Cisco Nexus 3000 Series NX-OS Layer 2 Switching Configuration Guide, Release 5.0(3)U2(1)

First Published: 2011-06-14

Last Modified: 2011-12-13

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-26508 -01

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

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

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

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

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: http://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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

© 2011-2012 Cisco Systems, Inc. All rights reserved.

CONTENTS

Preface

Preface xiii

Audience xiii

Document Organization xiii

Document Conventions xiv

Related Documentation for Nexus 3000 Series NX-OS Software xv

Obtaining Documentation and Submitting a Service Request xvi

CHAPTER 1

New and Changed Information for this Release 1

New and Changed Information 1

CHAPTER 2

Overview 3

Layer 2 Ethernet Switching Overview 3

VLANs 3

Private VLANs 4

Spanning Tree 4

STP Overview 4

Rapid PVST+ 5

MST 5

STP Extensions 5

CHAPTER 3

Configuring Ethernet Interfaces 7

Information About Ethernet Interfaces 7

About the Interface Command 7

About the Unidirectional Link Detection Parameter 8

Default UDLD Configuration 9

UDLD Aggressive and Nonaggressive Modes 9

About Interface Speed 10

About the Cisco Discovery Protocol 10
Default CDP Configuration 10
About the Error-Disabled State 11
About Port Profiles 11
Guidelines and Limitations for Port Profiles 11
About the Debounce Timer Parameters 11
About MTU Configuration 11
Configuring Ethernet Interfaces 12
Configuring the UDLD Mode 12
Configuring Interface Speed 13
Disabling Link Negotiation 14
Configuring the CDP Characteristics 15
Enabling or Disabling CDP 16
Enabling the Error-Disabled Detection 17
Enabling the Error-Disabled Recovery 18
Configuring the Error-Disabled Recovery Interval 19
Configuring the Debounce Timer 20
Configuring the Description Parameter 20
Disabling and Restarting Ethernet Interfaces 21
Displaying Interface Information 21
Displaying Input Packet Discard Information 23
Default Physical Ethernet Settings 24

CHAPTER 4 Configuring VLANs 25
Information About VLANs 25
Understanding VLANs 25
Understanding VLAN Ranges 26
Creating, Deleting, and Modifying VLANs 27
Configuring a VLAN 28
Creating and Deleting a VLAN 28
Configuring a VLAN 29
Adding Ports to a VLAN 30
Verifying VLAN Configuration 31

CHAPTER 5 Configuring Private VLANs 33

Information About Private VLANs 33
Primary and Secondary VLANs in Private VLANs 34
Private VLAN Ports 34
Primary, Isolated, and Community Private VLANs 35
Associating Primary and Secondary VLANs 36
Private VLAN Promiscuous Trunks 37
Private VLAN Isolated Trunks 37
Broadcast Traffic in Private VLANs 37
Private VLAN Port Isolation 37
Guidelines and Limitations for Private VLANs 38
Configuring a Private VLAN 38
Enabling Private VLANs 38
Configuring a VLAN as a Private VLAN 39
Associating Secondary VLANs with a Primary Private VLAN 40
Configuring an Interface as a Private VLAN Host Port 41
Configuring an Interface as a Private VLAN Promiscuous Port 42
Configuring a Promiscuous Trunk Port 43
Configuring an Isolated Trunk Port 43
Configuring the Allowed VLANs for PVLAN Trunking Ports 43
Configuring Native 802.1Q VLANs on Private VLANs 44
Verifying Private VLAN Configuration 44

CHAPTER 6 **Configuring Access and Trunk Interfaces 45**
Information About Access and Trunk Interfaces 45
Understanding Access and Trunk Interfaces 45
Understanding IEEE 802.1Q Encapsulation 46
Understanding Access VLANs 47
Understanding the Native VLAN ID for Trunk Ports 48
Understanding Allowed VLANs 48
Understanding Native 802.1Q VLANs 48
Configuring Access and Trunk Interfaces 49
Configuring a LAN Interface as an Ethernet Access Port 49
Configuring Access Host Ports 50
Configuring Trunk Ports 50
Configuring the Native VLAN for 802.1Q Trunking Ports 51

Configuring the Allowed VLANs for Trunking Ports 52
Configuring Native 802.1Q VLANs 53
Verifying Interface Configuration 54

CHAPTER 7 Configuring Port Channels 55
Information About Port Channels 55
Understanding Port Channels 55
Compatibility Requirements 56
Load Balancing Using Port Channels 58
Understanding LACP 59
LACP Overview 59
LACP ID Parameters 60
Channel Modes 61
LACP Marker Responders 62
LACP-Enabled and Static Port Channel Differences 62
Configuring Port Channels 62
Creating a Port Channel 62
Adding a Port to a Port Channel 63
Configuring Load Balancing Using Port Channels 64
Configuring Hardware Hashing for Multicast Traffic 65
Enabling LACP 66
Configuring the Channel Mode for a Port 67
Configuring the LACP Fast Timer Rate 68
Configuring the LACP System Priority and System ID 69
Configuring the LACP Port Priority 69
Verifying Port Channel Configuration 70
Verifying the Load-Balancing Outgoing Port ID 71

CHAPTER 8 Configuring Virtual Port Channels 73
Information About vPCs 73
vPC Overview 73
Terminology 74
vPC Terminology 74
Supported vPC Topologies 75
Cisco Nexus 3000 Series Switch vPC Topology 75

vPC Domain 75
Peer-Keepalive Link and Messages 76
Compatibility Parameters for vPC Peer Links 76
Configuration Parameters That Must Be Identical 77
Configuration Parameters That Should Be Identical 78
Graceful Type-1 Check 78
Per-VLAN Consistency Check 79
vPC Auto-Recovery 79
vPC Peer Links 79
vPC Peer Link Overview 79
vPC Number 80
vPC Interactions with Other Features 81
vPC and LACP 81
vPC Peer Links and STP 81
CFSoE 82
Guidelines and Limitations for vPCs 82
Configuring vPCs 83
Enabling vPCs 83
Disabling vPCs 83
Creating a vPC Domain 84
Configuring a vPC Keepalive Link and Messages 85
Configuring a Keepalive Link When Using a Front-Panel 10-Gigabit Ethernet Port 87
Creating a vPC Peer Link 88
Checking the Configuration Compatibility 89
Enabling vPC Auto-Recovery 90
Configuring the Restore Time Delay 90
Excluding VLAN Interfaces From Shutdown When vPC Peer Link Fails 91
Configuring the VRF Name 92
Binding a VRF Instance to a vPC 93
Moving Other EtherChannels into a vPC 93
Manually Configuring a vPC Domain MAC Address 94
Manually Configuring the System Priority 95
Manually Configuring a vPC Peer Switch Role 96
Verifying the vPC Configuration 97
Viewing The Graceful Type-1 Check Status 98

Viewing A Global Type-1 Inconsistency 98
Viewing An Interface-Specific Type-1 Inconsistency 99
Viewing a Per-VLAN Consistency Status 101
vPC Default Settings 102

CHAPTER 9 Configuring Rapid PVST+ 105
Information About Rapid PVST+ 105
Understanding STP 105
STP Overview 105
Understanding How a Topology is Created 106
Understanding the Bridge ID 106
Bridge Priority Value 106
Extended System ID 107
STP MAC Address Allocation 107
Understanding BPDUs 108
Election of the Root Bridge 109
Creating the Spanning Tree Topology 109
Understanding Rapid PVST+ 110
Rapid PVST+ Overview 110
Rapid PVST+ BPDUs 111
Proposal and Agreement Handshake 112
Protocol Timers 113
Port Roles 113
Port States 114
Rapid PVST+ Port State Overview 114
Blocking State 115
Learning State 115
Forwarding State 116
Disabled State 116
Summary of Port States 116
Synchronization of Port Roles 117
Processing Superior BPDU Information 117
Processing Inferior BPDU Information 118
Spanning-Tree Dispute Mechanism 118
Port Cost 118

Port Priority 119
Rapid PVST+ and IEEE 802.1Q Trunks 119
Rapid PVST+ Interoperation with Legacy 802.1D STP 120
Rapid PVST+ Interoperation with 802.1s MST 120
Configuring Rapid PVST+ 120
Enabling Rapid PVST+ 121
Enabling Rapid PVST+ per VLAN 121
Configuring the Root Bridge ID 122
Configuring a Secondary Root Bridge 124
Configuring the Rapid PVST+ Port Priority 124
Configuring the Rapid PVST+ Pathcost Method and Port Cost 125
Configuring the Rapid PVST+ Bridge Priority of a VLAN 126
Configuring the Rapid PVST+ Hello Time for a VLAN 127
Configuring the Rapid PVST+ Forward Delay Time for a VLAN 128
Configuring the Rapid PVST+ Maximum Age Time for a VLAN 128
Specifying the Link Type 129
Restarting the Protocol 130
Verifying Rapid PVST+ Configurations 130

CHAPTER 10 **Configuring Multiple Spanning Tree** 131
Information About MST 131
MST Overview 131
MST Regions 132
MST BPDUs 132
MST Configuration Information 133
IST, CIST, and CST 133
IST, CIST, and CST Overview 133
Spanning Tree Operation Within an MST Region 134
Spanning Tree Operations Between MST Regions 134
MST Terminology 135
Hop Count 136
Boundary Ports 136
Spanning-Tree Dispute Mechanism 137
Port Cost and Port Priority 138
Interoperability with IEEE 802.1D 138

Interoperability with Rapid PVST+: Understanding PVST Simulation 139
Configuring MST 139
MST Configuration Guidelines 139
Enabling MST 139
Entering MST Configuration Mode 140
Specifying the MST Name 141
Specifying the MST Configuration Revision Number 142
Specifying the Configuration on an MST Region 143
Mapping and Unmapping VLANs to MST Instances 144
Mapping Secondary VLANs to Same MSTI as Primary VLANs for Private VLANs 145
Configuring the Root Bridge 146
Configuring a Secondary Root Bridge 147
Configuring the Port Priority 148
Configuring the Port Cost 149
Configuring the Switch Priority 150
Configuring the Hello Time 151
Configuring the Forwarding-Delay Time 152
Configuring the Maximum-Aging Time 152
Configuring the Maximum-Hop Count 153
Configuring PVST Simulation Globally 154
Configuring PVST Simulation Per Port 154
Specifying the Link Type 155
Restarting the Protocol 156
Verifying MST Configurations 157

CHAPTER 11 **Configuring STP Extensions** 159
About STP Extensions 159
Information About STP Extensions 159
Understanding STP Port Types 159
Spanning Tree Edge Ports 159
Spanning Tree Network Ports 160
Spanning Tree Normal Ports 160
Understanding Bridge Assurance 160
Understanding BPDU Guard 160
Understanding BPDU Filtering 161

Understanding Loop Guard 162
Understanding Root Guard 162
Configuring STP Extensions 163
STP Extensions Configuration Guidelines 163
Configuring Spanning Tree Port Types Globally 163
Configuring Spanning Tree Edge Ports on Specified Interfaces 164
Configuring Spanning Tree Network Ports on Specified Interfaces 165
Enabling BPDU Guard Globally 167
Enabling BPDU Guard on Specified Interfaces 167
Enabling BPDU Filtering Globally 168
Enabling BPDU Filtering on Specified Interfaces 169
Enabling Loop Guard Globally 170
Enabling Loop Guard or Root Guard on Specified Interfaces 171
Verifying STP Extension Configuration 172

CHAPTER 12 **Configuring LLDP 173**
Configuring Global LLDP Commands 173
Configuring Interface LLDP Commands 175

CHAPTER 13 **Configuring the MAC Address Table 179**
Information About MAC Addresses 179
Configuring MAC Addresses 179
Configuring a Static MAC Address 179
Configuring the Aging Time for the MAC Table 180
Clearing Dynamic Addresses from the MAC Table 181
Verifying the MAC Address Configuration 181

CHAPTER 14 **Configuring IGMP Snooping 183**
Information About IGMP Snooping 183
IGMPv1 and IGMPv2 184
IGMPv3 185
IGMP Snooping Querier 185
IGMP Forwarding 185
Configuring IGMP Snooping Parameters 186
Verifying IGMP Snooping Configuration 188

CHAPTER 15 **Configuring Traffic Storm Control** 191

Information About Traffic Storm Control 191

Traffic Storm Guidelines and Limitations 192

Configuring Traffic Storm Control 193

Verifying Traffic Storm Control Configuration 194

Traffic Storm Control Example Configuration 194

Default Traffic Storm Settings 194

Preface

This preface describes the audience, organization, and conventions of the *Cisco Nexus 3000 Series NX-OS Layer 2 Switching Configuration Guide*. It also provides information on how to obtain related documentation.

- Audience, page xiii
- Document Organization, page xiii
- Document Conventions, page xiv
- Related Documentation for Nexus 3000 Series NX-OS Software, page xv
- Obtaining Documentation and Submitting a Service Request, page xvi

Audience

This publication is for experienced network administrators who configure and maintain Cisco NX-OS software.

Document Organization

This document is organized into the following chapters:

Chapter	Description
Overview	Describes the Layer 2 documented features.
Configuring Ethernet Interfaces	Provides information about Ethernet interfaces and describes configuration procedures.
Configuring VLANs	Provides details on configuring VLANs.
Configuring Private VLANs	Provides information on configuring private VLANs.
Configuring Access and Trunk Interfaces	Provides information about access ports or trunk ports and describes configuration procedures.
Configuring EtherChannels	Provides information about EtherChannels, compatibility requirements, and configuration information.

Chapter	Description
Configuring Rapid PVST+	Provides information on IEEE 802.1D STP and complete details for configuring Rapid PVST+.
Configuring Multiple Spanning Tree	Provides complete information on configuring MST.
Configuring STP Extensions	Provides details on configuring the Cisco-proprietary STP extensions Bridge Assurance, BPDU Guard, BPDU Filtering, Loop Guard, Root Guard, and PVST Simulation.
Configuring Link Layer Discovery Protocol	Provides information to configure the Link Layer Discovery Protocol (LLDP).
Configuring the MAC Address Table	Provides information about MAC addresses, how to configure static MAC addresses, and how to update the MAC address table.
Configuring IGMP Snooping	Provides information about IGMPv1, IGMPv2, and IGMPv3 and describes how to configure IGMP snooping parameters.
Configuring Traffic Storm Control	Provides information about traffic storm control, guidelines and limitations, and how to configure the traffic storm control settings.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element(keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.

Convention	Description
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

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

This document uses the following conventions:

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

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

Related Documentation for Nexus 3000 Series NX-OS Software

Cisco NX-OS documentation is available at the following URL:

http://www.cisco.com/en/US/products/ps11541/tsd_products_support_series_home.html

The documentation set for the Cisco Nexus 3000 Series NX-OS software includes the following documents:

Release Notes

- *Cisco Nexus 3000 Series Switch Release Notes*

Cisco NX-OS documentation is available at the following URL:

http://www.cisco.com/en/US/products/ps9670/tsd_products_support_series_home.htmll

The documentation set for the Cisco Nexus 3000 Series NX-OS software includes the following documents:

Configuration Guides

- *Cisco Nexus 3000 Series NX-OS Configuration Limits*
- *Cisco Nexus 3000 Series NX-OS Layer 2 Switching Configuration Guide*
- *Cisco Nexus 3000 Series NX-OS Fundamentals Configuration Guide*
- *Cisco Nexus 3000 Series NX-OS System Management Configuration Guide*
- *Cisco Nexus 3000 Series NX-OS Unicast Routing Configuration Guide*
- *Cisco Nexus 3000 Series NX-OS Multicast Configuration Guide*

Installation and Upgrade Guides

- *Cisco Nexus 3000 Series Hardware Installation Guide*
- *Regulatory Compliance and Safety Information for the Cisco Nexus 3000 Series Switches*

Licensing Guide

- *Cisco NX-OS Licensing Guide*

Command References

- *Cisco Nexus 3000 Series Command Reference*

Technical References

- *Cisco Nexus 3000 Series MIBs Reference*

Error and System Messages

- *Cisco NX-OS System Messages Reference*

Troubleshooting Guide

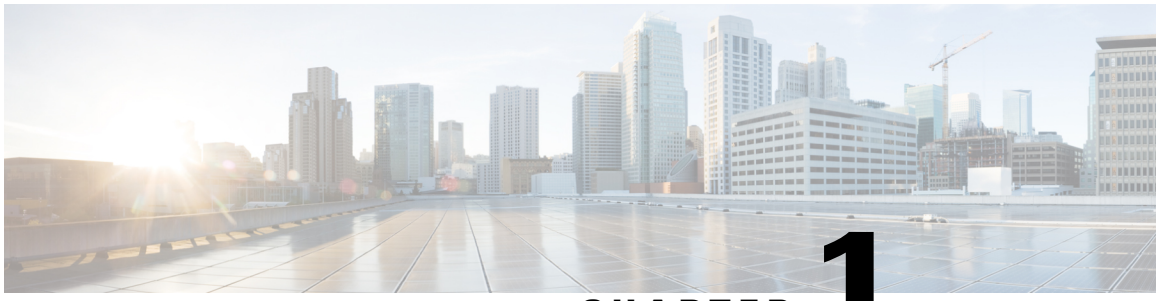
- *Cisco Nexus 3000 Series Troubleshooting Guide*

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

New and Changed Information for this Release

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

- New and Changed Information, page 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 3000 Series NX-OS Layer 2 Switching Configuration Guide*.

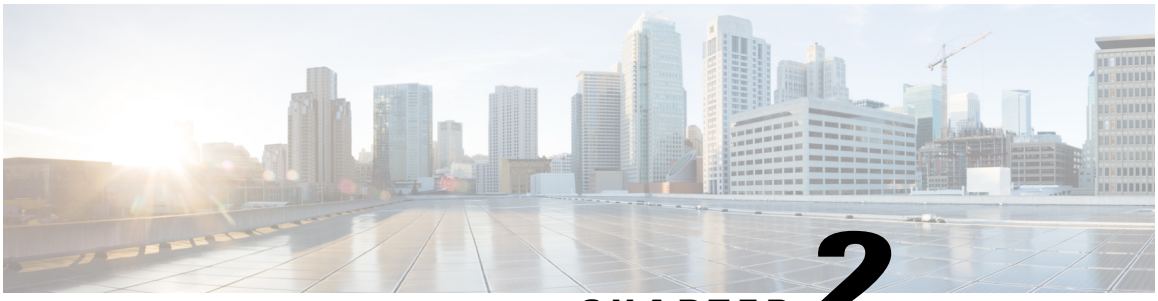
The latest version of this document is available at the following Cisco website:
http://www.cisco.com/en/US/products/ps11541/products_installation_and_configuration_guides_list.html.

To check for the latest information about Cisco NX-OS for the Cisco Nexus 3000 Series switch, see the *Cisco Nexus 3000 Series Release Notes* available at the following Cisco website:
http://www.cisco.com/en/US/products/ps11541/prod_release_notes_list.html

This table summarizes the new and changed features documented in the *Cisco Nexus 3000 Series NX-OS Layer 2 Switching Configuration Guide, Release 5.0(3)U2(1)*, and tells you where they are documented.

Table 1: New and Changed Layer 2 Switching Features for Cisco NX-OS Release 5.0(3)U2(1)

Feature	Description	Changed in Release	Where Documented
Link Negotiation	Added information to change status of link negotiation to disabled.	5.0(3)U2(1)	Disabling Link Negotiation
Virtual Port Channels	Added information to configure virtual port channels (vPC).	5.0(3)U2(1)	Configuring Virtual Port Channels



CHAPTER 2

Overview

This chapter contains the following sections:

- Layer 2 Ethernet Switching Overview, page 3
- VLANs, page 3
- Private VLANs, page 4
- Spanning Tree , page 4

Layer 2 Ethernet Switching Overview

The device supports simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

The device solves congestion problems caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own 10-, 100-, 1000-Mbps, or 10-Gigabit collision domain. Because each LAN port connects to a separate Ethernet collision domain, servers in a switched environment achieve full access to the bandwidth.

Because collisions cause significant congestion in Ethernet networks, an effective solution is full-duplex communication. Typically, 10/100-Mbps Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, which is configurable on these interfaces, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, the effective Ethernet bandwidth doubles. 1/10-Gigabit Ethernet operates in full-duplex only.

VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered as a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a bridge or a router.

All ports, including the management port, are assigned to the default VLAN (VLAN1) when the device first comes up. A VLAN interface, or switched virtual interface (SVI), is a Layer 3 interface that is created to provide communication between VLANs.

The devices support 4094 VLANs in accordance with the IEEE 802.1Q standard. These VLANs are organized into several ranges, and you use each range slightly differently. Some of these VLANs are reserved for internal use by the device and are not available for configuration.

Note Inter-Switch Link (ISL) trunking is not supported on the NX-OS software for the Cisco Nexus 3000 Series.

Private VLANs

Private VLANs provide traffic separation and security at the Layer 2 level.

A private VLAN is one or more pairs of a primary VLAN and a secondary VLAN, all with the same primary VLAN. The two types of secondary VLANs are isolated and community VLANs. Hosts on isolated VLANs communicate only with hosts in the primary VLAN. Hosts in a community VLAN can communicate only among themselves and with hosts in the primary VLAN but not with hosts in isolated VLANs or in other community VLANs.

Regardless of the combination of isolated and community secondary VLANs, all interfaces within the primary VLAN comprise one Layer 2 domain, and therefore, require only one IP subnet.

Spanning Tree

This section discusses the implementation of the Spanning Tree Protocol (STP). Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. When the IEEE 802.1D Spanning Tree Protocol is referred to in the publication, 802.1D is stated specifically.

STP Overview

STP provides a loop-free network at the Layer 2 level. Layer 2 LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Network devices do not forward these frames but use the frames to construct a loop-free path.

802.1D is the original standard for STP, and many improvements have enhanced the basic loop-free STP. You can create a separate loop-free path for each VLAN, which is named Per VLAN Spanning Tree (PVST+). Additionally, the entire standard was reworked to make the loop-free convergence process faster to keep up with the faster equipment. This STP standard with faster convergence is the 802.1w standard, which is known as Rapid Spanning Tree (RSTP).

Finally, the 802.1s standard, Multiple Spanning Trees (MST), allows you to map multiple VLANs into a single spanning tree instance. Each instance runs an independent spanning tree topology.

Although the software can interoperate with legacy 802.1D systems, the system runs Rapid PVST+ and MST. You can use either Rapid PVST+ or MST in a given VDC; you cannot mix both in one VDC. Rapid PVST+ is the default STP protocol for Cisco NX-OS for the Cisco Nexus 3000 Series.

Note Cisco NX-OS for the Cisco Nexus 3000 Series uses the extended system ID and MAC address reduction; you cannot disable these features.

In addition, Cisco has created some proprietary features to enhance the spanning tree activities.

Rapid PVST+

Rapid PVST+ is the default spanning tree mode for the software and is enabled by default on the default VLAN and all newly created VLANs.

A single instance, or topology, of RSTP runs on each configured VLAN, and each Rapid PVST+ instance on a VLAN has a single root device. You can enable and disable STP on a per-VLAN basis when you are running Rapid PVST+.

MST

The software also supports MST. The multiple independent spanning tree topologies enabled by MST provide multiple forwarding paths for data traffic, enable load balancing, and reduce the number of STP instances required to support a large number of VLANs.

MST incorporates RSTP, so it also allows rapid convergence. MST improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

Note Changing the spanning tree mode disrupts the traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

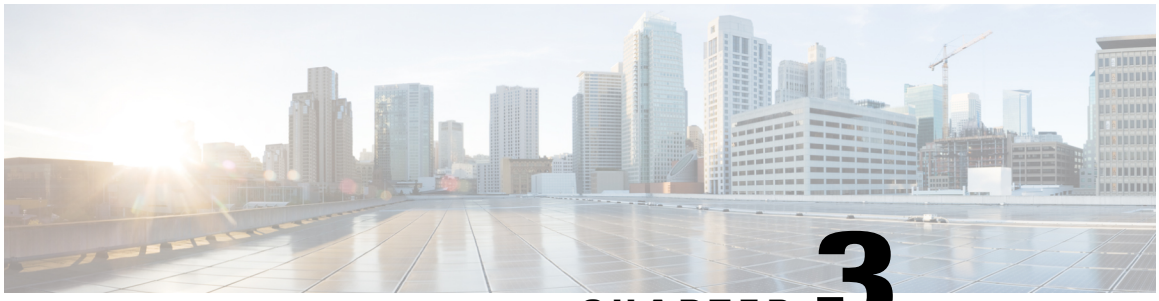
You can force specified interfaces to send prestandard, rather than standard, MST messages using the command-line interface.

STP Extensions

The software supports the following Cisco proprietary features:

- Spanning tree port types—The default spanning tree port type is normal. You can configure interfaces connected to Layer 2 hosts as edge ports and interfaces connected to Layer 2 switches or bridges as network ports.
- Bridge Assurance—Once you configure a port as a network port, Bridge Assurance sends BPDUs on all ports and moves a port into the blocking state if it no longer receives BPDUs. This enhancement is available only when you are running Rapid PVST+ or MST.
- BPDU Guard—BPDU Guard shuts down the port if that port receives a BPDU.
- BPDU Filter—BPDU Filter suppresses sending and receiving BPDUs on the port.
- Loop Guard—Loop Guard prevents the nondesignated ports from transitioning to the STP forwarding state, which prevents loops in the network.

- Root Guard—Root Guard prevents the port from becoming the root in an STP topology.



CHAPTER 3

Configuring Ethernet Interfaces

This chapter contains the following sections:

- Information About Ethernet Interfaces, page 7
- Configuring Ethernet Interfaces, page 12
- Displaying Interface Information, page 21
- Displaying Input Packet Discard Information, page 23
- Default Physical Ethernet Settings , page 24

Information About Ethernet Interfaces

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

On a Cisco Nexus 3000 Series switch, the Ethernet interfaces are enabled by default.

