEPS

1. **configure terminal**
2. **port-channel load-balance ethernet** {**dest-ip-port** | **dest-ip-port-vlan** | **destination-ip-vlan** | **destination-mac** | **destination-port** | **source-dest-ip-port** | **source-dest-ip-port-vlan** | **source-dest-ip-vlan** | **source-dest-mac** | **source-dest-port** | **source-ip-port** | **source-ip-port-vlan** | **source-ip-vlan** | **source-mac** | **source-port** | **source-virtual-port-id** | **vlan-only**} [**module** *module_number*]
3. **show port-channel load-balance**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**port-channel load-balance ethernet** {**dest-ip-port** | **dest-ip-port-vlan** | **destination-ip-vlan** | **destination-mac** | **destination-port** | **source-dest-ip-port** | **source-dest-ip-port-vlan** | **source-dest-ip-vlan** | **source-dest-mac** | **source-dest-port** | **source-ip-port** | **source-ip-port-vlan** | **source-ip-vlan** | **source-mac** | **source-port** | **source-virtual-port-id** | **vlan-only**} **Example:** `n1000v(config)# port-channel load-balance ethernet source-destination-mac`	Configures the load balance method for the device or module. The range depends on the device. The default load balancing method uses the source MAC address.
Step 3	**show port-channel load-balance** **Example**: `n1000v(config)# show port-channel load-balance`	(Optional) Displays the port channel load-balancing method.
Step 4	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLES

The following example shows how to configure the source IP load-balancing method for port channels on module 5:

```
n1000v# configure terminal
n1000v(config)# port-channel load-balance ethernet source-ip module 5
```

Restoring the Default Load-Balancing Method

You can restore the default load-balancing method.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.

SUMMARY STEPS

1. **configure terminal**
2. **no port-channel load-balance ethernet**
3. **show port-channel load-balance**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**no port-channel load-balance ethernet** **Example:** `n1000v(config)# no port-channel load-balance ethernet`	Restores the default load-balancing method, which is the source MAC address.
Step 3	**show port-channel load-balance** **Example:** `n1000v(config)# show port-channel load-balance`	(Optional) Displays the port channel load-balancing method.
Step 4	**copy running-config startup-config** **Example:** `n1000v(config)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Configuring LACP for Port Channels

This section includes the following procedures:

- Configuring an LACP Port Channel, page 5-41
- Configuring VEM Management of LACP, page 5-44

Configuring an LACP Port Channel

You can configure the following requirements for LACP:

- Enable LACP support for port channels.
- Configure the individual port channel links so that they are allowed to operate with LACP.
- Configure a system uplink port profile for LACP.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- The default port channel mode is **on**.
- The LACP feature support must be enabled before you can configure LACP. This procedure has a step for enabling the LACP feature.
- When you configure port channels with no associated aggregation protocol, all interfaces on both sides of the link remain in the **on** channel mode.
- The LACP mode for individual links in an LACP port channel indicates that the link is allowed to operate with LACP.
- You have defined a native VLAN for the trunk port. Although it may not be used for data, the native VLAN is used for LACP negotiation. If you want traffic forwarded on the native VLAN of the trunk port, the native VLAN must be in the allowed VLAN list and system VLAN list.

  This procedure includes steps to add VLANs to the allowed VLAN list and system VLAN list for the port channel.

SUMMARY STEPS

1. **configure terminal**
2. **feature lacp**
3. **port-profile** [**type** {**ethernet** | **vethernet**}] *name*
4. **vmware port-group** [*pg_name*]
5. **switchport mode** {**access** | **private-vlan** {**host** | **promiscuous**} | **trunk**}
6. **switchport trunk allowed vlan** *vlan-id-list*
7. **channel-group auto** [**mode** {**on** | **active** | **passive**}] **mac-pinning**
8. **system vlan** *vlan-id-list*
9. **state enabled**
10. **show port-channel summary**
11. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**feature lacp**
	
	Example:
	`n1000v(config)# feature lacp`
	`n1000v(config)#`
		Enables LACP support for port channels.

Step 3	**port-profile** [**type** {**ethernet** | **vethernet**}] *name*
	
	Example:
	`n1000v(config-if)# port-profile type ethernet system-uplink`
	`n1000v(config-port-prof)#`
		Enters port profile configuration mode for the named port profile.
		• *name*—Specifies the port profile name, which can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V.
		• **type**—(Optional) Specifies the port profile as an Ethernet or vEthernet type. Once configured, this setting cannot be changed. The default is the vEthernet type.
		For configuring port channels, specify the port profile as an Ethernet type.
		Defining a port profile as an Ethernet type allows the port profile to be used for physical (Ethernet) ports. In the vCenter Server, the corresponding port group can be selected and assigned to physical ports (PNICs).
		Note If a port profile is configured as an Ethernet type, then it cannot be used to configure VMware virtual ports.