About the Interface Command

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

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

The interface numbering convention is extended to support use with a Cisco Nexus 2000 Series Fabric Extender as follows:

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

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

About the Unidirectional Link Detection Parameter

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

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

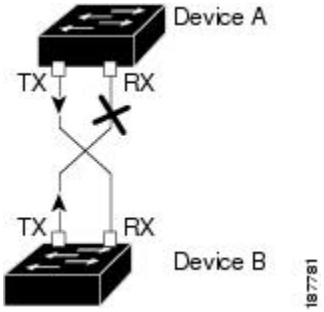
A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does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

A Cisco Nexus 3000 Series switch periodically transmits UDLD frames to neighbor devices on LAN ports with UDLD enabled. If the frames are echoed back within a specific time frame and they lack a specific acknowledgment (echo), the link is flagged as unidirectional and the LAN port is shut down. Devices on both ends of the link must support UDLD in order for the protocol to successfully identify and disable unidirectional links.

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

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

Figure 1: Unidirectional Link



Default UDLD Configuration

The following table shows the default UDLD configuration.

Table 2: UDLD Default Configuration

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

Related Topics

Configuring the UDLD Mode, on page 12

UDLD Aggressive and Nonaggressive Modes

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

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

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

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

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

About Interface Speed

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

About the Cisco Discovery Protocol

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

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

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

The switch supports both CDP Version 1 and Version 2.

Default CDP Configuration

The following table shows the default CDP configuration.

Table 3: Default CDP Configuration

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

About the Error-Disabled State

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

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

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

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

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

About Port Profiles

The Cisco Nexus 3000 Series device does not support Port Profiles.

Guidelines and Limitations for Port Profiles

The Cisco Nexus 3000 Series device does not support Port Profiles.

About the Debounce Timer Parameters

The debounce timer feature is not supported on Nexus 3000.

About MTU Configuration

The Cisco Nexus 3000 Series switch does not fragment frames. As a result, the switch cannot have two ports in the same Layer 2 domain with different maximum transmission units (MTUs). A per-physical Ethernet interface MTU is not supported. Instead, the MTU is set according to the QoS classes. You modify the MTU by setting Class and Policy maps.

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

Configuring Ethernet Interfaces

The section includes the following topics:

Configuring the UDLD Mode

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

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

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

To configure the UDLD mode, perform this task:

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature udld**
3. switch(config)# **no feature udld**
4. switch(config)# **show udld global**
5. switch(config)# **interface** *type slot*/*port*
6. switch(config-if)# **udld** {**enable** | **disable** | **aggressive**}
7. switch(config-if)# **show udld** *interface*

DETAILED STEPS

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

	Command or Action	Purpose
Step 7	switch(config-if)# **show udld** *interface*	Displays the UDLD status for the interface.

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

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

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

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

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

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

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

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

This example shows how to disable UDLD for the switch:

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

Configuring Interface Speed

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **speed** *speed*

DETAILED STEPS

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

	Command or Action	Purpose

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

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

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

Disabling Link Negotiation

You can disable link negotiation using the **no negotiate auto** command. Use the **negotiate auto** command to enable auto negotiation on 1-Gigabit ports when the connected peer does not support auto negotiation. By default, auto-negotiation is enabled on 1-Gigabit ports and disabled on 10-Gigabit ports.

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

Note Cisco does not recommend that you to enable auto negotiation on 10-Gigabit ports. Enabling auto-negotiation on 10-Gigabit ports brings the link down. By default, link negotiation is disabled on 10-Gigabit ports.

SUMMARY STEPS

1. switch# configure terminal
2. switch(config)# interface ethernet slot/port
3. switch(config-if)# no negotiate auto
4. (Optional) switch(config-if)# negotiate auto

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# configure terminal	Enters configuration mode.
Step 2	switch(config)# interface ethernet slot/port	Selects the interface and enters interface mode.
Step 3	switch(config-if)# no negotiate auto	Disables link negotiation on the selected Ethernet interface (1-Gigabit port).

	Command or Action	Purpose
Step 4	switch(config-if)# negotiate auto	(Optional) Enables link negotiation on the selected Ethernet interface. The default for 1-Gigabit ports is enabled.

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

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

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

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

Configuring the CDP Characteristics

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

To configure CDP characteristics for an interface, perform this task:

SUMMARY STEPS

1. switch# **configure terminal**
2. (Optional) switch(config)# [**no**] **cdp advertise** {**v1** | **v2** }
3. (Optional) switch(config)# [**no**] **cdp format device-id** {**mac-address** | **serial-number** | **system-name**}
4. (Optional) switch(config)# [**no**] **cdp holdtime** *seconds*
5. (Optional) switch(config)# [**no**] **cdp timer** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# [**no**] **cdp advertise** {**v1** | **v2** }	(Optional) Configures the version to use to send CDP advertisements. Version-2 is the default state. Use the **no** form of the command to return to its default setting.
Step 3	switch(config)# [**no**] **cdp format device-id** {**mac-address** | **serial-number** | **system-name**}	(Optional) Configures the format of the CDP device ID. The default is the system name, which can be expressed as a fully qualified domain name. Use the **no** form of the command to return to its default setting.

	Command or Action	Purpose
Step 4	switch(config)# [**no**] **cdp holdtime** *seconds*	(Optional) Specifies the amount of time a receiving device should hold the information sent by your device before discarding it. The range is 10 to 255 seconds; the default is 180 seconds. Use the **no** form of the command to return to its default setting.
Step 5	switch(config)# [**no**] **cdp timer** *seconds*	(Optional) Sets the transmission frequency of CDP updates in seconds. The range is 5 to 254; the default is 60 seconds. Use the **no** form of the command to return to its default setting.

This example shows how to configure CDP characteristics:

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

Enabling or Disabling CDP

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

To enable or disable CDP for an interface, perform this task:

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **cdp enable**
4. switch(config-if)# **no cdp enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **cdp enable**	Enables CDP for the interface. To work correctly, this parameter must be enabled for both interfaces on the same link.

	Command or Action	Purpose
Step 4	switch(config-if)# **no cdp enable**	Disables CDP for the interface.

The following example shows how to enable CDP for an Ethernet port:

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

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

Enabling the Error-Disabled Detection

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

SUMMARY STEPS

1. **config t**
2. **errdisable detect cause** {*all* | *link-flap* | *loopback*}
3. **shutdown**
4. **no shutdown**
5. **show interface status err-disabled**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**errdisable detect cause** {*all* | *link-flap* | *loopback*}
	
	Example:
	`switch(config)# errdisable detect cause all`
	`switch(config)#`
		Specifies a condition under which to place the interface in an err-disabled state. The default is enabled.

Step 3	**shutdown**
	
	Example:
	`switch(config)# shutdown`
	`switch(config)#`
		Brings the interface down administratively. To manually recover the interface from the err-disabled state, enter this command first.

	Command or Action	Purpose

Step 4	**no shutdown**
	
	Example:
	`switch(config)# no shutdown`
	`switch(config)#`
		Brings the interface up administratively and enables the interface to recover manually from the err-disabled state.

Step 5	**show interface status err-disabled** **Example:** `switch(config)# show interface status err-disabled`	Displays information about err-disabled interfaces.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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

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

Enabling the Error-Disabled Recovery

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

SUMMARY STEPS

1. **config t**
2. **errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit*}
3. **show interface status err-disabled**
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit*}	Specifies a condition under which the interface automatically recovers from the err-disabled state, and

	Command or Action	Purpose

	Example:
	`switch(config)#errdisable recovery cause all`
	`switch(config-if)#`
		the device retries bringing the interface up. The device waits 300 seconds to retry. The default is disabled.

Step 3	**show interface status err-disabled** **Example:** `switch(config)#show interface status err-disabled`	Displays information about err-disabled interfaces.
Step 4	**copy running-config startup-config** **Example:** `switch(config)#copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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

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

Configuring the Error-Disabled Recovery Interval

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

SUMMARY STEPS

1. **config t**
2. **errdisable recovery interval** *interval*
3. **show interface status err-disabled**
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**errdisable recovery interval** *interval*
	Example:
	`switch(config)#errdisable recovery interval 32`
	`switch(config-if)#`
		Specifies the interval for the interface to recover from the err-disabled state. The range is from 30 to 65535 seconds. The default is 300 seconds.

	Command or Action	Purpose
Step 3	**show interface status err-disabled** **Example:** `switch(config)#show interface status err-disabled`	Displays information about err-disabled interfaces.
Step 4	**copy running-config startup-config** **Example:** `switch(config)#copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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

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

Configuring the Debounce Timer

This feature is not supported on the Nexus 3000 product.

Configuring the Description Parameter

To provide textual interface descriptions for the Ethernet ports, perform this task:

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **description** *test*

DETAILED STEPS

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

This example shows how to set the interface description to "Server 3 Interface."

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

Disabling and Restarting Ethernet Interfaces

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

To disable an interface, perform this task:

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **shutdown**
4. switch(config-if)# **no shutdown**

DETAILED STEPS

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

The following example shows how to disable an Ethernet port:

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

The following example shows how to restart an Ethernet interface:

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

Displaying Interface Information

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

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

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

The following example shows how to display the physical Ethernet interface:

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

The following example shows how to display the physical Ethernet capabilities:

```
switch# show interface ethernet 1/1 capabilities
Ethernet1/1
  Model:                734510033
  Type:                 10Gbase-(unknown)
  Speed:                1000,10000
  Duplex:               full
```

```
  Trunk encap. type:     802.1Q
  Channel:               yes
  Broadcast suppression: percentage(0-100)
  Flowcontrol:           rx-(off/on),tx-(off/on)
  Rate mode:             none
  QOS scheduling:        rx-(6q1t),tx-(1p6q0t)
  CoS rewrite:           no
  ToS rewrite:           no
  SPAN:                  yes
  UDLD:                  yes

  MDIX:                  no
  FEX Fabric:            yes
```

The following example shows how to display the physical Ethernet transceiver:

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

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

```
switch# show interface brief

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

The following example shows how to display the CDP neighbors:

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

Displaying Input Packet Discard Information

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

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

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

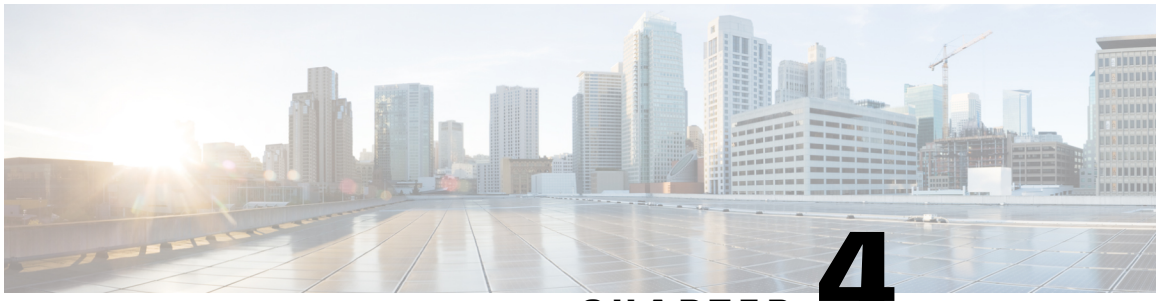
```
+---------------------------------------+-----------------+----------------+
|       Counter Description             |     Count       |                |
+---------------------------------------+-----------------+----------------+
        IPv4 Discards                             0
        STP Discards                              0
        Policy Discards                           0
        ACL Drops                                 0
        Receive Drops                             0
        Vlan Discards                             0
+---------------------------------------+-----------------+----------------+
```

Default Physical Ethernet Settings

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

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

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



CHAPTER 4

Configuring VLANs

This chapter contains the following sections:

- Information About VLANs, page 25
- Configuring a VLAN, page 28

Information About VLANs

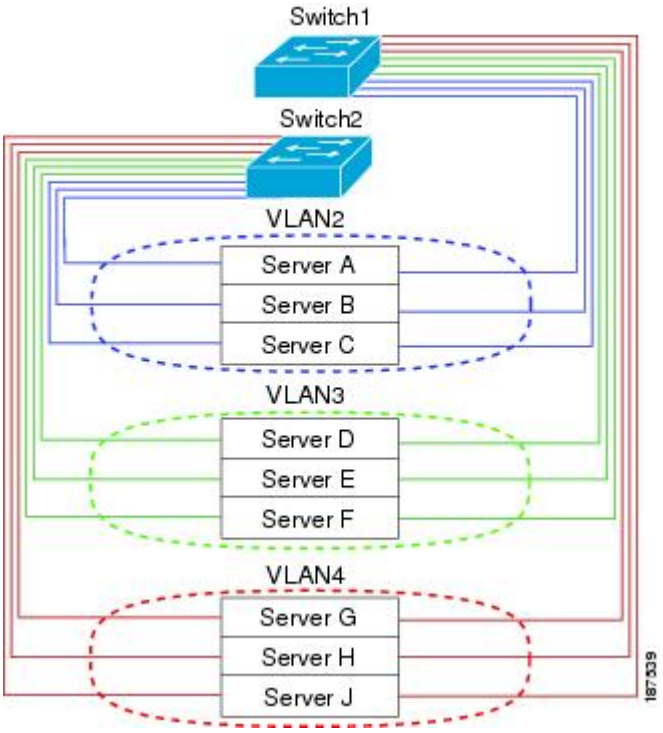
Understanding VLANs

A VLAN is a group of end stations in a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered a logical network. Packets destined for stations that do not belong to the VLAN must be forwarded through a router.

The following figure shows VLANs as logical networks. In this diagram, the stations in the engineering department are assigned to one VLAN, the stations in the marketing department are assigned to another VLAN, and the stations in the accounting department are assigned to yet another VLAN.

Figure 2: VLANs as Logically Defined Networks



VLANs are usually associated with IP subnetworks. For example, all the end stations in a particular IP subnet belong to the same VLAN. To communicate between VLANs, you must route the traffic.

By default, a newly created VLAN is operational. To disable the VLAN use the **shutdown** command. Additionally, you can configure VLANs to be in the active state, which is passing traffic, or the suspended state, in which the VLANs are not passing packets. By default, the VLANs are in the active state and pass traffic.

Understanding VLAN Ranges

The Cisco Nexus 3000 Series switch supports VLAN numbers 1to 4094 in accordance with the IEEE 802.1Q standard. These VLANs are organized into ranges. The switch is physically limited in the number of VLANs it can support. For information about VLAN configuration limits, see the configuration limits documentation for your switch.

The following table describes the details of the VLAN ranges:

Table 4: VLAN Ranges

VLANs Numbers	Range	Usage
1	Normal	Cisco default. You can use this VLAN, but you cannot modify or delete it.
2—1005	Normal	You can create, use, modify, and delete these VLANs.
1006—4094	Extended	You can create, name, and use these VLANs. You cannot change the following parameters: • State is always active. • VLAN is always enabled. You cannot shut down these VLANs.
3968—4047 and 4094	Internally allocated	These 80 VLANs, plus VLAN 4094, are allocated for internal use. You cannot create, delete, or modify any VLANs within the block reserved for internal use.

Note VLANs 3968 to 4047 and 4094 are reserved for internal use; these VLANs cannot be changed or used.

Cisco NX-OS allocates a group of 80 VLAN numbers for those features, such as multicast and diagnostics, that need to use internal VLANs for their operation. By default, the system allocates VLANs numbered 3968 to 4047 for internal use. VLAN 4094 is also reserved for internal use by the switch.

You cannot use, modify, or delete any of the VLANs in the reserved group. You can display the VLANs that are allocated internally and their associated use.

Creating, Deleting, and Modifying VLANs

VLANs are numbered from 1 to 4094. All configured ports belong to the default VLAN when you first bring up the switch. The default VLAN (VLAN1) uses only default values. You cannot create, delete, or suspend activity in the default VLAN.

You create a VLAN by assigning a number to it. You can delete VLANs as well as move them from the active operational state to the suspended operational state. If you attempt to create a VLAN with an existing VLAN ID, the switch goes into the VLAN submode but does not create the same VLAN again.

Newly created VLANs remain unused until ports are assigned to the specific VLAN. All the ports are assigned to VLAN1 by default.

Depending on the range of the VLAN, you can configure the following parameters for VLANs (except the default VLAN):

- VLAN name
- Shutdown or not shutdown

When you delete a specified VLAN, the ports associated to that VLAN are shut down and no traffic flows. However, the system retains all the VLAN-to-port mapping for that VLAN, and when you reenable, or recreate, the specified VLAN, the system automatically reinstates all the original ports to that VLAN.

Note Commands entered in the VLAN configuration submode are immediately executed.

VLANs 3968 to 4047 and 4094 are reserved for internal use; these VLANs cannot be changed or used.

Configuring a VLAN

Creating and Deleting a VLAN

You can create or delete all VLANs except the default VLAN and those VLANs that are internally allocated for use by the switch. Once a VLAN is created, it is automatically in the active state.

Note When you delete a VLAN, ports associated to that VLAN shut down. The traffic does not flow and the packets are dropped.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vlan** {*vlan-id* | *vlan-range*}
3. switch(config-vlan)# **no vlan** {*vlan-id* | *vlan-range*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vlan** {*vlan-id* | *vlan-range*}	Creates a VLAN or a range of VLANs. If you enter a number that is already assigned to a VLAN, the switch puts you into the VLAN configuration submode for that VLAN. If you enter a number that is assigned to an internally allocated VLAN, the system returns an error message. However, if you enter a range of VLANs and one or more of the specified VLANs is outside the range of internally allocated VLANs, the command takes effect on *only* those VLANs outside the range. The range is from 2 to 4094; VLAN1 is the

	Command or Action	Purpose
		default VLAN and cannot be created or deleted. You cannot create or delete those VLANs that are reserved for internal use.
Step 3	switch(config-vlan)# **no vlan** {*vlan-id* | *vlan-range*}	Deletes the specified VLAN or range of VLANs and removes you from the VLAN configuration submode. You cannot delete VLAN1 or the internally allocated VLANs.

This example shows how to create a range of VLANs from 15 to 20:

```
switch# configure terminal
switch(config)# vlan 15-20
```

Note You can also create and delete VLANs in the VLAN configuration submode.

Configuring a VLAN

To configure or modify the VLAN for the following parameters, you must be in the VLAN configuration submode:

- Name
- Shut down

Note You cannot create, delete, or modify the default VLAN or the internally allocated VLANs. Additionally, some of these parameters cannot be modified on some VLANs.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vlan** {*vlan-id* | *vlan-range*}
3. switch(config-vlan)# **name** *vlan-name*
4. switch(config-vlan)# **state** {**active** | **suspend**}
5. (Optional) switch(config-vlan)# **no shutdown**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	switch(config)# **vlan** {*vlan-id* | *vlan-range*}	Enters VLAN configuration submode. If the VLAN does not exist, the system first creates the specified VLAN.
Step 3	switch(config-vlan)# **name** *vlan-name*	Names the VLAN. You can enter up to 32 alphanumeric characters to name the VLAN. You cannot change the name of VLAN1 or the internally allocated VLANs. The default value is VLANxxxx where xxxx represent four numeric digits (including leading zeroes) equal to the VLAN ID number.
Step 4	switch(config-vlan)# **state** {**active** | **suspend**}	Sets the state of the VLAN to active or suspend. While the VLAN state is suspended, the ports associated with this VLAN are shut down, and that VLAN does not pass any traffic. The default state is active. You cannot suspend the state for the default VLAN or VLANs 1006 to 4094.
Step 5	switch(config-vlan)# **no shutdown**	(Optional) Enables the VLAN. The default value is **no shutdown** (or enabled). You cannot shut down the default VLAN, VLAN1, or VLANs 1006 to 4094.

This example shows how to configure optional parameters for VLAN 5:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# name accounting
switch(config-vlan)# state active
switch(config-vlan)# no shutdown
```

Adding Ports to a VLAN

After you have completed the configuration of a VLAN, assign ports to it. To add ports, perform this task:

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {**ethernet** *slot*/*port* | **port-channel** *number*}
3. switch(config-if)# **switchport access vlan** *vlan-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** {**ethernet** *slot*/*port* | **port-channel** *number*}	Specifies the interface to configure, and enters the interface configuration mode. The interface can be a physical Ethernet port or an EtherChannel.

	Command or Action	Purpose
Step 3	switch(config-if)# **switchport access vlan** *vlan-id*	Sets the access mode of the interface to the specified VLAN.

This example shows how to configure an Ethernet interface to join VLAN 5:

```
switch# configure terminal
switch(config)# interface ethernet 1/13
switch(config-if)# switchport access vlan 5
```

Verifying VLAN Configuration

To display VLAN configuration information, perform one of these tasks:

Command	Purpose
switch# **show running-config vlan** [*vlan_id* | *vlan_range*]	Displays VLAN information.
switch# **show vlan** [**brief** | **id** [*vlan_id* | *vlan_range*] | **name** *name* | **summary**]	Displays selected configuration information for the defined VLAN(s).

The following example shows all VLANs defined in the range of 1 to 21.

```
switch# show running-config vlan 1-21


version 5.0(3)U1(1)
vlan 1
vlan 5
```

The following example shows the VLANs created on the switch and their status:

```
switch# show vlan

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Eth1/1, Eth1/2, Eth1/3, Eth1/4
                                                 Eth1/5, Eth1/6, Eth1/7, Eth1/8
                                                 Eth1/9, Eth1/10, Eth1/11
                                                 Eth1/12, Eth1/15, Eth1/16
                                                 Eth1/17, Eth1/18, Eth1/19
                                                 Eth1/20, Eth1/21, Eth1/22
                                                 Eth1/23, Eth1/24, Eth1/25
                                                 Eth1/26, Eth1/27, Eth1/28
                                                 Eth1/29, Eth1/30, Eth1/31
                                                 Eth1/32, Eth1/33, Eth1/34
                                                 Eth1/35, Eth1/36, Eth1/37
                                                 Eth1/38, Eth1/39, Eth1/40
                                                 Eth3/1, Eth3/2, Eth3/3, Eth3/4

13   VLAN0005                          active    Eth1/13, Eth1/14
```

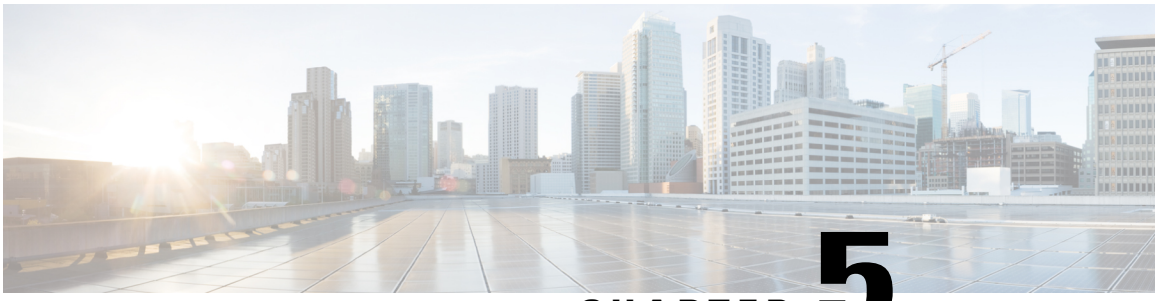
The following example shows the details of VLAN 13 including its member ports:

```
switch# show vlan id 13

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
13   VLAN0005                          active    Eth1/13, Eth1/14
```

```
VLAN Type   MTU
---- ----- -----
13   enet  576

Remote SPAN VLAN
----------------
Disabled

Primary  Secondary  Type             Ports
-------  ---------  ---------------  ------------------------------------------
```

The following example shows the VLAN settings summary:

```
switch# show vlan summary

 Number of existing VLANs           : 2
 Number of existing user VLANs      : 2
 Number of existing extended VLANs : 0
```



CHAPTER 5

Configuring Private VLANs

This chapter contains the following sections:

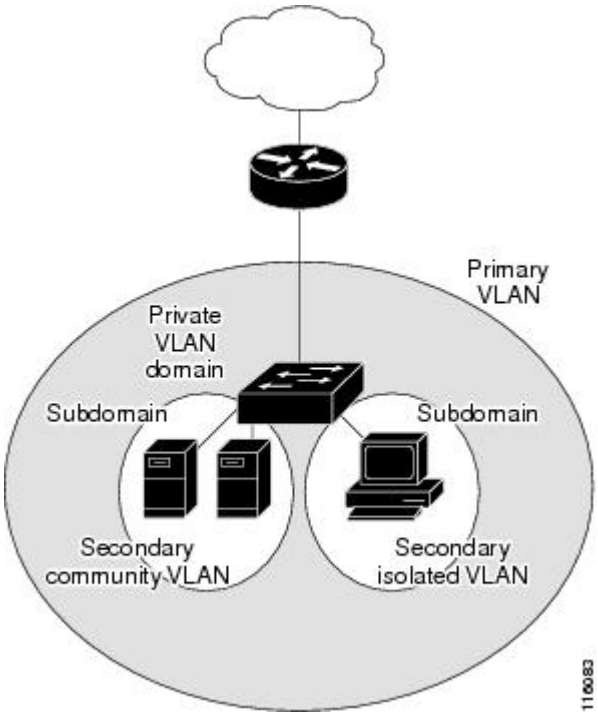
- Information About Private VLANs, page 33
- Guidelines and Limitations for Private VLANs, page 38
- Configuring a Private VLAN, page 38
- Verifying Private VLAN Configuration, page 44

Information About Private VLANs

A private VLAN partitions the Ethernet broadcast domain of a VLAN into subdomains, allowing you to isolate the ports on the switch from each other. A subdomain consists of a primary VLAN and one or more secondary VLANs (see the following figure). All VLANs in a private VLAN domain share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another. The secondary VLANs may either be isolated VLANs or community VLANs. A host on an isolated VLAN can only communicate with the associated

promiscuous port in its primary VLAN. Hosts on community VLANs can communicate among themselves and with their associated promiscuous port but not with ports in other community VLANs.

Figure 3: Private VLAN Domain



Note You must first create the VLAN before you can convert it to a private VLAN, either primary or secondary.