Step 4	**vmware port-group** [*pg_name*]
	
	Example:
	`n1000v(config-port-prof)# vmware port-group lacp`
	`n1000v(config-port-prof)#`
		Designates the port profile as a VMware port group.
		The port profile is mapped to a VMware port group of the same name unless you specify a name here. When you connect the VSM to vCenter Server, the port group is distributed to the virtual switch on the vCenter Server.

	Command	Purpose

Step 5	**switchport mode** {**access** | **private-vlan** {**host** | **promiscuous**} | **trunk**}
	
	Example:
	`n1000v(config-port-prof)# switchport mode trunk`
	`n1000v(config-port-prof)#`
		Designates how the interfaces are to be used.
		
		Allowable port modes:
		• access
		• private-vlan
		– host
		– promiscuous
		• trunk
		
		A trunk port transmits untagged packets for the native VLAN and transmits encapsulated, tagged packets for all other VLANs.

Step 6	**switchport trunk allowed vlan** *vlan-id-list*
	
	Example:
	`n1000v(config-port-prof)# switchport trunk allowed vlan 1-100`
	`n1000v(config-port-prof)#`
		Designates the port profile as trunking and defines VLAN access to it as follows:
		• **allowed-vlans**—Defines VLAN IDs that are allowed on the port.
		• **add**—Lists VLAN IDs to add to the list of those allowed on the port.
		• **except**—Lists VLAN IDs that are not allowed on the port.
		• **remove**—Lists VLAN IDs whose access is to be removed from the port.
		• **all**—Indicates that all VLAN IDs are allowed on the port, unless exceptions are also specified.
		• **none**—Indicates that no VLAN IDs are allowed on the port.
		
		If you do not configure allowed VLANs, then the default VLAN 1 is used as the allowed VLAN.
		
		If you want traffic forwarded on the native VLAN of the trunk port, the native VLAN must be in the allowed VLAN list.

Step 7	**channel-group auto** [**mode** {**on** | **active** | **passive**}] **mac-pinning**
	
	Example:
	`n1000v(config-port-prof)# channel-group auto mode active`
	`n1000v(config-port-prof)#`
		Defines a port channel group in which a unique port channel is created and automatically assigned when the port profile is assigned to the first interface.
		
		Each additional interface that belongs to the same module is added to the same port channel. In VMware environments, a different port channel is created for each module.
		• **mode—**Sets the port channel mode to **on**, **active**, or **passive** (active and passive use LACP).
		• **mac-pinning**—If the upstream switch does not support port channels, this designates that one subgroup per Ethernet member port must be automatically assigned,

	Command	Purpose

Step 8	**system vlan** *vlan-id-list*
	
	Example:
	`n1000v(config-port-prof)# system vlan 1,10,20`
	`n1000v(config-port-prof)#`
		Adds system VLANs to this port profile.
		
		If you want traffic forwarded on the native VLAN of the trunk port, the native VLAN must be in the system VLAN list.

Step 9	**state enabled**
	
	Example:
	`n1000v(config-port-prof)# state enabled`
	`n1000v(config-port-prof)#`
		Enables the port profile and applies its configuration to the assigned ports. If the port profile is a VMware port group, the port group will be created in the vswitch on vCenter Server.

Step 10	**show port-channel summary** **Example:** `n1000v(config-if)# show port-channel summary`	(Optional) Displays summary information about the port channels.
Step 11	**copy running-config startup-config** **Example:** `n1000v(config-if)# copy running-config startup-config`	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

EXAMPLE CONFIGURATION

The following example shows how to set the LACP-enabled interface to the active port channel mode for Ethernet interface 1/4 in channel group 5; and then configure an LACP port profile.

```
configure terminal
feature lacp
interface ethernet 1/4
channel-group 5 mode active
port-profile type ethernet system-uplink
vmware port-group lacp
switchport mode trunk
switchport trunk allowed vlan 1-100
channel-group auto mode active
system vlan 1,10,20
state enabled
show port-channel summary
copy running-config startup-config
```

Configuring VEM Management of LACP

Use this procedure to offload management of LACP from the VSM to the VEMs.

BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- After offloading the management of LACP from the VSM to the VEM, you must preserve the running configuration in the startup configuration and reload the VSM before the offload takes effect. This procedure has steps for doing this.
- Offloading of LACP management to the VEMs is enabled by default on the VSM.