Primary and Secondary VLANs in Private VLANs

A private VLAN domain has only one primary VLAN. Each port in a private VLAN domain is a member of the primary VLAN; the primary VLAN is the entire private VLAN domain.

Secondary VLANs provide isolation between ports within the same private VLAN domain. The following two types are secondary VLANs within a primary VLAN:

- Isolated VLANs—Ports within an isolated VLAN cannot communicate directly with each other at the Layer 2 level.
- Community VLANs—Ports within a community VLAN can communicate with each other but cannot communicate with ports in other community VLANs or in any isolated VLANs at the Layer 2 level.

Private VLAN Ports

The three types of PVLAN ports are as follows:

- Promiscuous port—A promiscuous port belongs to the primary VLAN. The promiscuous port can communicate with all interfaces, including the community and isolated host ports, that belong to those secondary VLANs associated to the promiscuous port and associated with the primary VLAN. You can have several promiscuous ports in a primary VLAN. Each promiscuous port can have several secondary VLANs or no secondary VLANs that are associated to that port. You can associate a secondary VLAN to more than one promiscuous port, as long as the promiscuous port and secondary VLANs are within the same primary VLAN. You may want to do this for load-balancing or redundancy purposes. You can also have secondary VLANs that are not associated to any promiscuous port.

  A promiscuous port can be configured as an access port.

- Isolated port—An isolated port is a host port that belongs to an isolated secondary VLAN. This port has complete isolation from other ports within the same PVLAN domain, except that it can communicate with associated promiscuous ports. PVLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports. You can have more than one isolated port in a specified isolated VLAN. Each port is completely isolated from all other ports in the isolated VLAN.

  An isolated port can be configured an access port.

- Community port—A community port is a host port that belongs to a community secondary VLAN. Community ports communicate with other ports in the same community VLAN and with associated promiscuous ports. These interfaces are isolated from all other interfaces in other communities and from all isolated ports within the PVLAN domain.

  A community port must be configured as an access port.

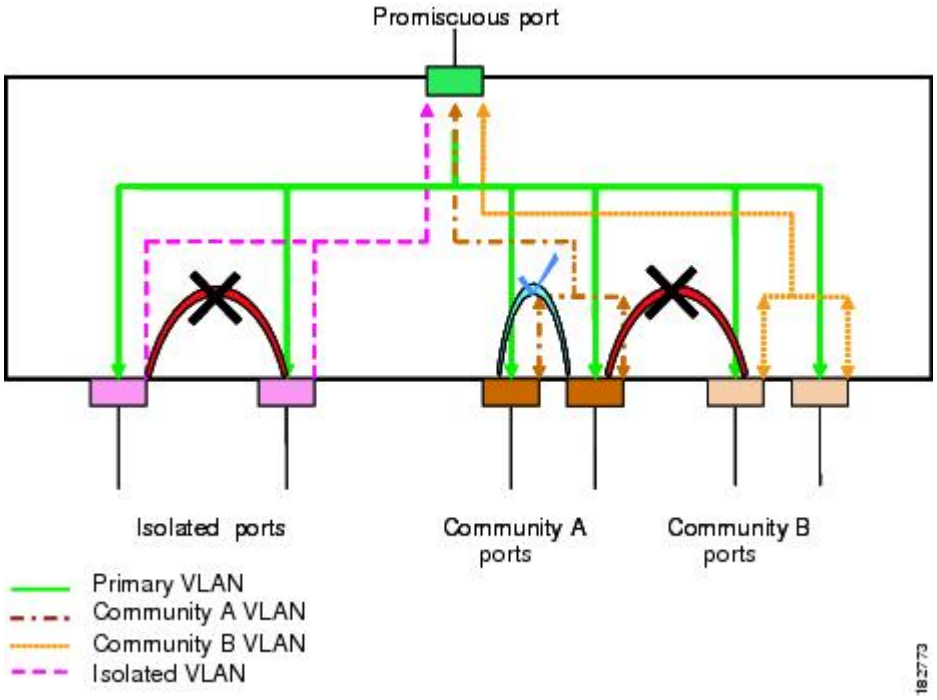
Primary, Isolated, and Community Private VLANs

Primary VLANs and the two types of secondary VLANs (isolated and community) have these characteristics:

- Primary VLAN— The primary VLAN carries traffic from the promiscuous ports to the host ports, both isolated and community, and to other promiscuous ports.
- Isolated VLAN —An isolated VLAN is a secondary VLAN that carries unidirectional traffic upstream from the hosts toward the promiscuous ports. You can configure only one isolated VLAN in a PVLAN domain. An isolated VLAN can have several isolated ports. The traffic from each isolated port also remains completely separate.
- Community VLAN—A community VLAN is a secondary VLAN that carries upstream traffic from the community ports to the promiscuous port and to other host ports in the same community. You can configure multiple community VLANs in a PVLAN domain. The ports within one community can communicate, but these ports cannot communicate with ports in any other community or isolated VLAN in the private VLAN.

The following figure shows the traffic flows within a PVLAN, along with the types of VLANs and types of ports.

Figure 4: Private VLAN Traffic Flows



Note The PVLAN traffic flows are unidirectional from the host ports to the promiscuous ports. Traffic received on primary VLAN enforces no separation and forwarding is done as in a normal VLAN.

A promiscuous access port can serve only one primary VLAN and multiple secondary VLANs (community and isolated VLANs). With a promiscuous port, you can connect a wide range of devices as access points to a PVLAN. For example, you can use a promiscuous port to monitor or back up all the PVLAN servers from an administration workstation.

In a switched environment, you can assign an individual PVLAN and associated IP subnet to each individual or common group of end stations. The end stations need to communicate only with a default gateway to communicate outside the private VLAN.

Associating Primary and Secondary VLANs

To allow host ports in secondary VLANs to communicate outside the private VLAN, you associate secondary VLANs to the primary VLAN. If the association is not operational, the host ports (community and isolated ports) in the secondary VLAN are brought down.

Note You can associate a secondary VLAN with only one primary VLAN.

For an association to be operational, the following conditions must be met:

- The primary VLAN must exist and be configured as a primary VLAN.
- The secondary VLAN must exist and be configured as either an isolated or community VLAN.

Note Use the **show vlan private-vlan** commmand to verify that the association is operational. The switch does not display an error message when the association is nonoperational.

If you delete either the primary or secondary VLAN, the ports that are associated with the VLAN become inactive. Use the **no private-vlan** command to return the VLAN to the normal mode. All primary and secondary associations on that VLAN are suspended, but the interfaces remain in private VLAN mode. When you convert the VLAN back to private VLAN mode, the original associations are reinstated.

If you enter the **no vlan** command for the primary VLAN, all private VLAN associations with that VLAN are deleted. However, if you enter the **no vlan** command for a secondary VLAN, the private VLAN associations with that VLAN are suspended and are restored when you recreate the specified VLAN and configure it as the previous secondary VLAN.

In order to change the association between a secondary and primary VLAN, you must first remove the current association and then add the desired association.

Private VLAN Promiscuous Trunks

The Cisco Nexus 3000 Series device does not support Private VLAN trunk ports.

Private VLAN Isolated Trunks

The Cisco Nexus 3000 Series device does not support Private VLAN trunk ports.

Broadcast Traffic in Private VLANs

Broadcast traffic from ports in a private VLAN flows in the following ways:

- The broadcast traffic flows from a promiscuous port to all ports in the primary VLAN (which includes all the ports in the community and isolated VLANs). This broadcast traffic is distributed to all ports within the primary VLAN, including those ports that are not configured with private VLAN parameters.
- The broadcast traffic from an isolated port is distributed only to those promiscuous ports in the primary VLAN that are associated to that isolated port.
- The broadcast traffic from community ports is distributed to all ports within the port's community and to all promiscuous ports that are associated to the community port. The broadcast packets are not distributed to any other communities within the primary VLAN, or to any isolated ports.

Private VLAN Port Isolation

You can use private VLANs to control access to end stations as follows:

- Configure selected interfaces connected to end stations as isolated ports to prevent any communication. For example, if the end stations are servers, this configuration prevents communication between the servers.
- Configure interfaces connected to default gateways and selected end stations (for example, backup servers) as promiscuous ports to allow all end stations access to a default gateway.

Guidelines and Limitations for Private VLANs

When configuring PVLANs, follow these guidelines:

- You must have already created the VLAN before you can assign the specified VLAN as a private VLAN.
- You must enable PVLANs before the switch can apply the PVLAN functionality.
- You cannot disable PVLANs if the switch has any operational ports in a PVLAN mode.
- Enter the **private-vlan synchronize** command from within the Multiple Spanning Tree (MST) region definition to map the secondary VLANs to the same MST instance as the primary VLAN.

Configuring a Private VLAN

Enabling Private VLANs

You must enable private VLANs on the switch to use the private VLAN functionality.

Note The private VLAN commands do not appear until you enable the private VLAN feature.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature private-vlan**
3. (Optional) switch(config)# **no feature private-vlan**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **feature private-vlan**	Enables the private VLAN feature on the switch.
Step 3	switch(config)# **no feature private-vlan**	(Optional) Disables the private VLAN feature on the switch. **Note** You cannot disable private VLANs if there are operational ports on the switch that are in private VLAN mode.

	Command or Action	Purpose

This example shows how to enable the private VLAN feature on the switch:

```
switch# configure terminal
switch(config)# feature private-vlan
```

Configuring a VLAN as a Private VLAN

To create a private VLAN, you first create a VLAN, and then configure that VLAN to be a private VLAN.

Before You Begin

Ensure that the private VLAN feature is enabled.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vlan** {*vlan-id* | *vlan-range*}
3. switch(config-vlan)# **private-vlan** {**community** | **isolated** | **primary**}
4. (Optional) switch(config-vlan)# **no private-vlan** {**community** | **isolated** | **primary**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vlan** {*vlan-id* | *vlan-range*}	Places you into the VLAN configuration submode.
Step 3	switch(config-vlan)# **private-vlan** {**community** | **isolated** | **primary**}	Configures the VLAN as either a community, isolated, or primary private VLAN. In a private VLAN, you must have one primary VLAN. You can have multiple community and isolated VLANs.
Step 4	switch(config-vlan)# **no private-vlan** {**community** | **isolated** | **primary**}	(Optional) Removes the private VLAN configuration from the specified VLAN(s) and returns it to normal VLAN mode. If you delete either the primary or secondary VLAN, the ports that are associated with the VLAN become inactive.

This example shows how to assign VLAN 5 to a private VLAN as the primary VLAN:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# private-vlan primary
```

This example shows how to assign VLAN 100 to a private VLAN as a community VLAN:

```
switch# configure terminal
switch(config)# vlan 100
switch(config-vlan)# private-vlan community
```

This example shows how to assign VLAN 200 to a private VLAN as an insolated VLAN:

```
switch# configure terminal
switch(config)# vlan 200
switch(config-vlan)# private-vlan isolated
```

Associating Secondary VLANs with a Primary Private VLAN

When you associate secondary VLANs with a primary VLAN, follow these guidelines:

- The *secondary-vlan-list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single secondary VLAN ID or a hyphenated range of secondary VLAN IDs.
- The *secondary-vlan-list* parameter can contain multiple community VLAN IDs and one isolated VLAN ID.
- Enter a *secondary-vlan-list* or use the add keyword with a *secondary-vlan-list* to associate secondary VLANs with a primary VLAN.
- Use the remove keyword with a *secondary-vlan-list* to clear the association between secondary VLANs and a primary VLAN.
- You change the association between a secondary and primary VLAN by removing the existing association and then adding the desired association.

If you delete either the primary or secondary VLAN, the VLAN becomes inactive on the port where the association is configured. When you enter the **no private-vlan** command, the VLAN returns to the normal VLAN mode. All primary and secondary associations on that VLAN are suspended, but the interfaces remain in private VLAN mode. If you again convert the specified VLAN to private VLAN mode, the original associations are reinstated.

If you enter the **no vlan** command for the primary VLAN, all private VLAN associations with that VLAN are lost. However, if you enter the **no vlan** command for a secondary VLAN, the private VLAN associations with that VLAN are suspended and are reinstated when you recreate the specified VLAN and configure it as the previous secondary VLAN.

Before You Begin

Ensure that the private VLAN feature is enabled.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vlan** *primary-vlan-id*
3. switch(config-vlan)# **private-vlan association** {[**add**] *secondary-vlan-list* | **remove** *secondary-vlan-list*}
4. (Optional) switch(config-vlan)# **no private-vlan association**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vlan** *primary-vlan-id*	Enters the number of the primary VLAN that you are working in for the private VLAN configuration.
Step 3	switch(config-vlan)# **private-vlan association** {[**add**] *secondary-vlan-list* | **remove** *secondary-vlan-list*}	Associates the secondary VLANs with the primary VLAN.
Step 4	switch(config-vlan)# **no private-vlan association**	(Optional) Removes all associations from the primary VLAN and returns it to normal VLAN mode.

This example shows how to associate community VLANs 100 through 110 and isolated VLAN 200 with primary VLAN 5:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# private-vlan association 100-110, 200
```

Configuring an Interface as a Private VLAN Host Port

In private VLANs, host ports are part of the secondary VLANs, which are either community VLANs or isolated VLANs. Configuring a private VLAN host port involves two steps. First, you define the port as a private VLAN host port and then you configure a host association between the primary and secondary VLANs.

Note We recommend that you enable BPDU Guard on all interfaces configured as a host ports.

Before You Begin

Ensure that the private VLAN feature is enabled.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type* [*chassis*/]*slot*/*port*
3. switch(config-if)# **switchport mode private-vlan host**
4. switch(config-if)# **switchport private-vlan host-association** {*primary-vlan-id*} {*secondary-vlan-id*}
5. (Optional) switch(config-if)# **no switchport private-vlan host-association**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type* [*chassis*/]*slot*/*port*	Selects the port to configure as a private VLAN host port. This port can be on a Fabric Extender (identified by the chassis option).
Step 3	switch(config-if)# **switchport mode private-vlan host**	Configures the port as a host port for a private VLAN.
Step 4	switch(config-if)# **switchport private-vlan host-association** {*primary-vlan-id*} {*secondary-vlan-id*}	Associates the port with the primary and secondary VLANs of a private VLAN. The secondary VLAN can be either an isolated or community VLAN.
Step 5	switch(config-if)# **no switchport private-vlan host-association**	(Optional) Removes the private VLAN association from the port.

This example shows how to configure Ethernet port 1/12 as a host port for a private VLAN and associate it to primary VLAN 5 and secondary VLAN 101:

```
switch# configure terminal
switch(config)# interface ethernet 1/12
switch(config-if)# switchport mode private-vlan host
switch(config-if)# switchport private-vlan host-association 5 101
```

Configuring an Interface as a Private VLAN Promiscuous Port

In a private VLAN domain, promiscuous ports are part of the primary VLAN. Configuring a promiscuous port involves two steps. First, you define the port as a promiscuous port and then you configure the mapping between a secondary VLAN and the primary VLAN.

Before You Begin

Ensure that the private VLAN feature is enabled.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **switchport mode private-vlan promiscuous**
4. switch(config-if)# **switchport private-vlan mapping** {*primary-vlan-id*} {*secondary-vlan-list* | **add** *secondary-vlan-list* | **remove** *secondary-vlan-list*}
5. (Optional) switch(config-if)# **no switchport private-vlan mapping**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Selects the port to configure as a private VLAN promiscuous port. A physical interface is required. This port cannot be on a Fabric Extender.
Step 3	switch(config-if)# **switchport mode private-vlan promiscuous**	Configures the port as a promiscuous port for a private VLAN. You can only enable a physical Ethernet port as the promiscuous port.
Step 4	switch(config-if)# **switchport private-vlan mapping** {*primary-vlan-id*} {*secondary-vlan-list* | **add** *secondary-vlan-list* | **remove** *secondary-vlan-list*}	Configures the port as a promiscuous port and associates the specified port with a primary VLAN and a selected list of secondary VLANs. The secondary VLAN can be either an isolated or community VLAN.
Step 5	switch(config-if)# **no switchport private-vlan mapping**	(Optional) Clears the mapping from the private VLAN.

This example shows how to configure Ethernet interface 1/4 as a promiscuous port associated with primary VLAN 5 and secondary isolated VLAN 200:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode private-vlan promiscuous
switch(config-if)# switchport private-vlan mapping 5 200
```

Configuring a Promiscuous Trunk Port

You cannot perform this task because the Cisco Nexus 3000 Series device does not support Private VLAN trunk ports.

Configuring an Isolated Trunk Port

You cannot perform this task because the Cisco Nexus 3000 Series device does not support Private VLAN trunk ports.

Configuring the Allowed VLANs for PVLAN Trunking Ports

You cannot perform this task because the Cisco Nexus 3000 Series device does not support Private VLAN trunk ports.

Configuring Native 802.1Q VLANs on Private VLANs

You cannot perform this task because the Cisco Nexus 3000 Series device does not support Private VLAN trunk ports.

Verifying Private VLAN Configuration

To display private VLAN configuration information, use the following commands:

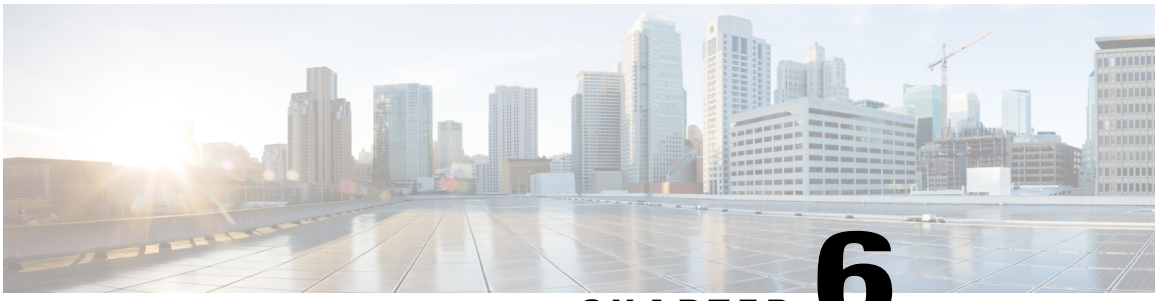
Command	Purpose
switch# **show feature**	Displays the features enabled on the switch.
switch# **show interface switchport**	Displays information on all interfaces configured as switch ports.
switch# **show vlan private-vlan** [**type**]	Displays the status of the private VLAN.

The following example shows how to display the private VLAN configuration:

```
switch# show vlan private-vlan
Primary  Secondary  Type             Ports
-------  ---------  ---------------  ------------------------------------------
5        100        community
5        101        community        Eth1/12, Eth100/1/1
5        102        community
5        110        community
5        200        isolated         Eth1/2
switch# show vlan private-vlan type
Vlan Type
---- -----------------
5    primary
100  community
101  community
102  community
110  community
200  isolated
```

The following example shows how to display enabled features (some of the output has been removed for brevity):