Note If you have upgraded from a previous release, then offloading of LACP management to the VEMs is disabled by default.

You can enable or disable the feature using the **[no] lacp offload** command.

SUMMARY STEPS

1. **configure terminal**
2. **[no] lacp offload**
3. **copy running-config startup-config**
4. **show lacp offload status**
5. **reload**
6. **show lacp offload status**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`n1000v# configure terminal`
	`n1000v(config)#`
		Enters global configuration mode.

Step 2	**lacp offload**
	
	Example:
	`n1000v(config)# lacp offload`
	`Please do a "copy running startup" to ensure the new setting takes effect on next reboot`
	`LACP Offload Status can be verified using "show lacp offload status"`
	`Change in LACP Offload Status takes effect only on the next VSM Reboot`
	`This can potentially cause modules with LACP uplinks to flap`
	
	`n1000v(config)#`
		(Optional) Offloads LACP management from the VSM to the VEMs.
		
		If enabling LACP offload, a message displays to let you know that a reload is required.
		
		Offload of LACP management to the VEMs is enabled by default.
		
		Note If you upgraded from a previous release, then offload of LACP management to the VEMs is disabled by default.

Step 3	**copy running-config startup-config**
	
	Example:
	`n1000v(config-if)# copy running-config startup-config`
	`[################################] 100%`
	`n1000v(config-if)#`
		(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

	Command	Purpose

Step 4	**show lacp offload status**
	
	Example:
	`n1000v(config)# show lacp offload status`
	`  Current Status  : Disabled`
	`  Running Config Status : Enabled`
	`  Saved Config Status  : Enabled`
	`n1000v(config)#`
		(Optional) Displays the LACP offload status for verification.
		Note Current status does not change to enabled until after reload.

Step 5	**reload**
	
	Example:
	`n1000v(config)# reload`
	`This command will reboot the system.`
	`(y/n)?  [n] y`
	`2010 Sep  3 11:33:35 n1000v`
	`%PLATFORM-2-PFM_SYSTEM_RESET: Manual`
	`system restart from Command Line`
	`Interface`
		Reboots both the primary and secondary VSM.

Step 6	**show lacp offload status**
	
	Example:
	`n1000v# show lacp offload status`
	`  Current Status  : Enabled`
	`  Running Config Status : Enabled`
	`  Saved Config Status  : Enabled`
	`n1000v(config)#`
		(Optional) After system reload, displays the LACP offload status for verification.
		Note Current status should now show enabled.

Verifying Port Channels

Use the following commands to display the port channel configuration.

For more information about the command output, see the *Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(5.1)*.

Command	Purpose
show feature	Displays the features available, such as LACP, and whether they are enabled.
show interface port-channel *channel-number*	Displays the status of a port channel interface.
show lacp port-channel [**interface port-channel** *channel-number*]	Displays information about LACP port channels.
show lacp interface ethernet *slot/port*	Displays information about specific LACP interfaces.
show lacp offload status	Displays whether LACP management is offloaded to the VEMs. • **Enabled**: LACP is managed by VEMs. • **Disabled**: LACP is managed by the VSM.
show network-state tracking config	Displays the Network State Tracking configuration for verification.

Command	Purpose
show network-state tracking {**module** *modID* | **interface** *channelID*}	Displays the Network State Tracking status for a module or interface.
show port-channel compatibility-parameters	Displays the parameters that must be the same among the member ports in order to join a port channel.
show port-channel database [**interface port-channel** *channel-number*]	Displays the aggregation state for one or more port channel interfaces.
show port-channel load-balance	Displays the type of load balancing in use for port channels.
show port-channel summary	Displays a summary for the port channel interfaces.
show port-channel traffic	Displays the traffic statistics for port channels.
show port-channel usage	Displays the range of used and unused channel numbers.
show running-config interface ethernet *port/slot*	Displays information about the running configuration of the specified Ethernet interface.
show running-config interface port-channel *channel-number*	Displays information on the running configuration of the port channel.
show running-config interface vethernet *interface-number*	Displays information about the running configuration of the specified vEthernet interface.

Monitoring Port Channels

Use the following commands to monitor the port channel interface configuration.

Command	Purpose
clear counters interface port-channel *channel-number*	Clears the counters.
show interface counters [**module** *module*]	Displays input and output octets unicast packets, multicast packets, and broadcast packets.
show interface counters detailed [**all**]	Displays input packets, bytes, and multicast and output packets and bytes.
show interface counters errors [**module** *module*]	Displays information on the number of error packets.
show lacp counters [**interface port-channel** *channel-number*]	Displays information about LACP statistics.