```
switch# show feature
Feature Name          Instance  State
--------------------  --------  --------
fcsp                  1         enabled
...
interface-vlan        1         enabled
private-vlan          1         enabled
udld                  1         disabled
...
```



CHAPTER 6

Configuring Access and Trunk Interfaces

This chapter contains the following sections:

- Information About Access and Trunk Interfaces, page 45
- Configuring Access and Trunk Interfaces, page 49
- Verifying Interface Configuration, page 54

Information About Access and Trunk Interfaces

Understanding Access and Trunk Interfaces

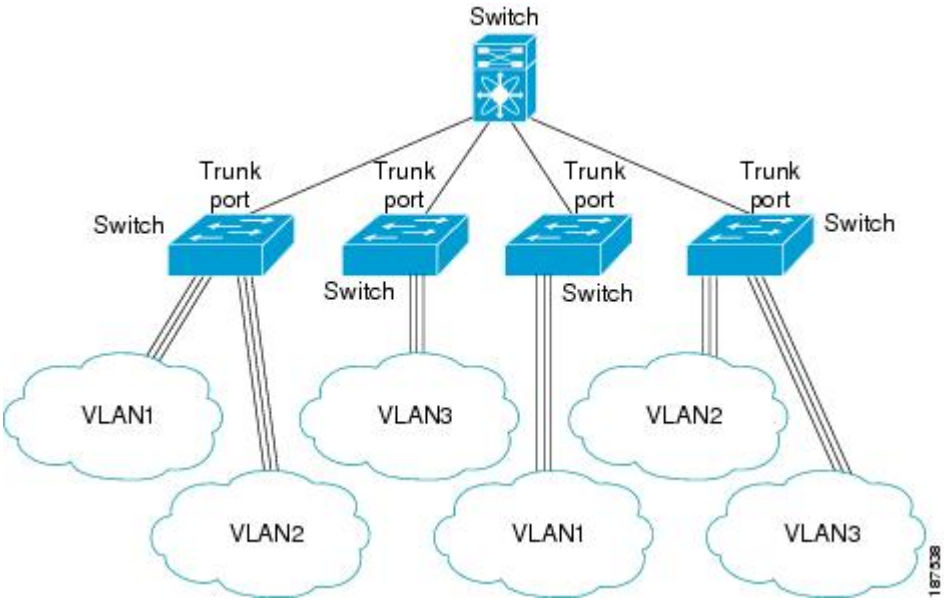
Ethernet interfaces can be configured either as access ports or a trunk ports, as follows:

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

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

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

Figure 5: Devices in a Trunking Environment



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

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

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

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

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

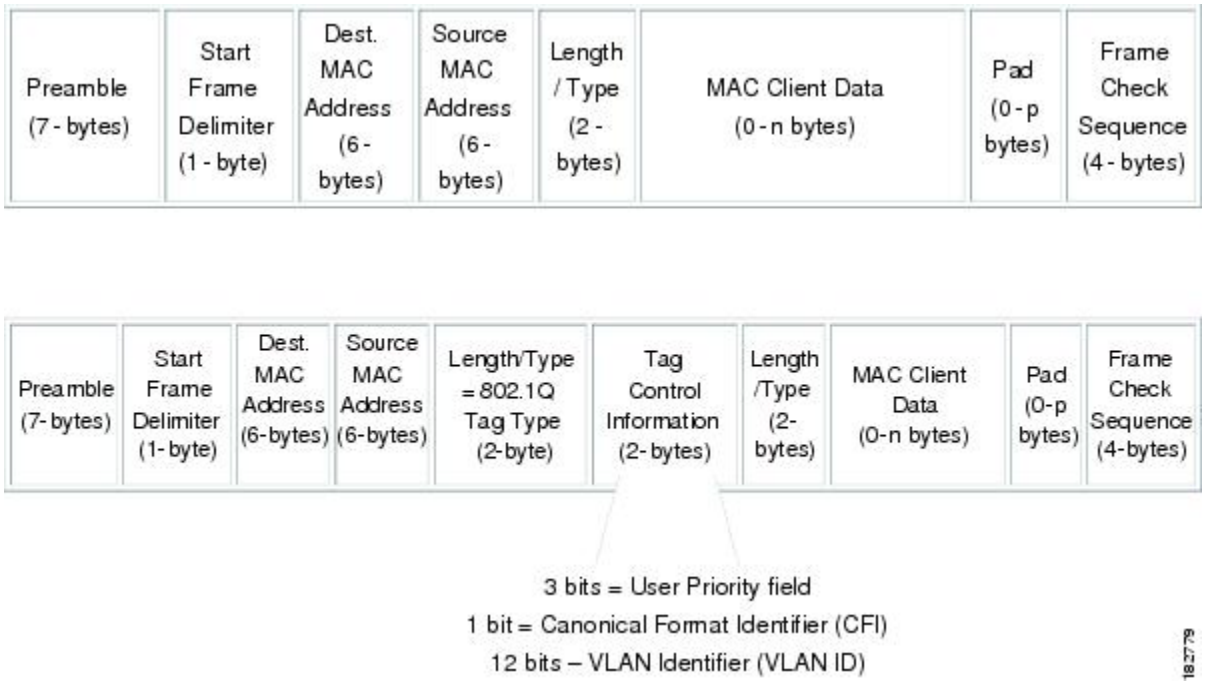
Understanding IEEE 802.1Q Encapsulation

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

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

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

Figure 6: Header without and with 802.1Q Tag Included



Understanding Access VLANs

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

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

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

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

Understanding the Native VLAN ID for Trunk Ports

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

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

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

Understanding Allowed VLANs

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

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

Understanding Native 802.1Q VLANs

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

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

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

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

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

This feature is supported on all the directly connected Ethernet and EtherChannel interfaces of the Cisco Nexus 3000 Series switch.

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

Configuring Access and Trunk Interfaces

Configuring a LAN Interface as an Ethernet Access Port

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

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}
3. switch(config-if)# **switchport mode** {**access** | **trunk**}
4. switch(config-if)# **switchport access vlan** *vlan-id*

DETAILED STEPS

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

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

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

Configuring Access Host Ports

By using switchport host, you can make an access port a spanning-tree edge port, and enable bpdu filtering and bpdu guard at the same time.

Before You Begin

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **switchport host**

DETAILED STEPS

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

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

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

Configuring Trunk Ports

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

Note Cisco NX-OS supports only 802.1Q encapsulation.

To configure a trunk port, perform this task:

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {*type slot*/*port* | **port-channel** *number*}
3. switch(config-if)# **switchport mode** {**access** | **trunk**}

DETAILED STEPS

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

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

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

Configuring the Native VLAN for 802.1Q Trunking Ports

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {*type slot*/*port* | **port-channel** *number*}
3. switch(config-if)# **switchport trunk native vlan** *vlan-id*

DETAILED STEPS

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

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

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

Configuring the Allowed VLANs for Trunking Ports

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

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {*type slot*/*port* | **port-channel** *number*}
3. switch(config-if)# **switchport trunk allowed vlan** {*vlan-list* **all** | **none** [**add** |except | none | **remove** {*vlan-list*}]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** {*type slot*/*port* | **port-channel** *number*}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport trunk allowed vlan** {*vlan-list* **all** | **none** [**add** |except | none | **remove** {*vlan-list*}]}	Sets allowed VLANs for the trunk interface. The default is to allow all VLANs on the trunk interface: 1 to 3967 and 4048 to 4094. VLANs 3968 to 4047 are the default VLANs reserved for internal use by default; this group of VLANs is configurable. By default, all VLANs are allowed on all trunk interfaces.

	Command or Action	Purpose
		Note You cannot add internally allocated VLANs as allowed VLANs on trunk ports. The system returns a message if you attempt to list an internally allocated VLAN as an allowed VLAN.

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

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

Configuring Native 802.1Q VLANs

Typically, you configure 802.1Q trunks with a native VLAN ID, which strips tagging from all packets on that VLAN. This configuration allows all untagged traffic and control traffic to transit the Cisco Nexus 3000 Series switch. Packets that enter the switch with 802.1Q tags that match the native VLAN ID value are similarly stripped of tagging.

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

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

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vlan dot1q tag native**
3. (Optional) switch(config)# **no vlan dot1q tag native**
4. (Optional) switch# **show vlan dot1q tag native**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vlan dot1q tag native**	Enables dot1q (IEEE 802.1Q) tagging for all native VLANs on all trunked ports on the Cisco Nexus 3000 Series switch. By default, this feature is disabled.

	Command or Action	Purpose
Step 3	switch(config)# **no vlan dot1q tag native**	(Optional) Disables dot1q (IEEE 802.1Q) tagging for all native VLANs on all trunked ports on the switch.
Step 4	switch# **show vlan dot1q tag native**	(Optional) Displays the status of tagging on the native VLANs.

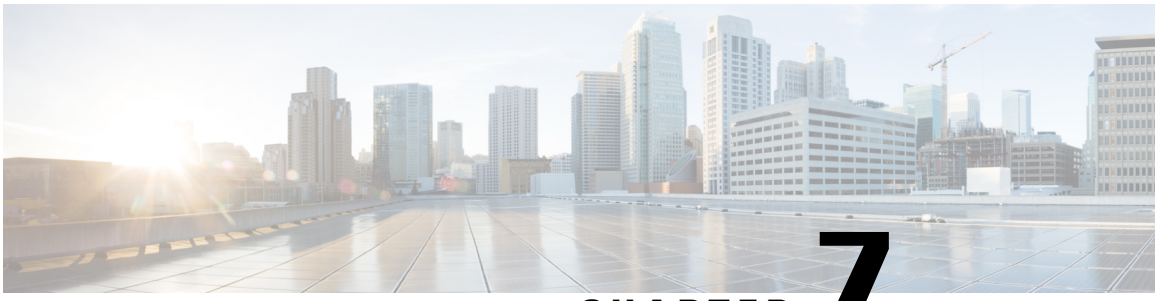
The following example shows how to enable 802.1Q tagging on the switch:

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

Verifying Interface Configuration

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

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



CHAPTER 7

Configuring Port Channels

This chapter contains the following sections:

- Information About Port Channels, page 55
- Configuring Port Channels, page 62
- Verifying Port Channel Configuration, page 70
- Verifying the Load-Balancing Outgoing Port ID , page 71

Information About Port Channels

A port channel bundles up to 16 individual interfaces into a group to provide increased bandwidth and redundancy. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational as long as at least one physical interface within the port channel is operational.

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

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

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

Related Topics

LACP Overview, on page 59

Understanding Port Channels

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

You can collect up to 16 ports into a static port channel or you can enable the Link Aggregation Control Protocol (LACP). Configuring port channels with LACP requires slightly different steps than configuring static port channels.

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

A port channel bundles individual links into a channel group to create a single logical link that provides the aggregate bandwidth of up to 16 physical links. If a member port within a port channel fails, traffic previously carried over the failed link switches to the remaining member ports within the port channel.

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

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

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

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

Compatibility Requirements

When you add an interface to a port channel group, Cisco NX-OS checks certain interface attributes to ensure that the interface is compatible with the channel group. Cisco NX-OS also checks a number of operational attributes for an interface before allowing that interface to participate in the port-channel aggregation.

The compatibility check includes the following operational attributes:

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

- Untagged CoS

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

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

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

- Bandwidth
- MAC address
- Spanning Tree Protocol

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

- Description
- CDP
- LACP port priority
- Debounce

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

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

- SNMP traps

Load Balancing Using Port Channels

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

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

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

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

- Destination MAC address
- Source MAC address
- Source and destination MAC address
- Destination IP address
- Source IP address
- Source and destination IP address
- Destination TCP/UDP port number
- Source TCP/UDP port number
- Source and destination TCP/UDP port number

Table 5: Port channel Load-Balancing Criteria

Configuration	Layer 2 Criteria	Layer 3 Criteria	Layer 4 Criteria
Destination MAC	Destination MAC	Destination MAC	Destination MAC
Source MAC	Source MAC	Source MAC	Source MAC
Source and destination MAC	Source and destination MAC	Source and destination MAC	Source and destination MAC

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

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

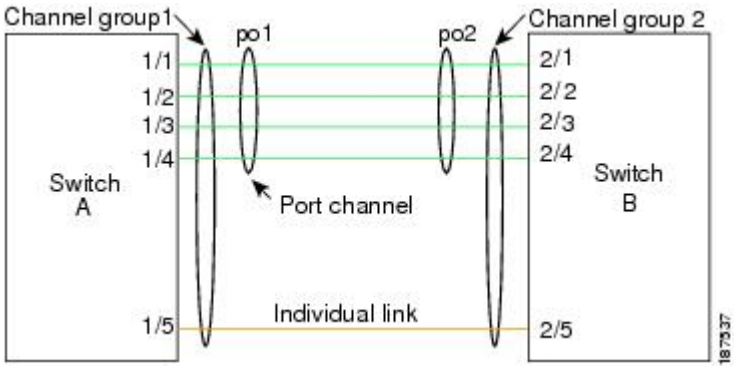
Understanding LACP

LACP Overview

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

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

Figure 7: Individual Links Combined into a Port channel



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

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

You cannot disable LACP while any LACP configurations are present.

LACP ID Parameters

LACP uses the following parameters:

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

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

- LACP port priority—Each port configured to use LACP has an LACP port priority. You can accept the default value of 32768 for the LACP port priority, or you can configure a value between 1 and 65535. LACP uses the port priority with the port number to form the port identifier. LACP uses the port priority to decide which ports should be put in standby mode when there is a limitation that prevents all compatible ports from aggregating and which ports should be put into active mode. A higher port priority value means a lower priority for LACP. You can configure the port priority so that specified ports have a lower priority for LACP and are most likely to be chosen as active links, rather than hot-standby links.
- LACP administrative key—LACP automatically configures an administrative key value equal to the channel-group number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port's ability to aggregate with other ports is determined by these factors:

- Port physical characteristics, such as the data rate, the duplex capability, and the point-to-point or shared medium state
- Configuration restrictions that you establish

Channel Modes

Individual interfaces in port channels are configured with channel modes. When you run static port channels, with no protocol, the channel mode is always set to on. After you enable LACP globally on the device, you enable LACP for each channel by setting the channel mode for each interface to active or passive. You can configure either channel mode for individual links in the LACP channel group.

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

The following table describes the channel modes.

Table 6: Channel Modes for Individual Links in a Port channel

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

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

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

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

LACP Marker Responders

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

LACP-Enabled and Static Port Channel Differences

The following table provides a brief summary of major differences between port channels with LACP enabled and static port channels.

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

Configurations	EtherChannels with LACP Enabled	Static EtherChannels
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	16

Configuring Port Channels

Creating a Port Channel

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

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface port-channel** *channel-number*
3. switch(config)# **no interface port-channel** *channel-number*

DETAILED STEPS

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

This example shows how to create an port channel:

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

Adding a Port to a Port Channel

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

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. (Optional) switch(config-if)# **switchport mode trunk**
4. (Optional) switch(config-if)# **switchport trunk** {**allowed vlan** *vlan-id* | **native vlan** *vlan-id*}
5. switch(config-if)# **channel-group** *channel-number*
6. (Optional) switch(config-if)# **no channel-group**

DETAILED STEPS

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

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

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

Configuring Load Balancing Using Port Channels

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

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-channel load-balance ethernet** {[**destination-ip** | **destination-mac** | **destination-port** | **source-dest-ip** | **source-dest-mac** | **source-dest-port** | **source-ip** | **source-mac** | **source-port**] crc-poly}
3. (Optional) switch(config)# **no port-channel load-balance ethernet**
4. (Optional) switch# **show port-channel load-balance**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **port-channel load-balance ethernet** {[**destination-ip** | **destination-mac** | **destination-port** | **source-dest-ip** | **source-dest-mac** | **source-dest-port** | **source-ip** | **source-mac** | **source-port**] crc-poly}	Specifies the load-balancing algorithm for the device. The range depends on the device. The default is source-dest-mac.
Step 3	switch(config)# **no port-channel load-balance ethernet**	(Optional) Restores the default load-balancing algorithm of source-dest-mac.
Step 4	switch# **show port-channel load-balance**	(Optional) Displays the port-channel load-balancing algorithm.

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

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

Configuring Hardware Hashing for Multicast Traffic

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface port-channel** *channel-number*
3. switch(config-if)# **hardware multicast hw-hash**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	switch(config-if)# **hardware multicast hw-hash**	Configures hardware hashing for the specified port channel.

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

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

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

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

Enabling LACP

LACP is disabled by default; you must enable LACP before you begin LACP configuration. You cannot disable LACP while any LACP configuration is present.

LACP learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once LACP identifies correctly matched Ethernet links, it facilitates grouping the links into an port channel. The port channel is then added to the spanning tree as a single bridge port.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature lacp**
3. (Optional) switch(config)# **show feature**

DETAILED STEPS

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

This example shows how to enable LACP:

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

Configuring the Channel Mode for a Port

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

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

Before You Begin

Ensure that you have enabled the LACP feature.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]
4. switch(config-if)# **no channel-group** *number* **mode**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]	Specifies the port mode for the link in a port channel. After LACP is enabled, you configure each link or the entire channel as active or passive. force—Specifies that the LAN port be forcefully added to the channel group. This option is available in Cisco NX-OS Release 5.0(2)N2(1). mode—Specifies the port channel mode of the interface. active—Specifies that when you enable LACP, this command enables LACP on the specified interface. The interface is in an active negotiating state in which the port initiates negotiations with other ports by sending LACP packets. on—(Default mode) Specifies that all port channels that are not running LACP remain in this mode. passive—Enables LACP only if an LACP device is detected. The interface is in a passive negotiation state in which the port responds to LACP packets that it receives but does not initiate LACP negotiation. When you run port channels with no associated protocol, the channel mode is always on.
Step 4	switch(config-if)# **no channel-group** *number* **mode**	Returns the port mode to on for the specified interface.

This example shows how to set the LACP-enabled interface to active port-channel mode for Ethernet interface 1/4 in channel group 5:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# channel-group 5 mode active
```

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

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

Configuring the LACP Fast Timer Rate

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

Before You Begin

Ensure that you have enabled the LACP feature.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **lacp rate fast**

DETAILED STEPS

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

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

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

switch(config-if)# lacp rate fast
```

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

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

Configuring the LACP System Priority and System ID

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

Before You Begin

Ensure that you have enabled the LACP feature.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **lacp system-priority** *priority*
3. (Optional) switch# **show lacp system-identifier**

DETAILED STEPS

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

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

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

Configuring the LACP Port Priority

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

Before You Begin

Ensure that you have enabled the LACP feature.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **lacp port-priority** *priority*

DETAILED STEPS

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

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

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

Verifying Port Channel Configuration

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

Command	Purpose
switch# **show interface port-channel** *channel-number*	Displays the status of a port channel interface.
switch# **show feature**	Displays enabled features.
switch# **show resource**	Displays the number of resources currently available in the system.
switch# **show lacp** {**counters** | **interface** *type slot*/*port* | **neighbor** | **port-channel** | **system-identifier**}	Displays LACP information.
switch# **show port-channel compatibility-parameters**	Displays the parameters that must be the same among the member ports in order to join a port channel.
switch# **show port-channel database** [**interface port-channel** *channel-number*]	Displays the aggregation state for one or more port-channel interfaces.

Command	Purpose
switch# **show port-channel summary**	Displays a summary for the port channel interfaces.
switch# **show port-channel traffic**	Displays the traffic statistics for port channels.
switch# **show port-channel usage**	Displays the range of used and unused channel numbers.
switch# **show port-channel database**	Displays information on current running of the port channel feature.
switch# **show port-channel load-balance**	Displays information about load-balancing using port channels.

Verifying the Load-Balancing Outgoing Port ID

Command Guidelines

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

Note Certain traffic flows are not subject to hashing, for example when there is a single port in a port-channel.

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

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

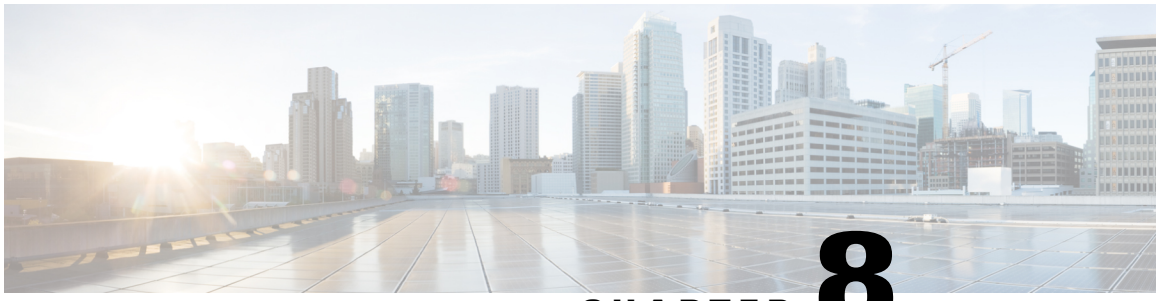
Example

The following example shows the output of the short **port-channel load-balance** command.

switch#**show port-channel load-balance forwarding-path interface port-channel** *10* **vlan** *1* **dst-ip** *1.225.225.225* **src-ip** *1.1.10.10* **src-mac** *aa:bb:cc:dd:ee:ff* **l4-src-port** *0* **l4-dst-port** *1*

```
Missing params will be substituted by 0's.Load-balance Algorithm on switch:
source-dest-portcrc8_hash: 204 Outgoing port id: Ehernet1/1 Param(s) used
to calculate load-balance:
dst-port: 1
src-port: 0
dst-ip: 1.225.225.225
```

```
src-ip: 1.1.10.10
dst-mac: 0000.0000.0000
src-mac: aabb.ccdd.eeff
```



CHAPTER 8

Configuring Virtual Port Channels

This chapter contains the following sections:

- Information About vPCs, page 73
- Guidelines and Limitations for vPCs, page 82
- Configuring vPCs, page 83
- Verifying the vPC Configuration, page 97
- vPC Default Settings, page 102

Information About vPCs

vPC Overview

A virtual port channel (vPC) allows links that are physically connected to two different Cisco Nexus 3000 Series switches to appear as a single port channel by a third device (see the following figure). The third device can be a switch, server, or any other networking device. A vPC can provide multipathing, which allows you to create redundancy by enabling multiple parallel paths between nodes and load balancing traffic where alternative paths exist.

You configure the EtherChannels by using one of the following:

- No protocol
- Link Aggregation Control Protocol (LACP)

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

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

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

To create a vPC peer link you configure an EtherChannel on one Cisco Nexus 3000 Series switch by using two or more Ethernet ports. On the other switch, you configure another EtherChannel again using two or more Ethernet ports. Connecting these two EtherChannels together creates a vPC peer link.

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

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

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

A vPC provides the following benefits:

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

Terminology

vPC Terminology

The terminology used in vPCs is as follows:

- vPC—The combined EtherChannel between the vPC peer devices and the downstream device.
- vPC peer device—One of a pair of devices that are connected with the special EtherChannel known as the vPC peer link.
- vPC peer link—The link used to synchronize states between the vPC peer devices.
- vPC member port—Interfaces that belong to the vPCs.
- vPC domain—This domain includes both vPC peer devices, the vPC peer-keepalive link, and all of the port channels in the vPC connected to the downstream devices. It is also associated to the configuration mode that you must use to assign vPC global parameters. The vPC domain ID must be the same on both switches.
- vPC peer-keepalive link—The peer-keepalive link monitors the vitality of a vPC peer Cisco Nexus 3000 Series device. The peer-keepalive link sends configurable, periodic keepalive messages between vPC peer devices.

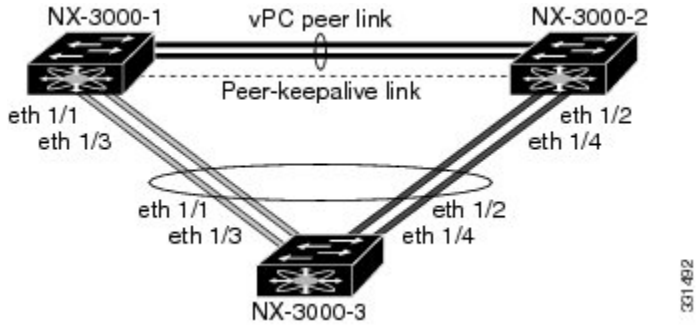
No data or synchronization traffic moves over the vPC peer-keepalive link; the only traffic on this link is a message that indicates that the originating switch is operating and running vPCs.

Supported vPC Topologies

Cisco Nexus 3000 Series Switch vPC Topology

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

Figure 8: Switch-to-Switch vPC Topology



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

vPC Domain

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

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

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

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

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

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

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

Peer-Keepalive Link and Messages

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

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

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

Note We recommend that you configure the vPC peer-keepalive link on the Cisco Nexus 3000 Series switch to run in the management VRF using the mgmt 0 interfaces. If you configure the default VRF, ensure that the vPC peer link is not used to carry the vPC peer-keepalive messages.

Compatibility Parameters for vPC Peer Links

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

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

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

Configuration Parameters That Must Be Identical

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

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

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

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

- Port-channel mode: on, off, or active
- Link speed per channel
- Duplex mode per channel
- Trunk mode per channel:
  - Native VLAN
  - VLANs allowed on trunk
  - Tagging of native VLAN traffic
- Spanning Tree Protocol (STP) mode
- STP region configuration for Multiple Spanning Tree (MST)
- Enable or disable state per VLAN
- STP global settings:
  - Bridge Assurance setting
  - Port type setting—We recommend that you set all vPC interfaces as normal ports
  - Loop Guard settings
- STP interface settings:
  - Port type setting
  - Loop Guard
  - Root Guard

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

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

Configuration Parameters That Should Be Identical

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

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

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

Graceful Type-1 Check

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

Per-VLAN Consistency Check

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

vPC Auto-Recovery

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

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

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

The vPC auto-recovery feature is disabled by default.

vPC Peer Links

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

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

vPC Peer Link Overview

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

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

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

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

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

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

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

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

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

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

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

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

vPC Number

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

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

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

vPC Interactions with Other Features

vPC and LACP

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

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

The vPC peer link supports 16 EtherChannel interfaces.

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

vPC Peer Links and STP

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

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

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

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

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

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

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

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

CFSoE

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

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

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

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

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

Guidelines and Limitations for vPCs

vPC has the following configuration guidelines and limitations:

- You must enable the vPC feature before you can configure vPC peer-link and vPC interfaces.
- You must configure the peer-keepalive link before the system can form the vPC peer link.
- The vPC peer-link needs to be formed using a minimum of two 10-Gigabit Ethernet interfaces.
- Cisco recommends that you configure the same VPC domain ID on both peers and, the domain ID should be unique in the network. For example, if there are two different VPCs (one in access and one in aggregation) then each vPC should have a unique domain ID.
- Only port channels can be in vPCs. A vPC can be configured on a normal port channel (switch-to-switch vPC topology) and on a port channel host interface (host interface vPC topology).
- You must configure both vPC peer switches; the configuration is not automatically synchronized between the vPC peer devices.
- Check that the necessary configuration parameters are compatible on both sides of the vPC peer link.
- You may experience minimal traffic disruption while configuring vPCs.
- You should configure all the port channels in the vPC using LACP with the interfaces in active mode.
- When the **peer-switch** command is configured and vPC keepalive messages exchanged through an SVI instead of a management interface, additional Spanning Tree Protocol (STP) configuration is required. STP needs to be disabled on the dedicated link that carries the keepalive traffic between the vPC peers. You can disable STP on the dedicated link by configuring STP BPDUfilter on the both ends of the dedicated link. We recommend that the VLAN of the vPC keepalive SVI be allowed on only the interconnecting dedicated link and disallowed on all other links, including the peer link.

- When the MAC address table exceeds its limit, MAC addresses learned on the peer switch do not age out. However, MAC addresses learned on the local switch continue to age out as expected.

Configuring vPCs

Enabling vPCs

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature vpc**
3. (Optional) switch# **show feature**
4. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

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

This example shows how to enable the vPC feature:

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

Disabling vPCs

You can disable the vPC feature.

Note When you disable the vPC feature, the Cisco Nexus 3000 Series switch clears all the vPC configurations.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no feature vpc**
3. (Optional) switch# **show feature**
4. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

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

This example shows how to disable the vPC feature:

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

Creating a vPC Domain

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

Before You Begin

Ensure that you have enabled the vPC feature.

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. (Optional) switch# **show vpc brief**
4. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

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

This example shows how to create a vPC domain:

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

Configuring a vPC Keepalive Link and Messages

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

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

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

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

Before You Begin

Ensure that you have enabled the vPC feature.

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

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **peer-keepalive destination** *ipaddress* [**hold-timeout** *secs* | **interval** *msecs* {**timeout** *secs*} | **precedence** {*prec-value* | **network** | **internet** | **critical** | **flash-override** | **flash** | **immediate priority** | **routine**} | **tos** {*tos-value* | **max-reliability** | **max-throughput** | **min-delay** | **min-monetary-cost** | **normal**} | **tos-byte** *tos-byte-value*} | **source** *ipaddress* | **vrf** {*name* | **management vpc-keepalive**}]
4. (Optional) switch(config-vpc-domain)# **vpc peer-keepalive destination** *ipaddress* **source** *ipaddress*
5. (Optional) switch# **show vpc peer-keepalive**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

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

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

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

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

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

Configuring a Keepalive Link When Using a Front-Panel 10-Gigabit Ethernet Port

When you use a front-panel 10-Gigabit Ethernet port as vPC keepalive link, we recommend that you create a separate VRF for vPC keepalive messages. A separate VRF eliminates the possibility of disrupted vPC keepalive links that are caused by learning the wrong routes from a dynamic routing protocol. In the following configuration , a new VRF named vpc_keepalive is created for vPC keepalive link

The following example shows how to create a separate VRF named vpc_keepalive for the vPC keepalive link and how to verify the new VRF.

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

L3-NEXUS-2# sh vpc peer-keepalive

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

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

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

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

Creating a vPC Peer Link

You can create a vPC peer link by designating the EtherChannel that you want on each switch as the peer link for the specified vPC domain. We recommend that you configure the EtherChannels that you are designating as the vPC peer link in trunk mode and that you use two ports on separate modules on each vPC peer switch for redundancy.

Before You Begin

Ensure that you have enabled the vPC feature.

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface port-channel** *channel-number*
3. switch(config-if)# **vpc peer-link**
4. (Optional) switch# **show vpc brief**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

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

This example shows how to configure a vPC peer link:

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

Checking the Configuration Compatibility

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

Note The following QoS parameters support Type 2 consistency checks:

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

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

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

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

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

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

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

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

Enabling vPC Auto-Recovery

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **auto-recovery reload-delay** *delay*

DETAILED STEPS

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

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

```
switch(config)# vpc domain 10
switch(config-vpc-domain)# auto-recovery reload-delay 240
Warning:
 Enables restoring of vPCs in a peer-detached state after reload, will wait for 240 seconds
  (by default) to determine if peer is un-reachable
```

This examples shows how to view the status of the auto-recovery feature in vPC domain 10.

```
switch(config-vpc-domain)# show running-config vpc
!Command: show running-config vpc
!Time: Tue Dec  7 02:38:44 2010


version 5.0(3)U2(1)
feature vpc
vpc domain 10
  peer-keepalive destination 10.193.51.170
  auto-recovery
```

Configuring the Restore Time Delay

You can configure a restore timer that delays the vPC from coming back up until after the peer adjacency forms and the VLAN interfaces are back up. This feature avoids packet drops when the routing tables may not be converged before the vPC is once again passing traffic.

Before You Begin

Ensure that you have enabled the vPC feature.

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **delay restore** *time*
4. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Creates a vPC domain on the switch if it does not already exist, and enters the vpc-domain configuration mode.
Step 3	switch(config-vpc-domain)# **delay restore** *time*	Configure the time delay before the vPC is restored. The restore time is the number of seconds to delay bringing up the restored vPC peer device. The range is from 1 to 3600. The default is 30 seconds.
Step 4	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure the delay reload time for a vPC link:

```
switch(config)# vpc domain 1
switch(config-vpc-domain)# delay restore 10
switch(config-vpc-domain)#
```

Excluding VLAN Interfaces From Shutdown When vPC Peer Link Fails

When a vPC peer-link is lost, the vPC secondary switch suspends its vPC member ports and its SVI interfaces. All Layer 3 forwarding is disabled for all VLANs on the vPC secondary switch. You can exclude specific SVI interfaces so that they are not suspended.

Before You Begin

Ensure that the VLAN interfaces have been configured.

-

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain))# **dual-active exclude interface-vlan** *range*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Creates a vPC domain on the switch if it does not already exist, and enters the vpc-domain configuration mode.
Step 3	switch(config-vpc-domain))# **dual-active exclude interface-vlan** *range*	Specifies the VLAN interfaces that should remain up when a vPC peer-link is lost. range—Range of VLAN interfaces that you want to exclude from shutting down. The range is from 1 to 4094.

This example shows how to keep the interfaces on VLAN 10 up on the vPC peer switch if a peer link fails:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# dual-active exclude interface-vlan 10
switch(config-vpc-domain)#
```

Configuring the VRF Name

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

You can specify the VRF name.

SUMMARY STEPS

1. switch# **ping** *ipaddress* **vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **ping** *ipaddress* **vrf** *vrf-name*	Specifies the virtual routing and forwarding (VRF) to use. The VRF name is case sensitive and can be a maximum of 32 characters..

This example shows how to specifiy the VRF named vpc_keepalive.

```
switch# ping 123.1.1.1 vrf vpc_keepalive
PING 123.1.1.1 (123.1.1.1): 56 data bytes
64 bytes from 123.1.1.1: icmp_seq=0 ttl=254 time=3.234 ms
64 bytes from 123.1.1.1: icmp_seq=1 ttl=254 time=4.931 ms
64 bytes from 123.1.1.1: icmp_seq=2 ttl=254 time=4.965 ms
64 bytes from 123.1.1.1: icmp_seq=3 ttl=254 time=4.971 ms
64 bytes from 123.1.1.1: icmp_seq=4 ttl=254 time=4.915 ms
```

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

Binding a VRF Instance to a vPC

You can bind a VRF instance to a vPC. One reserved VLAN is required for each VRF. Without this command, the receivers in a non-VPC VLAN and the receivers connected to a Layer 3 interface may not receive multicast traffic. The non-vPC VLANs are the VLANs that are not trunked over a peer-link.

Before You Begin

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc bind-vrf** *vrf-name* **vlan** *vlan-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vpc bind-vrf** *vrf-name* **vlan** *vlan-id*	Binds a VRF instance to a vPC and specifies the VLAN to bind to the vPC. The VLAN ID range is from 1 to 3967, and 4049 to 4093.

This example shows how to bind a vPC to the default VRF using VLAN 2:

```
switch(config)# vpc bind-vrf default vlan vlan2
```

Moving Other EtherChannels into a vPC

Before You Begin

Ensure that you have enabled the vPC feature.

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface port-channel** *channel-number*
3. switch(config-if)# **vpc** *number*
4. (Optional) switch# **show vpc brief**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

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

This example shows how to configure an EtherChannel that will connect to the downstream device:

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

Manually Configuring a vPC Domain MAC Address

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

Before You Begin

Ensure that you have enabled the vPC feature.

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **system-mac** *mac-address*
4. (Optional) switch# **show vpc role**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

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

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

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

Manually Configuring the System Priority

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

Before You Begin

Ensure that you have enabled the vPC feature.

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **system-priority** *priority*
4. (Optional) switch# **show vpc brief**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default *domain-id* ; the range is from 1 to 1000.
Step 3	switch(config-vpc-domain)# **system-priority** *priority*	Enters the system priority that you want for the specified vPC domain. The range of values is from 1 to 65535. The default value is 32667.
Step 4	switch# **show vpc brief**	(Optional) Displays information about each vPC, including information about the vPC peer link.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a vPC peer link:

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

Manually Configuring a vPC Peer Switch Role

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

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

Before You Begin

Ensure that you have enabled the vPC feature.

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **role priority** *priority*
4. (Optional) switch# **show vpc brief**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default *domain-id* ; the range is from 1 to 1000.
Step 3	switch(config-vpc-domain)# **role priority** *priority*	Enters the role priority that you want for the vPC system priority. The range of values is from 1 to 65535. The default value is 32667.
Step 4	switch# **show vpc brief**	(Optional) Displays information about each vPC, including information about the vPC peer link.
Step 5	switch# **copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a vPC peer link:

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

Verifying the vPC Configuration

Use the following commands to display vPC configuration information:

Command	Purpose
switch# **show feature**	Displays whether vPC is enabled or not.
switch# **show port-channel capacity**	Displays how many EtherChannels are configured and how many are still available on the switch.
switch# **show running-config vpc**	Displays running configuration information for vPCs.
switch# **show vpc brief**	Displays brief information on the vPCs.

Command	Purpose
switch# **show vpc consistency-parameters**	Displays the status of those parameters that must be consistent across all vPC interfaces.
switch# **show vpc peer-keepalive**	Displays information on the peer-keepalive messages.
switch# **show vpc role**	Displays the peer status, the role of the local switch, the vPC system MAC address and system priority, and the MAC address and priority for the local vPC switch.
switch# **show vpc statistics**	Displays statistics on the vPCs. **Note** This command displays the vPC statistics only for the vPC peer device that you are working on.

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

Viewing The Graceful Type-1 Check Status

To view the current status of the graceful Type-1 consistency check, enter the **show vpc brief** command.

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

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

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

Viewing A Global Type-1 Inconsistency

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

Enter the **show vpc** command on the secondary switch to view the status of the suspended vPC VLANs:

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

vPC domain id                     : 10
```

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

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

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

Enter the **show vpc** command on the primary switch to view the inconsistent status (the VLANs on the primary vPC are not suspended):

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

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

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

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

Viewing An Interface-Specific Type-1 Inconsistency

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

Enter the **show vpc brief** command on the secondary switch to view the status of the suspended vPC VLAN:

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

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

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

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

Enter the **show vpc brief** command on the primary switch to view the inconsistent status (the VLANs on the primary vPC are not suspended):

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

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

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

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

Viewing a Per-VLAN Consistency Status

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

This example shows the status of the VLAN in a consistent state before an inconsistency occurs. then, the **no spanning-tree vlan 5** command is entered which triggers the inconsistency between the primary and secondary switch.

The **show vpc brief** command shows the consistent status of the VLANs on the primary and the secondary switches.

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

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

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

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

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

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

The **show vpc brief** command on the secondary switch shows the per-VLAN consistency status as Failed.

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

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

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

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

The **show vpc brief** command on the primary switch also shows the per-VLAN consistency status as Failed.

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

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

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

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

This example shows the inconsistency as STP Disabled.

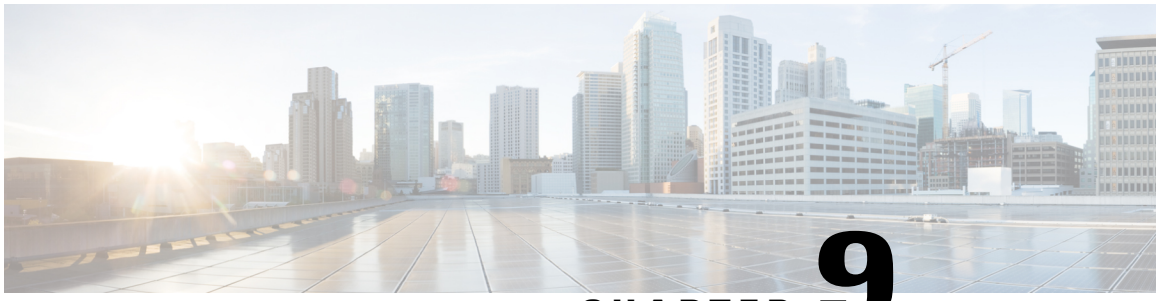
```
switch(config)# show vpc consistency-parameters vlans

Name                        Type  Reason Code             Pass Vlans

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

vPC Default Settings

The following table lists the default settings for vPC parameters.

Table 8: Default vPC Parameters

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



CHAPTER 9

Configuring Rapid PVST+

This chapter contains the following sections:

- Information About Rapid PVST+, page 105
- Configuring Rapid PVST+, page 120
- Verifying Rapid PVST+ Configurations, page 130

Information About Rapid PVST+

The Rapid PVST+ protocol is the IEEE 802.1w standard, Rapid Spanning Tree Protocol (RSTP), implemented on a per VLAN basis. Rapid PVST+ interoperates with the IEEE 802.1D standard, which mandates a single STP instance for all VLANs, rather than per VLAN.

Rapid PVST+ is enabled by default on the default VLAN (VLAN1) and on all newly created VLANs in software. Rapid PVST+ interoperates with switches that run legacy IEEE 802.1D STP.

RSTP is an improvement on the original STP standard, 802.1D, which allows faster convergence.

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

Understanding STP

STP Overview

For an Ethernet network to function properly, only one active path can exist between any two stations.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The STP algorithm calculates the best loop-free path throughout a switched network. LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Switches do not forward these frames, but use the frames to construct a loop-free path.

Multiple active paths between end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages and switches might learn end station MAC addresses on multiple LAN ports. These conditions result in a broadcast storm, which creates an unstable network.

STP defines a tree with a root bridge and a loop-free path from the root to all switches in the network. STP forces redundant data paths into a blocked state. If a network segment in the spanning tree fails and a redundant path exists, the STP algorithm recalculates the spanning tree topology and activates the blocked path.

When two LAN ports on a switch are part of a loop, the STP port priority and port path cost setting determine which port on the switch is put in the forwarding state and which port is put in the blocking state.

Understanding How a Topology is Created

All switches in an extended LAN that participate in a spanning tree gather information about other switches in the network by exchanging of BPDUs. This exchange of BPDUs results in the following actions:

- The system elects a unique root switch for the spanning tree network topology.
- The system elects a designated switch for each LAN segment.
- The system eliminates any loops in the switched network by placing redundant interfaces in a backup state; all paths that are not needed to reach the root switch from anywhere in the switched network are placed in an STP-blocked state.

The topology on an active switched network is determined by the following:

- The unique switch identifier Media Access Control (MAC) address of the switch that is associated with each switch
- The path cost to the root that is associated with each interface
- The port identifier that is associated with each interface

In a switched network, the root switch is the logical center of the spanning tree topology. STP uses BPDUs to elect the root switch and root port for the switched network, as well as the root port and designated port for each switched segment.

Understanding the Bridge ID

Each VLAN on each switch has a unique 64-bit bridge ID consisting of a bridge priority value, an extended system ID (IEEE 802.1t), and an STP MAC address allocation.

Bridge Priority Value

The bridge priority is a 4-bit value when the extended system ID is enabled.

Note In Cisco NX-OS, the extended system ID is always enabled; you cannot be disable the extended system ID.

Extended System ID

A 12-bit extended system ID field is part of the bridge ID.

Figure 9: Bridge ID with Extended System ID

The switches always use the 12-bit extended system ID.

Combined with the bridge ID, the system ID extension functions as the unique identifier for a VLAN.

Table 9: Bridge Priority Value and Extended System ID with the Extended System ID Enabled

Bridge Priority Value				Extended System ID (Set Equal to the VLAN ID)										
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

STP MAC Address Allocation

Note Extended system ID and MAC address reduction is always enabled on the software.

With MAC address reduction enabled on any switch, you should also enable MAC address reduction on all other connected switches to avoid undesirable root bridge election and spanning tree topology issues.

When MAC address reduction is enabled, the root bridge priority becomes a multiple of 4096 plus the VLAN ID. You can only specify a switch bridge ID (used by the spanning tree algorithm to determine the identity of the root bridge, the lowest being preferred) as a multiple of 4096. Only the following values are possible:

- 0
- 4096
- 8192
- 12288
- 16384
- 20480
- 24576
- 28672
- 32768

- 36864
- 40960
- 45056
- 49152
- 53248
- 57344
- 61440

STP uses the extended system ID plus a MAC address to make the bridge ID unique for each VLAN.

Note If another bridge in the same spanning tree domain does not run the MAC address reduction feature, it could achieve root bridge ownership because its bridge ID may fall between the values specified by the MAC address reduction feature.

Understanding BPDUs

Switches transmit bridge protocol data units (BPDUs) throughout the STP instance. Each switch sends configuration BPDUs to communicate and compute the spanning tree topology. Each configuration BPDU contains the following minimal information:

- The unique bridge ID of the switch that the transmitting switch determines is the root bridge
- The STP path cost to the root
- The bridge ID of the transmitting bridge
- Message age
- The identifier of the transmitting port
- Values for the hello, forward delay, and max-age protocol timer
- Additional information for STP extension protocols

When a switch transmits a Rapid PVST+ BPDU frame, all switches connected to the VLAN on which the frame is transmitted receive the BPDU. When a switch receives a BPDU, it does not forward the frame but instead uses the information in the frame to calculate a BPDU, and, if the topology changes, initiate a BPDU transmission.

A BPDU exchange results in the following:

- One switch is elected as the root bridge.
- The shortest distance to the root bridge is calculated for each switch based on the path cost.
- A designated bridge for each LAN segment is selected. This is the switch closest to the root bridge through which frames are forwarded to the root.
- A root port is selected. This is the port providing the best path from the bridge to the root bridge.
- Ports included in the spanning tree are selected.

Related Topics

Rapid PVST+ BPDUs, on page 111

Election of the Root Bridge

For each VLAN, the switch with the lowest numerical value of the bridge ID is elected as the root bridge. If all switches are configured with the default priority (32768), the switch with the lowest MAC address in the VLAN becomes the root bridge. The bridge priority value occupies the most significant bits of the bridge ID.

When you change the bridge priority value, you change the probability that the switch will be elected as the root bridge. Configuring a lower value increases the probability; a higher value decreases the probability.

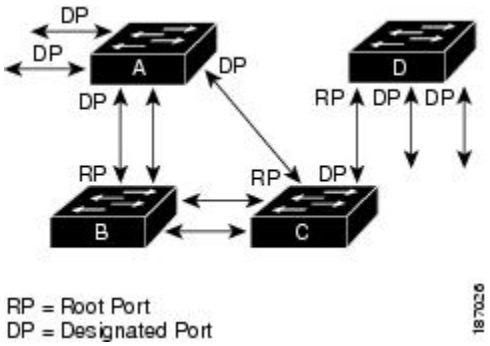
The STP root bridge is the logical center of each spanning tree topology in a network. All paths that are not needed to reach the root bridge from anywhere in the network are placed in STP blocking mode.

BPDUs contain information about the transmitting bridge and its ports, including bridge and MAC addresses, bridge priority, port priority, and path cost. STP uses this information to elect the root bridge for the STP instance, to elect the root port leading to the root bridge, and to determine the designated port for each segment.

Creating the Spanning Tree Topology

In the following figure, Switch A is elected as the root bridge because the bridge priority of all the switches is set to the default (32768) and Switch A has the lowest MAC address. However, due to traffic patterns, number of forwarding ports, or link types, Switch A might not be the ideal root bridge. By increasing the priority (lowering the numerical value) of the ideal switch so that it becomes the root bridge, you force an STP recalculation to form a new spanning tree topology with the ideal switch as the root.

Figure 10: Spanning Tree Topology



When the spanning tree topology is calculated based on default parameters, the path between source and destination end stations in a switched network might not be ideal. For instance, connecting higher-speed links to a port that has a higher number than the current root port can cause a root-port change. The goal is to make the fastest link the root port.

For example, assume that one port on Switch B is a fiber-optic link, and another port on Switch B (an unshielded twisted-pair [UTP] link) is the root port. Network traffic might be more efficient over the high-speed fiber-optic link. By changing the STP port priority on the fiber-optic port to a higher priority (lower numerical value) than the root port, the fiber-optic port becomes the new root port.

Understanding Rapid PVST+

Rapid PVST+ Overview

Rapid PVST+ is the IEEE 802.1w (RSTP) standard implemented per VLAN. A single instance of STP runs on each configured VLAN (if you do not manually disable STP). Each Rapid PVST+ instance on a VLAN has a single root switch. You can enable and disable STP on a per-VLAN basis when you are running Rapid PVST+.

Note Rapid PVST+ is the default STP mode for the switch.

Rapid PVST+ uses point-to-point wiring to provide rapid convergence of the spanning tree. The spanning tree reconfiguration can occur in less than 1 second with Rapid PVST+ (in contrast to 50 seconds with the default settings in the 802.1D STP).

Note Rapid PVST+ supports one STP instance for each VLAN.

Using Rapid PVST+, STP convergence occurs rapidly. Each designated or root port in the STP sends out a BPDU every 2 seconds by default. On a designated or root port in the topology, if hello messages are missed three consecutive times, or if the maximum age expires, the port immediately flushes all protocol information in the table. A port considers that it loses connectivity to its direct neighbor root or designated port if it misses three BPDUs or if the maximum age expires. This rapid aging of the protocol information allows quick failure detection. The switch automatically checks the PVID.

Rapid PVST+ provides for rapid recovery of connectivity following the failure of a network device, a switch port, or a LAN. It provides rapid convergence for edge ports, new root ports, and ports connected through point-to-point links as follows:

- Edge ports—When you configure a port as an edge port on an RSTP switch, the edge port immediately transitions to the forwarding state. (This immediate transition was previously a Cisco-proprietary feature named PortFast.) You should only configure on ports that connect to a single end station as edge ports. Edge ports do not generate topology changes when the link changes.

  Enter the **spanning-tree port type** interface configuration command to configure a port as an STP edge port.

  Note We recommend that you configure all ports connected to a host as edge ports.

- Root ports—If Rapid PVST+ selects a new root port, it blocks the old root port and immediately transitions the new root port to the forwarding state.
- Point-to-point links—If you connect a port to another port through a point-to-point link and the local port becomes a designated port, it negotiates a rapid transition with the other port by using the proposal-agreement handshake to ensure a loop-free topology.

Rapid PVST+ achieves rapid transition to the forwarding state only on edge ports and point-to-point links. Although the link type is configurable, the system automatically derives the link type information from the

duplex setting of the port. Full-duplex ports are assumed to be point-to-point ports, while half-duplex ports are assumed to be shared ports.

Edge ports do not generate topology changes, but all other designated and root ports generate a topology change (TC) BPDU when they either fail to receive three consecutive BPDUs from the directly connected neighbor or the maximum age times out. At this point, the designated or root port sends out a BPDU with the TC flag set. The BPDUs continue to set the TC flag as long as the TC While timer runs on that port. The value of the TC While timer is the value set for the hello time plus 1 second. The initial detector of the topology change immediately floods this information throughout the entire topology.

When Rapid PVST+ detects a topology change, the protocol does the following:

- Starts the TC While timer with a value equal to twice the hello time for all the non-edge root and designated ports, if necessary.
- Flushes the MAC addresses associated with all these ports.

The topology change notification floods quickly across the entire topology. The system flushes dynamic entries immediately on a per-port basis when it receives a topology change.

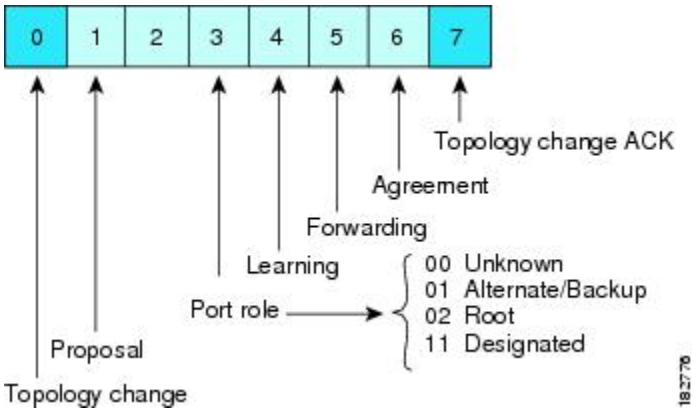
Note The TCA flag is used only when the switch is interacting with switches that are running legacy 802.1D STP.

The proposal and agreement sequence then quickly propagates toward the edge of the network and quickly restores connectivity after a topology change.

Rapid PVST+ BPDUs

Rapid PVST+ and 802.1w use all six bits of the flag byte to add the role and state of the port that originates the BPDU, and the proposal and agreement handshake. The following figure shows the use of the BPDU flags in Rapid PVST+.

Figure 11: Rapid PVST+ Flag Byte in BPDU

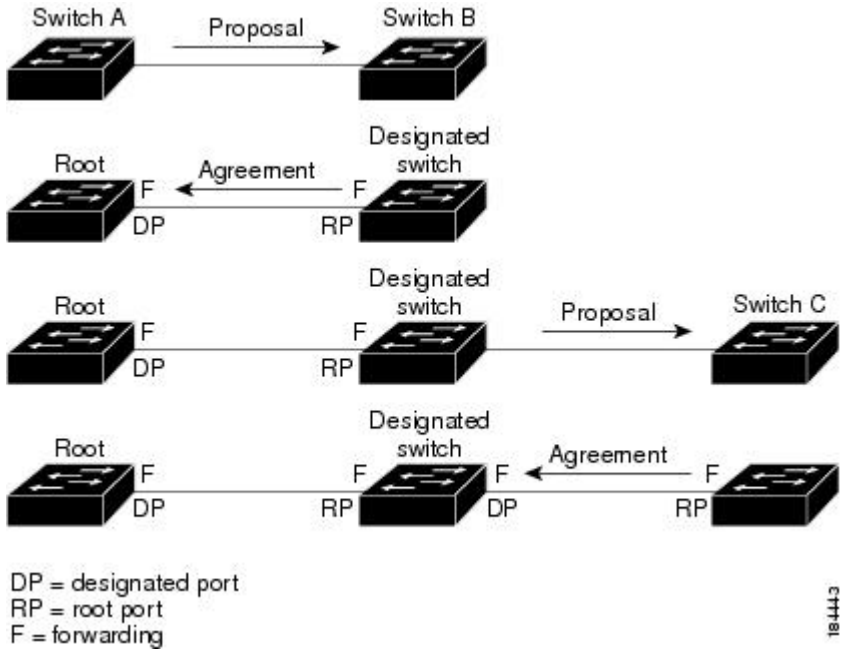


Another important change is that the Rapid PVST+ BPDU is type 2, version 2, which makes it possible for the switch to detect connected legacy (802.1D) bridges. The BPDU for 802.1D is version 0.

Proposal and Agreement Handshake

As shown in the following figure, switch A is connected to switch B through a point-to-point link, and all of the ports are in the blocking state. Assume that the priority of switch A is a smaller numerical value than the priority of switch B.

Figure 12: Proposal and Agreement Handshaking for Rapid Convergence



Switch A sends a proposal message (a configuration BPDU with the proposal flag set) to switch B, proposing itself as the designated switch.

After receiving the proposal message, switch B selects as its new root port the port from which the proposal message was received, forces all non-edge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

After receiving the agreement message from switch B, switch A also immediately transitions its designated port to the forwarding state. No loops in the network can form because switch B blocked all of its non-edge ports and because there is a point-to-point link between switches A and B.

When switch C connects to switch B, a similar set of handshaking messages are exchanged. Switch C selects the port connected to switch B as its root port, and both ends of the link immediately transition to the forwarding state. With each iteration of this handshaking process, one more network device joins the active topology. As the network converges, this proposal-agreement handshaking progresses from the root toward the leaves of the spanning tree.

The switch learns the link type from the port duplex mode: a full-duplex port is considered to have a point-to-point connection and a half-duplex port is considered to have a shared connection. You can override the default setting that is controlled by the duplex setting by entering the **spanning-tree link-type** interface configuration command.

This proposal/agreement handshake is initiated only when a non-edge port moves from the blocking to the forwarding state. The handshaking process then proliferates step-by-step throughout the topology.

Related Topics

Summary of Port States, on page 116

Protocol Timers

The following table describes the protocol timers that affect the Rapid PVST+ performance.

Table 10: Rapid PVST+ Protocol Timers

Variable	Description
Hello timer	Determines how often each switch broadcasts BPDUs to other switches. The default is 2 seconds, and the range is from 1 to 10.
Forward delay timer	Determines how long each of the listening and learning states last before the port begins forwarding. This timer is generally not used by the protocol but is used as a backup. The default is 15 seconds, and the range is from 4 to 30 seconds.
Maximum age timer	Determines the amount of time protocol information received on an port is stored by the switch. This timer is generally not used by the protocol, but it is used when interoperating with 802.1D spanning tree. The default is 20 seconds; the range is from 6 to 40 seconds.

Port Roles

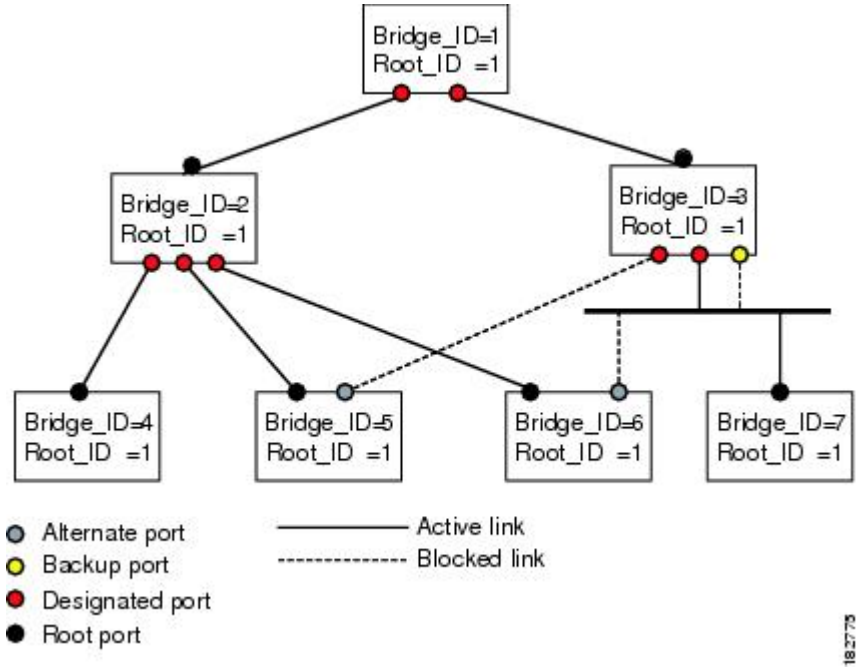
Rapid PVST+ provides rapid convergence of the spanning tree by assigning port roles and learning the active topology. Rapid PVST+ builds upon the 802.1D STP to select the switch with the highest priority (lowest numerical priority value) as the root bridge. Rapid PVST+ then assigns one of these port roles to individual ports:

- Root port—Provides the best path (lowest cost) when the switch forwards packets to the root bridge.
- Designated port—Connects to the designated switch, which incurs the lowest path cost when forwarding packets from that LAN to the root bridge. The port through which the designated switch is attached to the LAN is called the designated port.
- Alternate port—Offers an alternate path toward the root bridge to the path provided by the current root port. An alternate port provides a path to another switch in the topology.
- Backup port—Acts as a backup for the path provided by a designated port toward the leaves of the spanning tree. A backup port can exist only when two ports are connected in a loopback by a point-to-point link or when a switch has two or more connections to a shared LAN segment. A backup port provides another path in the topology to the switch.
- Disabled port—Has no role within the operation of the spanning tree.