Configuration Examples for Port Channels

This section includes the following examples:

- Configuration Example: Create a Port Channel and Add Interfaces, page 5-48

Configuration Example: Create a Port Channel and Add Interfaces

The following example shows how to create a port channel and add two Layer 2 interfaces to that port channel:

```
configure terminal
interface port-channel 5
interface ethernet 1/4
switchport
channel-group 5 mode active
interface ethernet 1/7
switchport
channel-group 5 mode
```

Configuration Example: Create an LACP Port Channel

The following example shows how to set the LACP-enabled interface to the active port channel mode for Ethernet interface 1/4 in channel group 5; and then configure an LACP port profile for the port channel.

```
configure terminal
feature lacp
interface ethernet 1/4
channel-group 5 mode active
port-profile type ethernet system-uplink
vmware port-group lacp
switchport mode trunk
switchport trunk allowed vlan 1-100
channel-group auto mode active
system vlan 1,10,20
state enabled
show port-channel summary
copy running-config startup-config
```

Configuration Example: Configuring Network State Tracking for vPC-HM

The following example shows how to configure Network State Tracking with an 8 second interval between sent broadcasts, a maximum of 7 missed broadcasts before declaring a split network, and repin traffic to another uplink if a split network is detected:

```
configure terminal
track network-state enable
track network-state interval 8
track network-state split action repin
track network-state threshold miss-count 7
show network-state tracking config
Tracking mode        : enabled
Tracking Interval    : 8 sec
Miss count threshold : 7 pkts
Split-network action : repin
n1000v(config)#
```

Additional References

For additional information related to implementing port channels, see the following sections:

- Related Documents, page 5-49
- Standards, page 5-49

Related Documents

Related Topic	Document Title
Complete command syntax, command modes, command history, defaults, usage guidelines, and examples for all Cisco Nexus 1000V commands.	*Cisco Nexus 1000V Command Reference, Release 4.2(1)SV1(5.1)*
Configuring Layer 2 interface	Chapter 3, "Configuring Layer 2 Interfaces"
System management	*Cisco Nexus 1000V System Management Configuration Guide, Release 4.2(1)SV1(5.1)*
Release Notes	*Cisco Nexus 1000V Release Notes, Release 4.2(1)SV1(5.1)*
Port Profiles	*Cisco Nexus 1000V Port Profile Configuration Guide, Release 4.2(1)SV1(5.1)*

Standards

Standards	Title
IEEE 802.3ad	Link Aggregation

Feature History for Port Channels

This section provides the feature history for port channels.

Feature Name	Releases	Feature Information
Backup subgroups	4.2(1)SV1(4a)	You can assign up to seven backup subgroups when pinning the primary subgroup.
Port channel relative numbering	4.2(1)SV1(4a)	The subgroup numbering begins at zero and is not tied to the vmnic number.
Port channel vPC-HM	4.2(1)SV1(4)	The interface **sub-group cdp** command is removed from port channel vPC-HM configuration when connecting to multiple upstream switches.
Network State Tracking for vPC-HM port channels	4.2(1)SV1(4)	Pinpoints link failure on a port channel configured for vPC-HM.
VEM management of LACP	4.2(1)SV1(4)	Offloading management of LACP from the VSM to the VEMs.

Feature Name	Releases	Feature Information
Enabling the LACP port channel function	4.2(1)SV1(4)	The command, **feature lacp**, is added to enable support of LACP port-channels. Previously LACP was enabled automatically.
vPC-Host Mode	4.0(4)SV1(2)	Support for manual creation of subgroups.
Static Pinning	4.0(4)SV1(2)	Support for attaching (or pinning) a vEthernet interface to a specific port channel subgroup.
Port Channels	4.0(4)SV1(1)	This feature was introduced.

APPENDIX 6

Supported RFCs

This section lists the supported IETF RFCs for interfaces.

IP Services RFCs

RFCs	Title
RFC 786	UDP
RFC 791	IP
RFC 792	ICMP
RFC 793	TCP
RFC 826	ARP
RFC 1027	Proxy ARP
RFC 1591	DNS Client
RFC 1812	IPv4 routers

APPENDIX 7

Interface Configuration Limits

Table 7-1 lists the configuration limits for interfaces.

Table 7-1 Interface Configuration Limits

Interface	Maximum per DVS	Maximum per Host
vEthernet interfaces	2048	216
vEthernet trunks	256	8
Port channels	256	8

INDEX

A

C

D

E

G

I

L

M

N

P

R

S

T

V