In a stable topology with consistent port roles throughout the network, Rapid PVST+ ensures that every root port and designated port immediately transition to the forwarding state while all alternate and backup ports are always in the blocking state. Designated ports start in the blocking state. The port state controls the operation of the forwarding and learning processes.

A port with the root or a designated port role is included in the active topology. A port with the alternate or backup port role is excluded from the active topology (see the following figure).

Figure 13: Sample Topology Demonstrating Port Roles



Related Topics

Election of the Root Bridge, on page 109

Port States

Rapid PVST+ Port State Overview

Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When a LAN port transitions directly from nonparticipation in the spanning tree topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate through the switched LAN before starting to forward frames.

Each LAN port on a software using Rapid PVST+ or MST exists in one of the following four states:

- Blocking—The LAN port does not participate in frame forwarding.
- Learning—The LAN port prepares to participate in frame forwarding.
- Forwarding—The LAN port forwards frames.

- Disabled—The LAN port does not participate in STP and is not forwarding frames.

When you enable Rapid PVST+, every port in the software, VLAN, and network goes through the blocking state and the transitory states of learning at power up. If properly configured, each LAN port stabilizes to the forwarding or blocking state.

When the STP algorithm places a LAN port in the forwarding state, the following process occurs:

- The LAN port is put into the blocking state while it waits for protocol information that suggests it should go to the learning state.
- The LAN port waits for the forward delay timer to expire, moves the LAN port to the learning state, and restarts the forward delay timer.
- In the learning state, the LAN port continues to block frame forwarding as it learns the end station location information for the forwarding database.
- The LAN port waits for the forward delay timer to expire and then moves the LAN port to the forwarding state, where both learning and frame forwarding are enabled.

Blocking State

A LAN port in the blocking state does not participate in frame forwarding.

A LAN port in the blocking state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate the end station location into its address database. (There is no learning on a blocking LAN port, so there is no address database update.)
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to network management messages.

Learning State

A LAN port in the learning state prepares to participate in frame forwarding by learning the MAC addresses for the frames. The LAN port enters the learning state from the blocking state.

A LAN port in the learning state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Incorporates the end station location into its address database.
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to network management messages.

Forwarding State

A LAN port in the forwarding state forwards frames. The LAN port enters the forwarding state from the learning state.

A LAN port in the forwarding state performs as follows:

- Forwards frames received from the attached segment.
- Forwards frames switched from another port for forwarding.
- Incorporates the end station location information into its address database.
- Receives BPDUs and directs them to the system module.
- Processes BPDUs received from the system module.
- Receives and responds to network management messages.

Disabled State

A LAN port in the disabled state does not participate in frame forwarding or STP. A LAN port in the disabled state is virtually nonoperational.

A disabled LAN port performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate the end station location into its address database. (There is no learning, so there is no address database update.)
- Does not receive BPDUs from neighbors.
- Does not receive BPDUs for transmission from the system module.

Summary of Port States

The following table lists the possible operational and Rapid PVST+ states for ports and the corresponding inclusion in the active topology.

Table 11: Port State Active Topology

Operational Status	Port State	Is Port Included in the Active Topology?
Enabled	Blocking	No
Enabled	Learning	Yes
Enabled	Forwarding	Yes
Disabled	Disabled	No

Synchronization of Port Roles

When the switch receives a proposal message on one of its ports and that port is selected as the new root port, Rapid PVST+ forces all other ports to synchronize with the new root information.

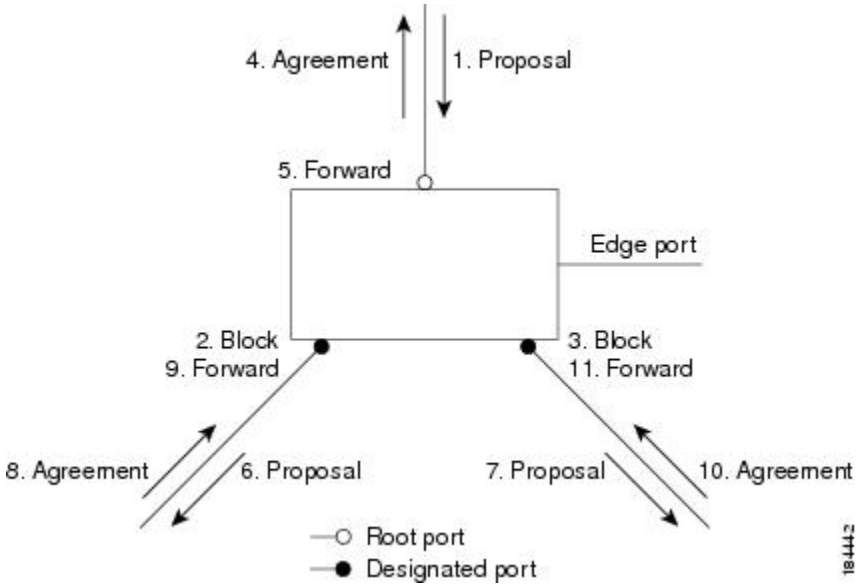
The switch is synchronized with superior root information received on the root port if all other ports are synchronized. An individual port on the switch is synchronized if either of the following applies:

- That port is in the blocking state.
- It is an edge port (a port configured to be at the edge of the network).

If a designated port is in the forwarding state and is not configured as an edge port, it transitions to the blocking state when the Rapid PVST+ forces it to synchronize with new root information. In general, when the Rapid PVST+ forces a port to synchronize with root information and the port does not satisfy any of the above conditions, its port state is set to blocking.

After ensuring that all of the ports are synchronized, the switch sends an agreement message to the designated switch that corresponds to its root port. When the switches connected by a point-to-point link are in agreement about their port roles, Rapid PVST+ immediately transitions the port states to the forwarding state. The sequence of events is shown in the following figure.

Figure 14: Sequence of Events During Rapid Convergence



Processing Superior BPDU Information

A superior BPDU is a BPDU with root information (such as a lower switch ID or lower path cost) that is superior to what is currently stored for the port.

If a port receives a superior BPDU, Rapid PVST+ triggers a reconfiguration. If the port is proposed and is selected as the new root port, Rapid PVST+ forces all the other ports to synchronize.

If the received BPDU is a Rapid PVST+ BPDU with the proposal flag set, the switch sends an agreement message after all of the other ports are synchronized. The new root port transitions to the forwarding state as soon as the previous port reaches the blocking state.

If the superior information received on the port causes the port to become a backup port or an alternate port, Rapid PVST+ sets the port to the blocking state and sends an agreement message. The designated port continues sending BPDUs with the proposal flag set until the forward-delay timer expires. At that time, the port transitions to the forwarding state.

Processing Inferior BPDU Information

An inferior BPDU is a BPDU with root information (such as a higher switch ID or higher path cost) that is inferior to what is currently stored for the port.

If a designated port receives an inferior BPDU, it immediately replies with its own information.

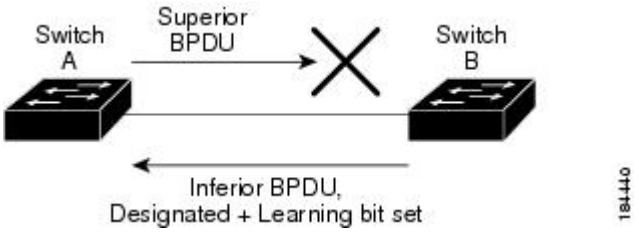
Spanning-Tree Dispute Mechanism

The software checks the consistency of the port role and state in the received BPDUs to detect unidirectional link failures that could cause bridging loops.

When a designated port detects a conflict, it keeps its role, but reverts to a discarding state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

The following figure illustrates a unidirectional link failure that typically creates a bridging loop. Switch A is the root bridge, and its BPDUs are lost on the link leading to switch B. The 802.1w-standard BPDUs include the role and state of the sending port. With this information, switch A can detect that switch B does not react to the superior BPDUs it sends and that switch B is the designated, not root port. As a result, switch A blocks (or keeps blocking) its port, thus preventing the bridging loop. The block is shown as an STP dispute.

Figure 15: Detecting Unidirectional Link Failure



Port Cost

Note Rapid PVST+ uses the short (16-bit) pathcost method to calculate the cost by default. With the short pathcost method, you can assign any value in the range of 1 to 65535. However, you can configure the switch to use the long (32-bit) pathcost method, which allows you to assign any value in the range of 1 to 200,000,000. You configure the pathcost calculation method globally.

The STP port path-cost default value is determined from the media speed and path-cost calculation method of a LAN interface. If a loop occurs, STP considers the port cost when selecting a LAN interface to put into the forwarding state.

Table 12: Default Port Cost

Bandwidth	Short Path-cost Method of Port Cost	Long Path-cost Method of Port Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gigabit Ethernet	4	20,000
10 Gigabit Ethernet	2	2,000

You can assign lower cost values to LAN interfaces that you want STP to select first and higher cost values to LAN interfaces that you want STP to select last. If all LAN interfaces have the same cost value, STP puts the LAN interface with the lowest LAN interface number in the forwarding state and blocks other LAN interfaces.

On access ports, you assign port cost by the port. On trunk ports, you assign the port cost by the VLAN; you can configure the same port cost to all the VLANs on a trunk port.

Port Priority

If a loop occurs and multiple ports have the same path cost, Rapid PVST+ considers the port priority when selecting which LAN port to put into the forwarding state. You can assign lower priority values to LAN ports that you want Rapid PVST+ to select first and higher priority values to LAN ports that you want Rapid PVST+ to select last.

If all LAN ports have the same priority value, Rapid PVST+ puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports. The possible priority range is from 0 through 224 (the default is128), configurable in increments of 32. software uses the port priority value when the LAN port is configured as an access port and uses VLAN port priority values when the LAN port is configured as a trunk port.

Rapid PVST+ and IEEE 802.1Q Trunks

In a network of Cisco switches connected through 802.1Q trunks, the switches maintain one instance of STP for each VLAN allowed on the trunks. However, non-Cisco 802.1Q switches maintain only one instance of STP for all VLANs allowed on the trunks.

When you connect a Cisco switch to a non-Cisco switch through an 802.1Q trunk, the Cisco switch combines the STP instance of the 802.1Q VLAN of the trunk with the STP instance of the non-Cisco 802.1Q switch. However, all per-VLAN STP information that is maintained by Cisco switches is separated by a cloud of non-Cisco 802.1Q switches. The non-Cisco 802.1Q cloud that separates the Cisco switches is treated as a single trunk link between the switches.

Rapid PVST+ Interoperation with Legacy 802.1D STP

Rapid PVST+ can interoperate with switches that are running the legacy 802.1D protocol. The switch knows that it is interoperating with equipment running 802.1D when it receives a BPDU version 0. The BPDUs for Rapid PVST+ are version 2. If the BPDU received is an 802.1w BPDU version 2 with the proposal flag set, the switch sends an agreement message after all of the other ports are synchronized. If the BPDU is an 802.1D BPDU version 0, the switch does not set the proposal flag and starts the forward-delay timer for the port. The new root port requires twice the forward-delay time to transition to the forwarding state.

The switch interoperates with legacy 802.1D switches as follows:

- Notification—Unlike 802.1D BPDUs, 802.1w does not use TCN BPDUs. However, for interoperability with 802.1D switches, Cisco NX-OS processes and generates TCN BPDUs.
- Acknowledgement—When an 802.1w switch receives a TCN message on a designated port from an 802.1D switch, it replies with an 802.1D configuration BPDU with the TCA bit set. However, if the TC-while timer (the same as the TC timer in 802.1D) is active on a root port connected to an 802.1D switch and a configuration BPDU with the TCA set is received, the TC-while timer is reset.

This method of operation is required only for 802.1D switches. The 802.1w BPDUs do not have the TCA bit set.

- Protocol migration—For backward compatibility with 802.1D switches, 802.1w selectively sends 802.1D configuration BPDUs and TCN BPDUs on a per-port basis.

When a port is initialized, the migrate-delay timer is started (specifies the minimum time during which 802.1w BPDUs are sent), and 802.1w BPDUs are sent. While this timer is active, the switch processes all BPDUs received on that port and ignores the protocol type.

If the switch receives an 802.1D BPDU after the port migration-delay timer has expired, it assumes that it is connected to an 802.1D switch and starts using only 802.1D BPDUs. However, if the 802.1w switch is using 802.1D BPDUs on a port and receives an 802.1w BPDU after the timer has expired, it restarts the timer and starts using 802.1w BPDUs on that port.

Note If you want all switches to renegotiate the protocol, you must restart Rapid PVST+.

Rapid PVST+ Interoperation with 802.1s MST

Rapid PVST+ interoperates seamlessly with the IEEE 802.1s Multiple Spanning Tree (MST) standard. No user configuration is needed.

Configuring Rapid PVST+

Rapid PVST+, which has the 802.1w standard applied to the Rapid PVST+ protocol, is the default STP setting in the software.

You enable Rapid PVST+ on a per-VLAN basis. The software maintains a separate instance of STP for each VLAN (except on those VLANS on which you disable STP). By default, Rapid PVST+ is enabled on the default VLAN and on each VLAN that you create.

Enabling Rapid PVST+

Once you enable Rapid PVST+ on the switch, you must enable Rapid PVST+ on the specified VLANs.

Rapid PVST+ is the default STP mode. You cannot simultaneously run MST and Rapid PVST+.

Note Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mode rapid-pvst**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mode rapid-pvst**	Enables Rapid PVST+ on the switch. Rapid PVST+ is the default spanning tree mode. **Note** Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

This example shows how to enable Rapid PVST+ on the switch:

```
switch# configure terminal
switch(config)# spanning-tree mode rapid-pvst
```

Note Because STP is enabled by default, entering the **show running-config** command to view the resulting configuration does not display the command that you entered to enable Rapid PVST+.

Enabling Rapid PVST+ per VLAN

You can enable or disable Rapid PVST+ on each VLAN.

Note Rapid PVST+ is enabled by default on the default VLAN and on all VLANs that you create.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree** *vlan-range*
3. (Optional) switch(config)# **no spanning-tree** *vlan-range*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree** *vlan-range*	Enables Rapid PVST+ (default STP) on a per VLAN basis. The *vlan-range* value can be 2 through 4094 (except reserved VLAN values).
Step 3	switch(config)# **no spanning-tree** *vlan-range*	(Optional) Disables Rapid PVST+ on the specified VLAN. **Caution** Do not disable spanning tree on a VLAN unless all switches and bridges in the VLAN have spanning tree disabled. You cannot disable spanning tree on some of the switches and bridges in a VLAN and leave it enabled on other switches and bridges. This action can have unexpected results because switches and bridges with spanning tree enabled will have incomplete information regarding the physical topology of the network. Do not disable spanning tree in a VLAN without ensuring that there are no physical loops present in the VLAN. Spanning tree serves as a safeguard against misconfigurations and cabling errors.

This example shows how to enable STP on a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5
```

Configuring the Root Bridge ID

The software maintains a separate instance of STP for each active VLAN in Rapid PVST+. For each VLAN, the switch with the lowest bridge ID becomes the root bridge for that VLAN.

To configure a VLAN instance to become the root bridge, modify the bridge priority from the default value (32768) to a significantly lower value.

When you enter the **spanning-tree vlan** *vlan_ID* **root** command, the switch checks the bridge priority of the current root bridges for each VLAN. The switch sets the bridge priority for the specified VLANs to 24576 if this value will cause the switch to become the root for the specified VLANs. If any root bridge for the

specified VLANs has a bridge priority lower than 24576, the switch sets the bridge priority for the specified VLANs to 4096 less than the lowest bridge priority.

Note The **spanning-tree vlan** *vlan_ID* **root** command fails if the value required to be the root bridge is less than 1.

Caution The root bridge for each instance of STP should be a backbone or distribution switch. Do not configure an access switch as the STP primary root.

Enter the **diameter** keyword to specify the network diameter (that is, the maximum number of bridge hops between any two end stations in the network). When you specify the network diameter, the software automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can enter the **hello-time** keyword to override the automatically calculated hello time.

Note With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the **spanning-tree mst hello-time**, **spanning-tree mst forward-time**, and **spanning-tree mst max-age** configuration commands.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan** *vlan-range* **root primary** [**diameter** *dia* [**hello-time** *hello-time*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **root primary** [**diameter** *dia* [**hello-time** *hello-time*]]	Configures a software switch as the primary root bridge. The *vlan-range* value can be 2 through 4094 (except reserved VLAN values.) The *dia* default is 7. The *hello-time* can be from 1 to 10 seconds, and the default value is 2 seconds.

This example shows how to configure the switch as the root bridge for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 root primary diameter 4
```

Configuring a Secondary Root Bridge

When you configure a software switch as the secondary root, the STP bridge priority is modified from the default value (32768) so that the switch is likely to become the root bridge for the specified VLANs if the primary root bridge fails (assuming the other switches in the network use the default bridge priority of 32768). STP sets the bridge priority to 28672.

Enter the **diameter** keyword to specify the network diameter (that is, the maximum number of bridge hops between any two end stations in the network). When you specify the network diameter, the software automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can enter the **hello-time** keyword to override the automatically calculated hello time.

You configure more than one switch in this manner to have multiple backup root bridges. Enter the same network diameter and hello time values that you used when configuring the primary root bridge.

Note With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the **spanning-tree mst hello-time**, **spanning-tree mst forward-time**, and **spanning-tree mst max-age** global configuration commands.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan** *vlan-range* **root secondary** [**diameter** *dia* [**hello-time** *hello-time*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **root secondary** [**diameter** *dia* [**hello-time** *hello-time*]]	Configures a software switch as the secondary root bridge. The *vlan-range* value can be 2 through 4094 (except reserved VLAN values.) The *dia* default is 7. The *hello-time* can be from 1 to 10 seconds, and the default value is 2 seconds.

This example shows how to configure the switch as the secondary root bridge for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 root secondary diameter 4
```

Configuring the Rapid PVST+ Port Priority

You can assign lower priority values to LAN ports that you want Rapid PVST+ to select first and higher priority values to LAN ports that you want Rapid PVST+ to select last. If all LAN ports have the same priority

value, Rapid PVST+ puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports.

The software uses the port priority value when the LAN port is configured as an access port and uses VLAN port priority values when the LAN port is configured as a trunk port.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **spanning-tree** [**vlan** *vlan-list*] **port-priority** *priority*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **spanning-tree** [**vlan** *vlan-list*] **port-priority** *priority*	Configures the port priority for the LAN interface. The *priority* value can be from 0 to 224. The lower the value, the higher the priority. The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. All other values are rejected. The default value is 128.

This example shows how to configure the access port priority of an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port-priority 160
```

You can only apply this command to a physical Ethernet interface.

Configuring the Rapid PVST+ Pathcost Method and Port Cost

On access ports, you assign port cost by the port. On trunk ports, you assign the port cost by VLAN; you can configure the same port cost on all the VLANs on a trunk.

Note In Rapid PVST+ mode, you can use either the short or long pathcost method, and you can configure the method in either the interface or configuration submode.The default pathcost method is short.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree pathcost method** {**long** | **short**}
3. switch(config)# **interface** *type slot*/*port*
4. switch(config-if)# **spanning-tree** [**vlan** *vlan-id*] **cost** [*value* | **auto**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree pathcost method** {**long**	**short**}	Selects the method used for Rapid PVST+ pathcost calculations. The default method is the short method.
Step 3	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters interface configuration mode.
Step 4	switch(config-if)# **spanning-tree** [**vlan** *vlan-id*] **cost** [*value*	**auto**]	Configures the port cost for the LAN interface. The cost value, depending on the pathcost calculation method, can be as follows: • short—1 to 65535 • long—1 to 200000000 **Note** You configure this parameter per interface on access ports and per VLAN on trunk ports. The default is **auto** , which sets the port cost on both the pathcost calculation method and the media speed.

This example shows how to configure the access port cost of an Ethernet interface:

```
switch# configure terminal
switch (config)# spanning-tree pathcost method long
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree cost 1000
```

You can only apply this command to a physical Ethernet interface.

Configuring the Rapid PVST+ Bridge Priority of a VLAN

You can configure the Rapid PVST+ bridge priority of a VLAN.

Note Be careful when using this configuration. For most situations, we recommend that you configure the primary root and secondary root to modify the bridge priority.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan** *vlan-range* **priority** *value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **priority** *value*	Configures the bridge priority of a VLAN. Valid values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. All other values are rejected. The default value is 32768.

This example shows how to configure the bridge priority of a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 priority 8192
```

Configuring the Rapid PVST+ Hello Time for a VLAN

You can configure the Rapid PVST+ hello time for a VLAN.

Note Be careful when using this configuration. For most situations, we recommend that you configure the primary root and secondary root to modify the hello time.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan** *vlan-range* **hello-time** *hello-time*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **hello-time** *hello-time*	Configures the hello time of a VLAN. The hello time value can be from 1 to 10 seconds. The default is 2 seconds.

This example shows how to configure the hello time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 hello-time 7
```

Configuring the Rapid PVST+ Forward Delay Time for a VLAN

You can configure the forward delay time per VLAN when using Rapid PVST+.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan** *vlan-range* **forward-time** *forward-time*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **forward-time** *forward-time*	Configures the forward delay time of a VLAN. The forward delay time value can be from 4 to 30 seconds, and the default is 15 seconds.

This example shows how to configure the forward delay time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 forward-time 21
```

Configuring the Rapid PVST+ Maximum Age Time for a VLAN

You can configure the maximum age time per VLAN when using Rapid PVST+.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan** *vlan-range* **max-age** *max-age*

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	switch(config)# **spanning-tree vlan** *vlan-range* **max-age** *max-age*	Configures the maximum aging time of a VLAN. The maximum aging time value can be from 6 to 40 seconds, and the default is 20 seconds.

This example shows how to configure the maximum aging time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 max-age 36
```

Specifying the Link Type

Rapid connectivity (802.1w standard) is established only on point-to-point links. By default, the link type is controlled from the duplex mode of the interface. A full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection.

If you have a half-duplex link physically connected point-to-point to a single port on a remote switch, you can override the default setting on the link type and enable rapid transitions.

If you set the link to shared, STP moves back to 802.1D.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **spanning-tree link-type** {**auto** | **point-to-point** | **shared**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **spanning-tree link-type** {**auto** | **point-to-point** | **shared**}	Configures the link type to be either a point-to-point link or shared link. The system reads the default value from the switch connection, as follows: half duplex links are shared and full-duplex links are point-to-point. If the link type is shared, the STP reverts to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface.

This example shows how to configure the link type as a point-to-point link:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
```

You can only apply this command to a physical Ethernet interface.

Restarting the Protocol

A bridge running Rapid PVST+ can send 802.1D BPDUs on one of its ports when it is connected to a legacy bridge. However, the STP protocol migration cannot determine whether the legacy switch has been removed from the link unless the legacy switch is the designated switch. You can restart the protocol negotiation (force the renegotiation with neighboring switches) on the entire switch or on specified interfaces.

Command	Purpose
switch# **clear spanning-tree detected-protocol** [**interface** *interface* [*interface-num* | *port-channel*]]	Restarts Rapid PVST+ on all interfaces on the switch or specified interfaces.

The following example shows how to restart Rapid PVST+ on an Ethernet interface:

```
switch# clear spanning-tree detected-protocol interface ethernet 1/8
```

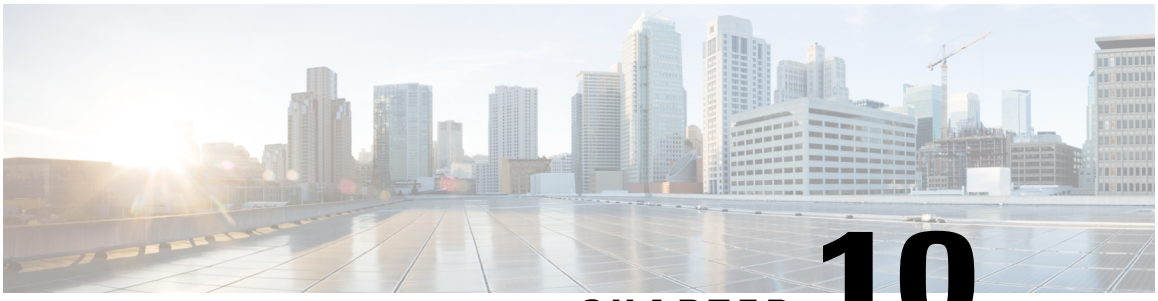
Verifying Rapid PVST+ Configurations

To display Rapid PVST+ configuration information, perform one of these tasks:

Command	Purpose
switch# **show running-config spanning-tree** [**all**]	Displays the current spanning tree configuration.
switch# **show spanning-tree** [*options*]	Displays selected detailed information for the current spanning tree configuration.

This example shows how to display spanning tree status:

```
switch# show spanning-tree brief

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32768
             Address     001c.b05a.5447
             Cost        2
             Port        131 (Ethernet1/3)
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     000d.ec6d.7841
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Eth1/3           Root FWD 2         128.131  P2p Peer(STP)
```



CHAPTER 10

Configuring Multiple Spanning Tree

This chapter contains the following sections:

- Information About MST, page 131
- Configuring MST, page 139
- Verifying MST Configurations, page 157

Information About MST

MST Overview

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

MST maps multiple VLANs into a spanning tree instance, with each instance having a spanning tree topology independent of other spanning tree instances. This architecture provides multiple forwarding paths for data traffic, enables load balancing, and reduces the number of STP instances required to support a large number of VLANs.

MST provides rapid convergence through explicit handshaking as each MST instance uses the IEEE 802.1w standard, which eliminates the 802.1D forwarding delay and quickly transitions root bridge ports and designated ports to the forwarding state.

MAC address reduction is always enabled while you are using MST. You cannot disable this feature.

MST improves spanning tree operation and maintains backward compatibility with these STP versions:

- Original 802.1D spanning tree
- Rapid per-VLAN spanning tree (Rapid PVST+)

  IEEE 802.1w defined the Rapid Spanning Tree Protocol (RSTP) and was incorporated into IEEE 802.1D.
- IEEE 802.1s defined MST and was incorporated into IEEE 802.1Q.

Note You must enable MST; Rapid PVST+ is the default spanning tree mode.

MST Regions

To allow switches to participate in MST instances, you must consistently configure the switches with the same MST configuration information.

A collection of interconnected switches that have the same MST configuration is an MST region. An MST region is a linked group of MST bridges with the same MST configuration.

The MST configuration controls the MST region to which each switch belongs. The configuration includes the name of the region, the revision number, and the MST VLAN-to-instance assignment map.

A region can have one or multiple members with the same MST configuration. Each member must be capable of processing 802.1w bridge protocol data units (BPDUs). There is no limit to the number of MST regions in a network.

Each region can support up to 65 MST instances (MSTIs). Instances are identified by any number in the range from 1 to 4094. The system reserves Instance 0 for a special instance, which is the IST. You can assign a VLAN to only one MST instance at a time.

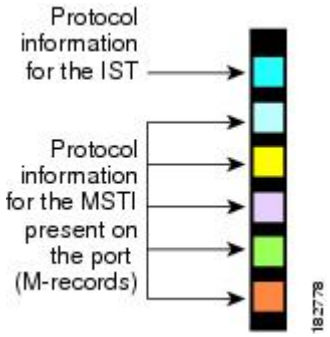
The MST region appears as a single bridge to adjacent MST regions and to other Rapid PVST+ regions and 802.1D spanning tree protocols.

Note We recommend that you do not partition the network into a large number of regions.

MST BPDUs

Each region has only one MST BPDU, and that BPDU carries an M-record for each MSTI within the region (see the following figure). Only the IST sends BPDUs for the MST region; all M-records are encapsulated in that one BPDU that the IST sends. Because the MST BPDU carries information for all instances, the number of BPDUs that need to be processed to support MSTIs is significantly reduced.

Figure 16: MST BPDU with M-Records for MSTIs



MST Configuration Information

The MST configuration that must be identical on all switches within a single MST region is configured by the user.

You can configure the following three parameters of the MST configuration:

- Name—32-character string, null padded and null terminated, identifying the MST region
- Revision number—Unsigned 16-bit number that identifies the revision of the current MST configuration

Note You must set the revision number when required as part of the MST configuration. The revision number is *not* incremented automatically each time that the MST configuration is committed.

- MST configuration table—4096-element table that associates each of the potential 4094 VLANs supported to a given instance with the first (0) and last element (4095) set to 0. The value of element number X represents the instance to which VLAN X is mapped.

Caution When you change the VLAN-to-MSTI mapping, the system restarts MST.

MST BPDUs contain these three configuration parameters. An MST bridge accepts an MST BPDU into its own region only if these three configuration parameters match exactly. If one configuration attribute differs, the MST bridge considers the BPDU to be from another MST region.

IST, CIST, and CST

IST, CIST, and CST Overview

Unlike Rapid PVST+, in which all the STP instances are independent, MST establishes and maintains IST, CIST, and CST spanning trees, as follows:

- An IST is the spanning tree that runs in an MST region.

MST establishes and maintains additional spanning trees within each MST region; these spanning trees are called, multiple spanning tree instances (MSTIs).

Instance 0 is a special instance for a region, known as the IST. The IST always exists on all ports; you cannot delete the IST, or Instance 0. By default, all VLANs are assigned to the IST. All other MST instances are numbered from 1 to 4094.

The IST is the only STP instance that sends and receives BPDUs. All of the other MSTI information is contained in MST records (M-records), which are encapsulated within MST BPDUs.

All MSTIs within the same region share the same protocol timers, but each MSTI has its own topology parameters, such as the root bridge ID, the root path cost, and so forth.

An MSTI is local to the region; for example, MSTI 9 in region A is independent of MSTI 9 in region B, even if regions A and B are interconnected.

- The CST interconnects the MST regions and any instance of 802.1D and 802.1w STP that may be running on the network. The CST is the one STP instance for the entire bridged network and encompasses all MST regions and 802.1w and 802.1D instances.
- A CIST is a collection of the ISTs in each MST region. The CIST is the same as an IST inside an MST region, and the same as a CST outside an MST region.

The spanning tree computed in an MST region appears as a subtree in the CST that encompasses the entire switched domain. The CIST is formed by the spanning tree algorithm running among switches that support the 802.1w, 802.1s, and 802.1D standards. The CIST inside an MST region is the same as the CST outside a region.

Spanning Tree Operation Within an MST Region

The IST connects all the MST switches in a region. When the IST converges, the root of the IST becomes the CIST regional root. The CIST regional root is also the CIST root if there is only one region in the network. If the CIST root is outside the region, the protocol selects one of the MST switches at the boundary of the region as the CIST regional root.

When an MST switch initializes, it sends BPDUs that identify itself as the root of the CIST and the CIST regional root, with both the path costs to the CIST root and to the CIST regional root set to zero. The switch also initializes all of its MSTIs and claims to be the root for all of them. If the switch receives superior MST root information (lower switch ID, lower path cost, and so forth) than the information that is currently stored for the port, it relinquishes its claim as the CIST regional root.

During initialization, an MST region might have many subregions, each with its own CIST regional root. As switches receive superior IST information from a neighbor in the same region, they leave their old subregions and join the new subregion that contains the true CIST regional root. This action causes all subregions to shrink except for the subregion that contains the true CIST regional root.

All switches in the MST region must agree on the same CIST regional root. Any two switches in the region will only synchronize their port roles for an MSTI if they converge to a common CIST regional root.

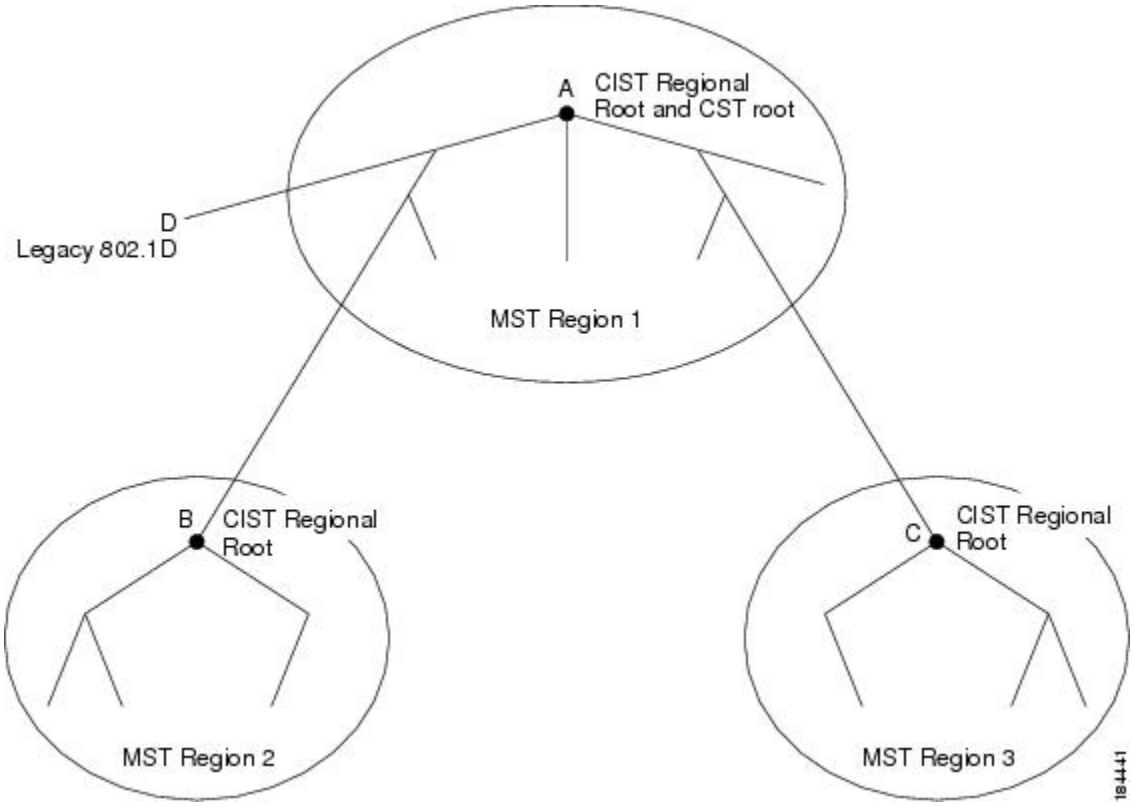
Spanning Tree Operations Between MST Regions

If you have multiple regions or 802.1 w or 802.1D STP instances within a network, MST establishes and maintains the CST, which includes all MST regions and all 802.1w and 802.1D STP switches in the network. The MSTIs combine with the IST at the boundary of the region to become the CST.

The IST connects all the MST switches in the region and appears as a subtree in the CIST that encompasses the entire switched domain. The root of the subtree is the CIST regional root. The MST region appears as a virtual switch to adjacent STP switches and MST regions.

The following figure shows a network with three MST regions and an 802.1D switch (D). The CIST regional root for region 1 (A) is also the CIST root. The CIST regional root for region 2 (B) and the CIST regional root for region 3 (C) are the roots for their respective subtrees within the CIST.

Figure 17: MST Regions, CIST Regional Roots, and CST Root



Only the CST instance sends and receives BPDUs. MSTIs add their spanning tree information into the BPDUs (as M-records) to interact with neighboring switches and compute the final spanning tree topology. Because of this, the spanning tree parameters related to the BPDU transmission (for example, hello time, forward time, max-age, and max-hops) are configured only on the CST instance but affect all MSTIs. You can configure the parameters related to the spanning tree topology (for example, the switch priority, the port VLAN cost, and the port VLAN priority) on both the CST instance and the MSTI.

MST switches use Version 3 BPDUs or 802.1D STP BPDUs to communicate with 802.1D-only switches. MST switches use MST BPDUs to communicate with MST switches.

MST Terminology

MST naming conventions include identification of some internal or regional parameters. These parameters are used only within an MST region, compared to external parameters that are used throughout the whole network. Because the CIST is the only spanning tree instance that spans the whole network, only the CIST parameters require the external qualifiers and not the internal or regional qualifiers. The MST terminology is as follows:

- The CIST root is the root bridge for the CIST, which is the unique instance that spans the whole network.

- The CIST external root path cost is the cost to the CIST root. This cost is left unchanged within an MST region. An MST region looks like a single switch to the CIST. The CIST external root path cost is the root path cost calculated between these virtual switches and switches that do not belong to any region.
- If the CIST root is in the region, the CIST regional root is the CIST root. Otherwise, the CIST regional root is the closest switch to the CIST root in the region. The CIST regional root acts as a root bridge for the IST.
- The CIST internal root path cost is the cost to the CIST regional root in a region. This cost is only relevant to the IST, instance 0.

Hop Count

MST does not use the message-age and maximum-age information in the configuration BPDU to compute the STP topology inside the MST region. Instead, the protocol uses the path cost to the root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism.

By using the **spanning-tree mst max-hops** global configuration command, you can configure the maximum hops inside the region and apply it to the IST and all MST instances in that region.

The hop count achieves the same result as the message-age information (triggers a reconfiguration). The root bridge of the instance always sends a BPDU (or M-record) with a cost of 0 and the hop count set to the maximum value. When a switch receives this BPDU, it decrements the received remaining hop count by one and propagates this value as the remaining hop count in the BPDUs that it generates. When the count reaches zero, the switch discards the BPDU and ages the information held for the port.

The message-age and maximum-age information in the 802.1w portion of the BPDU remain the same throughout the region (only on the IST), and the same values are propagated by the region-designated ports at the boundary.

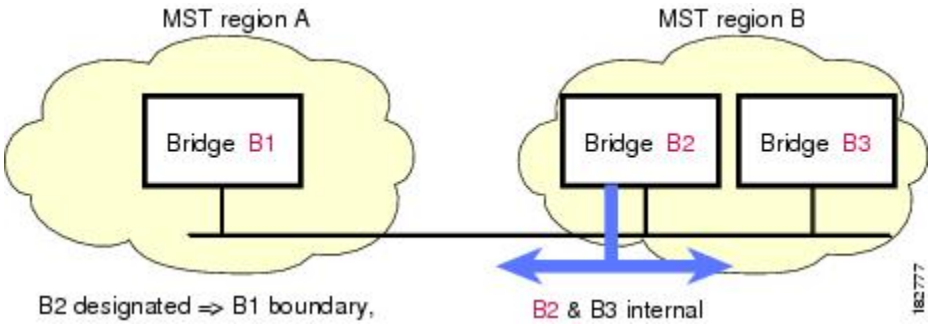
You configure a maximum aging time as the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration.

Boundary Ports

A boundary port is a port that connects one region to another. A designated port knows that it is on the boundary if it detects an STP bridge or receives an agreement proposal from an MST bridge with a different configuration or a Rapid PVST+ bridge. This definition allows two ports that are internal to a region to share a segment

with a port that belongs to a different region, creating the possibility of receiving both internal and external messages on a port (see the following figure).

Figure 18: MST Boundary Ports



At the boundary, the roles of MST ports do not matter; the system forces their state to be the same as the IST port state. If the boundary flag is set for the port, the MST port-role selection process assigns a port role to the boundary and assigns the same state as the state of the IST port. The IST port at the boundary can take up any port role except a backup port role.

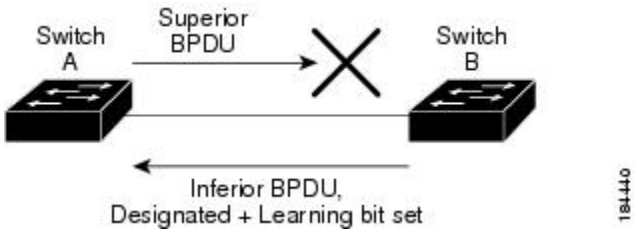
Spanning-Tree Dispute Mechanism

Currently, this feature is not present in the IEEE MST standard, but it is included in the standard-compliant implementation. The software checks the consistency of the port role and state in the received BPDUs to detect unidirectional link failures that could cause bridging loops.

When a designated port detects a conflict, it keeps its role, but reverts to a discarding state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

The following figure shows a unidirectional link failure that typically creates a bridging loop. Switch A is the root bridge, and its BPDUs are lost on the link leading to switch B. Rapid PVST+ (802.1w) and MST BPDUs include the role and state of the sending port. With this information, switch A can detect that switch B does not react to the superior BPDUs that it sends and that switch B is the designated, not root port. As a result, switch A blocks (or keeps blocking) its port, which prevents the bridging loop. The block is shown as an STP dispute.

Figure 19: Detecting a Unidirectional Link Failure



Port Cost and Port Priority

Spanning tree uses port costs to break a tie for the designated port. Lower values indicate lower port costs, and spanning tree chooses the least costly path. Default port costs are taken from the bandwidth of the interface, as follows:

- 10 Mbps—2,000,000
- 100 Mbps—200,000
- 1 Gigabit Ethernet—20,000
- 10 Gigabit Ethernet—2,000

You can configure the port costs in order to influence which port is chosen.

Note MST always uses the long path cost calculation method, so the range of valid values is between 1 and 200,000,000.

The system uses port priorities to break ties among ports with the same cost. A lower number indicates a higher priority. The default port priority is 128. You can configure the priority to values between 0 and 224, in increments of 32.

Interoperability with IEEE 802.1D

A switch that runs MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D STP switches. If this switch receives an 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. In addition, an MST switch can detect that a port is at the boundary of a region when it receives an 802.1D BPDU, an MST BPDU (Version 3) associated with a different region, or an 802.1w BPDU (Version 2).

However, the switch does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D switch has been removed from the link unless the 802.1D switch is the designated switch. A switch might also continue to assign a boundary role to a port when the switch to which this switch is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches), enter the **clear spanning-tree detected-protocols** command.

All Rapid PVST+ switches (and all 8021.D STP switches) on the link can process MST BPDUs as if they are 802.1w BPDUs. MST switches can send either Version 0 configuration and topology change notification (TCN) BPDUs or Version 3 MST BPDUs on a boundary port. A boundary port connects to a LAN, the designated switch of which is either a single spanning tree switch or a switch with a different MST configuration.

Note MST interoperates with the Cisco prestandard MSTP whenever it receives prestandard MSTP on an MST port; no explicit configuration is necessary.

Interoperability with Rapid PVST+: Understanding PVST Simulation

MST interoperates with Rapid PVST+ with no need for user configuration. The PVST simulation feature enables this seamless interoperability.

Note PVST simulation is enabled by default. That is, by default, all interfaces on the switch interoperate between MST and Rapid PVST+.

However, you may want to control the connection between MST and Rapid PVST+ to protect against accidentally connecting an MST-enabled port to a Rapid PVST+-enabled port. Because Rapid PVST+ is the default STP mode, you may encounter many Rapid PVST+-enabled connections.

Disabling Rapid PVST+ simulation, which can be done per port or globally for the entire switch, moves the MST-enabled port to the blocking state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving Rapid PVST+/SSTP BPDUs, and then the port resumes the normal STP transition process.

Configuring MST

MST Configuration Guidelines

When configuring MST, follow these guidelines:

- When you work with private VLANs, enter the **private-vlan synchronize** command to map the secondary VLANs to the same MST instance as the primary VLAN.
- When you are in the MST configuration mode, the following guidelines apply:
  - Each command reference line creates its pending regional configuration.
  - The pending region configuration starts with the current region configuration.
  - To leave the MST configuration mode without committing any changes, enter the **abort** command.
  - To leave the MST configuration mode and commit all the changes that you made before you left the mode, enter the **exit** command.

Enabling MST

You must enable MST; Rapid PVST+ is the default.

Caution Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mode mst**
3. (Optional) switch(config)# **no spanning-tree mode mst**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mode mst**	Enables MST on the switch.
Step 3	switch(config)# **no spanning-tree mode mst**	(Optional) Disables MST on the switch and returns you to Rapid PVST+.

This example shows how to enable MST on the switch:

```
switch# configure terminal
switch(config)# spanning-tree mode mst
```

Note Because STP is enabled by default, entering a **show running-config** command to view the resulting configuration does not display the command that you entered to enable STP.

Entering MST Configuration Mode

You enter MST configuration mode to configure the MST name, VLAN-to-instance mapping, and MST revision number on the switch.

For two or more switches to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

Note Each command reference line creates its pending regional configuration in MST configuration mode. In addition, the pending region configuration starts with the current region configuration.

When you are working in MST configuration mode, note the difference between the **exit** and **abort** commands.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **exit** or switch(config-mst)# **abort**
4. (Optional) switch(config)# **no spanning-tree mst configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration mode on the system. You must be in the MST configuration mode to assign the MST configuration parameters, as follows: • MST name • Instance-to-VLAN mapping • MST revision number • Synchronize primary and secondary VLANs in private VLANs
Step 3	switch(config-mst)# **exit** or switch(config-mst)# **abort**	• The first form commits all the changes and exits MST configuration mode. • The second form exits the MST configuration mode without committing any of the changes.
Step 4	switch(config)# **no spanning-tree mst configuration**	(Optional) Returns the MST region configuration to the following default values: • The region name is an empty string. • No VLANs are mapped to any MST instance (all VLANs are mapped to the CIST instance). • The revision number is 0.

Specifying the MST Name

You configure a region name on the bridge. For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **name** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration submode.
Step 3	switch(config-mst)# **name** *name*	Specifies the name for MST region. The *name* string has a maximum length of 32 characters and is case-sensitive. The default is an empty string.

This example shows how to set the name of the MST region:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# name accounting
```

Specifying the MST Configuration Revision Number

You configure the revision number on the bridge. For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **revision** *version*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration submode.
Step 3	switch(config-mst)# **revision** *version*	Specifies the revision number for the MST region. The range is from 0 to 65535, and the default value is 0.

This example shows how to configure the revision number of the MSTI region for 5:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# revision 5
```

Specifying the Configuration on an MST Region

For two or more switches to be in the same MST region, they must have the same VLAN-to-instance mapping, the same configuration revision number, and the same MST name.

A region can have one member or multiple members with the same MST configuration; each member must be capable of processing IEEE 802.1w RSTP BPDUs. There is no limit to the number of MST regions in a network, but each region can support only up to 65 MST instances. You can assign a VLAN to only one MST instance at a time.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **instance** *instance-id* **vlan** *vlan-range*
4. switch(config-mst)# **name** *name*
5. switch(config-mst)# **revision** *version*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration submode.
Step 3	switch(config-mst)# **instance** *instance-id* **vlan** *vlan-range*	Maps VLANs to an MST instance as follows: • For *instance-id* , the range is from 1 to 4094. • For **vlan** *vlan-range* , the range is from 1 to 4094. When you map VLANs to an MST instance, the mapping is incremental, and the VLANs specified in the command are added to or removed from the VLANs that were previously mapped. To specify a VLAN range, enter a hyphen; for example, enter the **instance 1 vlan 1-63** command to map VLANs 1 through 63 to MST instance 1. To specify a VLAN series, enter a comma; for example, enter the **instance 1 vlan 10, 20, 30** command to map VLANs 10, 20, and 30 to MST instance 1.

	Command or Action	Purpose
Step 4	switch(config-mst)# **name** *name*	Specifies the instance name. The *name* string has a maximum length of 32 characters and is case sensitive.
Step 5	switch(config-mst)# **revision** *version*	Specifies the configuration revision number. The range is from 0 to 65535.

To return to defaults, do the following:

- To return to the default MST region configuration settings, enter the **no spanning-tree mst configuration** configuration command.
- To return to the default VLAN-to-instance map, enter the **no instance** *instance-id* **vlan** *vlan-range* MST configuration command.
- To return to the default name, enter the **no name** MST configuration command.
- To return to the default revision number, enter the **no revision** MST configuration command.
- To reenable Rapid PVST+, enter the **no spanning-tree mode** or the **spanning-tree mode rapid-pvst** global configuration command.

This example shows how to enter MST configuration mode, map VLANs 10 to 20 to MST instance 1, name the region region1, set the configuration revision to 1, display the pending configuration, apply the changes, and return to global configuration mode:

```
switch(config)# spanning-tree mst configuration
switch(config-mst)# instance 1 vlan 10-20
switch(config-mst)# name region1
switch(config-mst)# revision 1
switch(config-mst)# show pending
Pending MST configuration
Name      [region1]
Revision  1
Instances configured 2
Instance  Vlans Mapped
--------  --------------------
0         1-9,21-4094
1         10-20
-------------------------------
```

Mapping and Unmapping VLANs to MST Instances

Caution When you change the VLAN-to-MSTI mapping, the system restarts MST.

Note You cannot disable an MSTI.

For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **instance** *instance-id* **vlan** *vlan-range*
4. switch(config-mst)# **no instance** *instance-id* **vlan** *vlan-range*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration submode.
Step 3	switch(config-mst)# **instance** *instance-id* **vlan** *vlan-range*	Maps VLANs to an MST instance, as follows: • For *instance-id* the range is from 1 to 4094. Instance 0 is reserved for the IST for each MST region. • For *vlan-range* the range is from 1 to 4094. When you map VLANs to an MSTI, the mapping is incremental, and the VLANs specified in the command are added to or removed from the VLANs that were previously mapped.
Step 4	switch(config-mst)# **no instance** *instance-id* **vlan** *vlan-range*	Deletes the specified instance and returns the VLANs to the default MSTI, which is the CIST.

This example shows how to map VLAN 200 to MSTI 3:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# instance 3 vlan 200
```

Mapping Secondary VLANs to Same MSTI as Primary VLANs for Private VLANs

When you are working with private VLANs on the system, all secondary VLANs must be in the same MSTI and their associated primary VLAN.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **private-vlan synchronize**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst configuration**	Enters MST configuration submode.
Step 3	switch(config-mst)# **private-vlan synchronize**	Automatically maps all secondary VLANs to the same MSTI and their associated primary VLAN for all private VLANs.

This example shows how to automatically map all the secondary VLANs to the same MSTI as their associated primary VLANs in all private VLANs:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# private-vlan synchronize
```

Configuring the Root Bridge

You can configure the switch to become the root bridge.

Note The root bridge for each MSTI should be a backbone or distribution switch. Do not configure an access switch as the spanning tree primary root bridge.

Enter the diameter keyword, which is available only for MSTI 0 (or the IST), to specify the network diameter (that is, the maximum number of hops between any two end stations in the network). When you specify the network diameter, the switch automatically sets an optimal hello time, forward-delay time, and maximum-age time for a network of that diameter, which can significantly reduce the convergence time. You can enter the hello keyword to override the automatically calculated hello time.

Note With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the **spanning-tree mst hello-time**, **spanning-tree mst forward-time**, and **spanning-tree mst max-age** global configuration commands.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]]
3. (Optional) switch(config)# **no spanning-tree mst** *instance-id* **root**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]]	Configures a switch as the root bridge as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094. • For diameter *net-diameter*, specify the maximum number of hops between any two end stations. The default is 7. This keyword is available only for MST instance 0. • For hello-time *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds.
Step 3	switch(config)# **no spanning-tree mst** *instance-id* **root**	(Optional) Returns the switch priority, diameter, and hello time to default values.

This example shows how to configure the switch as the root switch for MSTI 5:

```
switch# configure terminal
switch(config)# spanning-tree mst 5 root primary
```

Configuring a Secondary Root Bridge

You can execute this command on more than one switch to configure multiple backup root bridges. Enter the same network diameter and hello-time values that you used when you configured the primary root bridge with the **spanning-tree mst root primary** configuration command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]]
3. (Optional) switch(config)# **no spanning-tree mst** *instance-id* **root**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	switch(config)# **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]]	Configures a switch as the secondary root bridge as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094. • For diameter *net-diameter*, specify the maximum number of hops between any two end stations. The default is 7. This keyword is available only for MST instance 0. • For hello-time *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds.
Step 3	switch(config)# **no spanning-tree mst** *instance-id* **root**	(Optional) Returns the switch priority, diameter, and hello-time to default values.

This example shows how to configure the switch as the secondary root switch for MSTI 5:

```
switch# configure terminal
switch(config)# spanning-tree mst 5 root secondary
```

Configuring the Port Priority

If a loop occurs, MST uses the port priority when selecting an interface to put into the forwarding state. You can assign lower priority values to interfaces that you want selected first and higher priority values to the interface that you want selected last. If all interfaces have the same priority value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}
3. switch(config-if)# **spanning-tree mst** *instance-id* **port-priority** *priority*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **spanning-tree mst** *instance-id* **port-priority** *priority*	Configures the port priority as follows:

	Command or Action	Purpose
		• For *instance-id*, you can specify a single MSTI, a range of MSTIs separated by a hyphen, or a series of MSTIs separated by a comma. The range is from 1 to 4094. • For *priority*, the range is 0 to 224 in increments of 32. The default is 128. A lower number indicates a higher priority. The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. The system rejects all other values.

This example shows how to set the MST interface port priority for MSTI 3 on Ethernet port 3/1 to 64:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# spanning-tree mst 3 port-priority 64
```

You can only apply this command to a physical Ethernet interface.

Configuring the Port Cost

The MST path cost default value is derived from the media speed of an interface. If a loop occurs, MST uses the cost when selecting an interface to put in the forwarding state. You can assign lower cost values to interfaces that you want selected first and higher cost to interfaces values that you want selected last. If all interfaces have the same cost value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Note MST uses the long pathcost calculation method.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}
3. switch(config-if)# **spanning-tree mst** *instance-id* **cost** [*cost* | **auto**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **spanning-tree mst** *instance-id* **cost** [*cost* | **auto**]	Configures the cost.

	Command or Action	Purpose
		If a loop occurs, MST uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094. • For *cost*, the range is from 1 to 200000000. The default value is auto, which is derived from the media speed of the interface.

This example shows how to set the MST interface port cost on Ethernet 3/1 for MSTI 4:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# spanning-tree mst 4 cost 17031970
```

Configuring the Switch Priority

You can configure the switch priority for an MST instance so that it is more likely that the specified switch is chosen as the root bridge.

Note Exercise care when using this command. For most situations, we recommend that you enter the **spanning-tree mst root primary** and the **spanning-tree mst root secondary** global configuration commands to modify the switch priority.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst** *instance-id* **priority** *priority-value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst** *instance-id* **priority** *priority-value*	Configures a switch priority as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094.

	Command or Action	Purpose
		• For priority, the range is from 0 to 61440 in increments of 4096; the default is 32768. A lower number indicates that the switch will most likely be chosen as the root bridge. Priority values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. The system rejects all other values.

This example shows how to configure the priority of the bridge to 4096 for MSTI 5:

```
switch# configure terminal
switch(config)# spanning-tree mst 5 priority 4096
```

Configuring the Hello Time

You can configure the interval between the generation of configuration messages by the root bridge for all instances on the switch by changing the hello time.

Note Exercise care when using this command. For most situations, we recommend that you enter the **spanning-tree mst** *instance-id* **root primary** and the **spanning-tree mst** *instance-id* **root secondary** configuration commands to modify the hello time.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst hello-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst hello-time** *seconds*	Configures the hello time for all MST instances. The hello time is the interval between the generation of configuration messages by the root bridge. These messages mean that the switch is alive. For *seconds*, the range is from 1 to 10, and the default is 2 seconds.

This example shows how to configure the hello time of the switch to 1 second:

```
switch# configure terminal
switch(config)# spanning-tree mst hello-time 1
```

Configuring the Forwarding-Delay Time

You can set the forward delay timer for all MST instances on the switch with one command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst forward-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst forward-time** *seconds*	Configures the forward time for all MST instances. The forward delay is the number of seconds that a port waits before changing from its spanning tree blocking and learning states to the forwarding state. For *seconds*, the range is from 4 to 30, and the default is 15 seconds.

This example shows how to configure the forward-delay time of the switch to 10 seconds:

```
switch# configure terminal
switch(config)# spanning-tree mst forward-time 10
```

Configuring the Maximum-Aging Time

The maximum-aging timer is the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration.

You set the maximum-aging timer for all MST instances on the switch with one command (the maximum age time only applies to the IST).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst max-age** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst max-age** *seconds*	Configures the maximum-aging time for all MST instances. The maximum-aging time is the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration. For *seconds*, the range is from 6 to 40, and the default is 20 seconds.

This example shows how to configure the maximum-aging timer of the switch to 40 seconds:

```
switch# configure terminal
switch(config)# spanning-tree mst max-age 40
```

Configuring the Maximum-Hop Count

MST uses the path cost to the IST regional root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism. You configure the maximum hops inside the region and apply it to the IST and all MST instances in that region. The hop count achieves the same result as the message-age information (triggers a reconfiguration).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst max-hops** *hop-count*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree mst max-hops** *hop-count*	Specifies the number of hops in a region before the BPDU is discarded, and the information held for a port is aged. For *hop-count*, the range is from 1 to 255, and the default value is 20 hops.

This example shows how to set the maximum hops to 40:

```
switch# configure terminal
switch(config)# spanning-tree mst max-hops 40
```

Configuring PVST Simulation Globally

You can block this automatic feature either globally or per port. You can enter the global command, and change the PVST simulation setting for the entire switch while you are in interface command mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no spanning-tree mst simulate pvst global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **no spanning-tree mst simulate pvst global**	Disables all interfaces on the switch from automatically interoperating with connected switch that is running in Rapid PVST+ mode. The default for this is enabled; that is, by default, all interfaces on the switch operate seamlessly between Rapid PVST+ and MST.

This example shows how to prevent the switch from automatically interoperating with a connecting switch that is running Rapid PVST+:

```
switch# configure terminal
switch(config)# no spanning-tree mst simulate pvst global
```

Configuring PVST Simulation Per Port

MST interoperates seamlessly with Rapid PVST+. However, to prevent an accidental connection to a switch that does not run MST as the default STP mode, you may want to disable this automatic feature. If you disable PVST simulation, the MST-enabled port moves to the blocking state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving BPDUs, and then the port resumes the normal STP transition process.

You can block this automatic feature either globally or per port.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}
3. switch(config-if)# **spanning-tree mst simulate pvst disable**
4. switch(config-if)# **spanning-tree mst simulate pvst**
5. switch(config-if)# **no spanning-tree mst simulate pvst**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** {{*type slot*/*port*} | {**port-channel** *number*}}	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **spanning-tree mst simulate pvst disable**	Disables specified interfaces from automatically interoperating with connected switch that is running in Rapid PVST+ mode. By default, all interfaces on the switch operate seamlessly between Rapid PVST+ and MST.
Step 4	switch(config-if)# **spanning-tree mst simulate pvst**	Re-enables seamless operation between MST and Rapid PVST+ on specified interfaces.
Step 5	switch(config-if)# **no spanning-tree mst simulate pvst**	Sets the interface to the switch-wide MST and Rapid PVST+ interoperation that you configured using the **spanning-tree mst simulate pvst global** command.

This example shows how to prevent the specified interfaces from automatically interoperating with a connecting switch that is not running MST:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree mst simulate pvst disable
```

Specifying the Link Type

Rapid connectivity (802.1w standard) is established only on point-to-point links. By default, the link type is controlled from the duplex mode of the interface. A full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection.

If you have a half-duplex link physically connected point-to-point to a single port on a remote switch, you can override the default setting on the link type and enable rapid transitions.

If you set the link to shared, STP reverts to 802.1D.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **spanning-tree link-type** {**auto** | **point-to-point** | **shared**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Specifies the interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **spanning-tree link-type** {**auto** | **point-to-point** | **shared**}	Configures the link type to be either point to point or shared. The system reads the default value from the switch connection. Half-duplex links are shared and full-duplex links are point to point. If the link type is shared, the STP reverts to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface.

This example shows how to configure the link type as point to point:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
```

Restarting the Protocol

An MST bridge can detect that a port is at the boundary of a region when it receives a legacy BPDU or an MST BPDU that is associated with a different region. However, the STP protocol migration cannot determine whether the legacy switch, which is a switch that runs only IEEE 802.1D, has been removed from the link unless the legacy switch is the designated switch. Enter this command to restart the protocol negotiation (force the renegotiation with neighboring switches) on the entire switch or on specified interfaces.

SUMMARY STEPS

1. switch# **clear spanning-tree detected-protocol** [**interface** *interface* [*interface-num* | *port-channel*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **clear spanning-tree detected-protocol** [**interface** *interface* [*interface-num* | *port-channel*]]	Restarts MST on entire switch or specified interfaces.

This example shows how to restart MST on the Ethernet interface on slot 2, port 8:

```
switch# clear spanning-tree detected-protocol interface ethernet 2/8
```

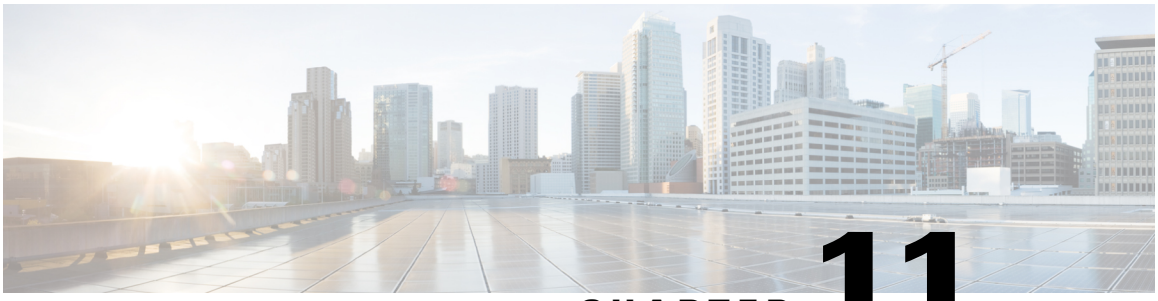
Verifying MST Configurations

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

Command	Purpose
switch# **show running-config spanning-tree** [**all**]	Displays the current spanning tree configuration.
switch# **show spanning-tree mst** [*options*]	Displays detailed information for the current MST configuration.

The following example shows how to display current MST configuration:

```
switch# show spanning-tree mst configuration
% Switch is not in mst mode
Name      [mist-attempt]
Revision  1     Instances configured 2
Instance  Vlans mapped
--------  ---------------------------------------------------------------------
0         1-12,14-41,43-4094
1         13,42
```



CHAPTER 11

Configuring STP Extensions

This chapter contains the following sections:

- About STP Extensions, page 159

About STP Extensions

Cisco has added extensions to STP that make convergence more efficient. In some cases, even though similar functionality may be incorporated into the IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) standard, we recommend using these extensions. All of these extensions can be used with both RPVST+ and MST.

The available extensions are spanning tree port types, Bridge Assurance, BPDU Guard, BPDU Filtering, Loop Guard, and Root Guard. Many of these features can be applied either globally or on specified interfaces.

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

Information About STP Extensions

Understanding STP Port Types

You can configure a spanning tree port as an edge port, a network port, or a normal port. A port can be in only one of these states at a given time. The default spanning tree port type is normal. Depending on the type of device to which the interface is connected, you can configure a spanning tree port as one of these port types.

Spanning Tree Edge Ports

Edge ports, which are connected to hosts, can be either an access port or a trunk port. The edge port interface immediately transitions to the forwarding state, without moving through the blocking or learning states. (This immediate transition was previously configured as the Cisco-proprietary feature PortFast.)

Interfaces that are connected to hosts should not receive STP Bridge Protocol Data Units (BPDUs).

Note If you configure a port connected to another switch as an edge port, you might create a bridging loop.

Spanning Tree Network Ports

Network ports are connected only to switches or bridges. Configuring a port as "network" while Bridge Assurance is enabled globally, enables Bridge Assurance on that port.

Note If you mistakenly configure ports that are connected to hosts or other edge devices, as spanning tree network ports, those ports will automatically move into the blocking state.

Spanning Tree Normal Ports

Normal ports can be connected to either hosts, switches, or bridges. These ports function as normal spanning tree ports.

The default spanning tree interface is a normal port.

Understanding Bridge Assurance

You can use Bridge Assurance to protect against certain problems that can cause bridging loops in the network. Specifically, you use Bridge Assurance to protect against a unidirectional link failure and a device that continues to forward data traffic when it is no longer running the spanning tree algorithm.

Note Bridge Assurance is supported only by Rapid PVST+ and MST. Legacy 802.1D spanning tree does not support Bridge Assurance.

Bridge Assurance is enabled by default and can only be disabled globally. Also, Bridge Assurance can be enabled only on spanning tree network ports that are point-to-point links. Finally, both ends of the link must have Bridge Assurance enabled.

With Bridge Assurance enabled, BPDUs are sent out on all operational network ports, including alternate and backup ports, for each hello time period. If the port does not receive a BPDU for a specified period, the port moves into the blocking state and is not used in the root port calculation. Once that port receives a BPDU, it resumes the normal spanning tree transitions.

Understanding BPDU Guard

Enabling BPDU Guard shuts down that interface if a BPDU is received.

You can configure BPDU Guard at the interface level. When configured at the interface level, BPDU Guard shuts the port down as soon as the port receives a BPDU, regardless of the port type configuration.

When you configure BPDU Guard globally, it is effective only on operational spanning tree edge ports. In a valid configuration, LAN edge interfaces do not receive BPDUs. A BPDU that is received by an edge LAN

interface signals an invalid configuration, such as the connection of an unauthorized host or switch. BPDU Guard, when enabled globally, shuts down all spanning tree edge ports when they receive a BPDU.

BPDU Guard provides a secure response to invalid configurations, because you must manually put the LAN interface back in service after an invalid configuration.

Note When enabled globally, BPDU Guard applies to all operational spanning tree edge interfaces.

Understanding BPDU Filtering

You can use BPDU Filtering to prevent the switch from sending or even receiving BPDUs on specified ports.

When configured globally, BPDU Filtering applies to all operational spanning tree edge ports. You should connect edge ports only to hosts, which typically drop BPDUs. If an operational spanning tree edge port receives a BPDU, it immediately returns to a normal spanning tree port type and moves through the regular transitions. In that case, BPDU Filtering is disabled on this port, and spanning tree resumes sending BPDUs on this port.

In addition, you can configure BPDU Filtering by the individual interface. When you explicitly configure BPDU Filtering on a port, that port does not send any BPDUs and drops all BPDUs that it receives. You can effectively override the global BPDU Filtering setting on individual ports by configuring the specific interface. This BPDU Filtering command on the interface applies to the entire interface, whether the interface is trunking or not.

Caution Use care when configuring BPDU Filtering per interface. If you explicitly configuring BPDU Filtering on a port that is not connected to a host, it can result in bridging loops because the port will ignore any BPDU that it receives and go to forwarding.

If the port configuration is not set to default BPDU Filtering, then the edge configuration will not affect BPDU Filtering. The following table lists all the BPDU Filtering combinations.

Table 13: BPDU Filtering Configurations

BPDU Filtering Per Port Configuration	BPDU Filtering Global Configuration	STP Edge Port Configuration	BPDU Filtering State
Default	Enable	Enable	EnableThe port transmits at least 10 BPDUs. If this port receives any BPDUs, the port returns to the spanning tree normal port state and BPDU Filtering is disabled.
Default	Enable	Disable	Disable
Default	Disable	Enabled/Disabled	Disable
Disable	Enabled/Disabled	Enabled/Disabled	Disable

BPDU Filtering Per Port Configuration	BPDU Filtering Global Configuration	STP Edge Port Configuration	BPDU Filtering State
Enable	Enabled/Disabled	Enabled/Disabled	Enable **Caution** BPDUs are never sent and if received, they do not trigger the regular STP behavior - use with caution.

Understanding Loop Guard

Loop Guard protects networks from loops that are caused by the following:

- Network interfaces that malfunction
- Busy CPUs
- Anything that prevents the normal forwarding of BPDUs

An STP loop occurs when a blocking port in a redundant topology erroneously transitions to the forwarding state. This transition usually happens because one of the ports in a physically redundant topology (not necessarily the blocking port) stops receiving BPDUs.

Loop Guard is only useful in switched networks where devices are connected by point-to-point links. On a point-to-point link, a designated bridge cannot disappear unless it sends an inferior BPDU or brings the link down.

Note Loop Guard can be enabled only on network and normal spanning tree port types.

You can use Loop Guard to determine if a root port or an alternate/backup root port receives BPDUs. If the port does not receive BPDUs, Loop Guard puts the port into an inconsistent state (blocking) until the port starts to receive BPDUs again. A port in the inconsistent state does not transmit BPDUs. If the port receives BPDUs again, the protocol removes its loop-inconsistent condition, and the STP determines the port state because such recovery is automatic.

Loop Guard isolates the failure and allows STP to converge to a stable topology without the failed link or bridge. Disabling Loop Guard moves all loop-inconsistent ports to the listening state.

You can enable Loop Guard on a per-port basis. When you enable Loop Guard on a port, it is automatically applied to all of the active instances or VLANs to which that port belongs. When you disable Loop Guard, it is disabled for the specified ports.

Understanding Root Guard

When you enable Root Guard on a port, Root Guard does not allow that port to become a root port. If a received BPDU triggers an STP convergence that makes that designated port become a root port, that port is

put into a root-inconsistent (blocked) state. After the port stops send superior BPDUs, the port is unblocked again. Through STP, the port moves to the forwarding state. Recovery is automatic.

Root Guard enabled on an interface applies this functionality to all VLANs to which that interface belongs.

You can use Root Guard to enforce the root bridge placement in the network. Root Guard ensures that the port on which Root Guard is enabled is the designated port. Normally, root bridge ports are all designated ports, unless two or more of the ports of the root bridge are connected. If the bridge receives superior BPDUs on a Root Guard-enabled port, the bridge moves this port to a root-inconsistent STP state. In this way, Root Guard enforces the position of the root bridge.

You cannot configure Root Guard globally.

Note You can enable Root Guard on all spanning tree port types: normal, edge, and network ports.

Configuring STP Extensions

STP Extensions Configuration Guidelines

When configuring STP extensions, follow these guidelines:

- Configure all access and trunk ports connected to hosts as edge ports.
- Bridge Assurance runs only on point-to-point spanning tree network ports. You must configure each side of the link for this feature.
- Loop Guard does not run on spanning tree edge ports.
- Enabling Loop Guard on ports that are not connected to a point-to-point link will not work.
- You cannot enable Loop Guard if Root Guard is enabled.

Configuring Spanning Tree Port Types Globally

The spanning tree port type designation depends on the type of device the port is connected to, as follows:

- Edge—Edge ports are connected to hosts and can be either an access port or a trunk port.
- Network—Network ports are connected only to switches or bridges.
- Normal—Normal ports are neither edge ports nor network ports; they are normal spanning tree ports. These ports can be connected to any type of device.

You can configure the port type either globally or per interface. By default, the spanning tree port type is normal.

Before You Begin

Ensure that STP is configured.

Ensure that you are configuring the ports correctly for the type of device to which the interface is connected.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree port type edge default**
3. switch(config)# **spanning-tree port type network default**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree port type edge default**	Configures all interfaces as edge ports. This assumes all ports are connected to hosts/servers. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types.
Step 3	switch(config)# **spanning-tree port type network default**	Configures all interfaces as spanning tree network ports. This assumes all ports are connected to switches and bridges. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types. **Note** If you configure interfaces connected to hosts as network ports, those ports automatically move into the blocking state.

This example shows how to configure all access and trunk ports connected to hosts as spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge default
```

This example shows how to configure all ports connected to switches or bridges as spanning tree network ports:

```
switch# configure terminal
switch(config)# spanning-tree port type network default
```

Configuring Spanning Tree Edge Ports on Specified Interfaces

You can configure spanning tree edge ports on specified interfaces. Interfaces configured as spanning tree edge ports immediately transition to the forwarding state, without passing through the blocking or learning states, on linkup.

This command has four states:

- **spanning-tree port type edge—**This command explicitly enables edge behavior on the access port.
- **spanning-tree port type edge trunk—**This command explicitly enables edge behavior on the trunk port.

Note If you enter the **spanning-tree port type edge trunk** command, the port is configured as an edge port even in the access mode.

- **spanning-tree port type normal—**This command explicitly configures the port as a normal spanning tree port and the immediate transition to the forwarding state is not enabled.
- **no spanning-tree port type—**This command implicitly enables edge behavior if you define the **spanning-tree port type edge default** command in global configuration mode. If you do not configure the edge ports globally, the **no spanning-tree port type** command is equivalent to the **spanning-tree port type disable** command.

Before You Begin

Ensure that STP is configured.

Ensure that the interface is connected to hosts.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **spanning-tree port type edge**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **spanning-tree port type edge**	Configures the specified access interfaces to be spanning edge ports. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types.

This example shows how to configure the Ethernet access interface 1/4 to be a spanning tree edge port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type edge
```

Configuring Spanning Tree Network Ports on Specified Interfaces

You can configure spanning tree network ports on specified interfaces.

Bridge Assurance runs only on spanning tree network ports.

This command has three states:

- **spanning-tree port type network—**This command explicitly configures the port as a network port. If you enable Bridge Assurance globally, it automatically runs on a spanning tree network port.
- **spanning-tree port type normal—**This command explicitly configures the port as a normal spanning tree port and Bridge Assurance cannot run on this interface.
- **no spanning-tree port type—**This command implicitly enables the port as a spanning tree network port if you define the **spanning-tree port type network default** command in global configuration mode. If you enable Bridge Assurance globally, it automatically runs on this port.

Note A port connected to a host that is configured as a network port automatically moves into the blocking state.

Before You Begin

Ensure that STP is configured.

Ensure that the interface is connected to switches or routers.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** type slot/port
3. switch(config-if)# **spanning-tree port type network**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** type slot/port	Specifies the interface to configure, and enters the interface configuration mode. The interface can be a physical Ethernet port.
Step 3	switch(config-if)# **spanning-tree port type network**	Configures the specified interfaces to be spanning network ports. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types.

This example shows how to configure the Ethernet interface 1/4 to be a spanning tree network port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type network
```

Enabling BPDU Guard Globally

You can enable BPDU Guard globally by default. In this condition, the system shuts down an edge port that receives a BPDU.

Note We recommend that you enable BPDU Guard on all edge ports.

Before You Begin

Ensure that STP is configured.

Ensure that you have configured some spanning tree edge ports.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree port type edge bpduguard default**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree port type edge bpduguard default**	Enables BPDU Guard by default on all spanning tree edge ports. By default, global BPDU Guard is disabled.

This example shows how to enable BPDU Guard on all spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge bpduguard default
```

Enabling BPDU Guard on Specified Interfaces

You can enable BPDU Guard on specified interfaces. Enabling BPDU Guard shuts down the port if it receives a BPDU.

You can configure BPDU Guard on specified interfaces as follows:

- **spanning-tree bpduguard enable—**Unconditionally enables BPDU Guard on the interface.
- **spanning-tree bpduguard disable—**Unconditionally disables BPDU Guard on the interface.
- **no spanning-tree bpduguard—**Enables BPDU Guard on the interface if it is an operational edge port and if the **spanning-tree port type edge bpduguard default** command is configured.

Before You Begin

Ensure that STP is configured.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **spanning-tree bpduguard** {**enable** | **disable**}
4. (Optional) switch(config-if)# **no spanning-tree bpduguard**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **spanning-tree bpduguard** {**enable** | **disable**}	Enables or disables BPDU Guard for the specified spanning tree edge interface. By default, BPDU Guard is disabled on physical Ethernet interfaces.
Step 4	switch(config-if)# **no spanning-tree bpduguard**	(Optional) Disables BPDU Guard on the interface. **Note** Enables BPDU Guard on the interface if it is an operational edge port and if you enter the **spanning-tree port type edge bpduguard default** command.

This example shows how to explicitly enable BPDU Guard on the Ethernet edge port 1/4:

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

switch(config-if)# no spanning-tree bpduguard

Enabling BPDU Filtering Globally

You can enable BPDU Filtering globally by default on spanning tree edge ports.

If an edge port with BPDU Filtering enabled receives a BPDU, it loses its operation status and as edge port and resumes the regular STP transitions. However, this port maintains it configuration as an edge port.

Caution Be careful when using this command: using it incorrectly can cause bridging loops.

Note When enabled globally, BPDU Filtering is applied *only* on ports that are operational edge ports. Ports send a few BPDUs at linkup before they effectively filter outbound BPDUs. If a BPDU is received on an edge port, it immediately loses its operational edge port status and BPDU Filtering is disabled.

Before You Begin

Ensure that STP is configured.

Ensure that you have configured some spanning tree edge ports.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree port type edge bpdufilter default**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree port type edge bpdufilter default**	Enables BPDU Filtering by default on all operational spanning tree edge ports. Global BPDU Filtering is disabled by default.

This example shows how to enable BPDU Filtering on all operational spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge bpdufilter default
```

Enabling BPDU Filtering on Specified Interfaces

You can apply BPDU Filtering to specified interfaces. When enabled on an interface, that interface does not send any BPDUs and drops all BPDUs that it receives. This BPDU Filtering functionality applies to the entire interface, whether trunking or not.

Caution Be careful when you enter the **spanning-tree bpdufilter enable** command on specified interfaces. Explicitly configuring BPDU Filtering on a port that is not connected to a host can result in bridging loops as the port will ignore any BPDU it receives and go to forwarding.

You can enter this command to override the port configuration on specified interfaces.

This command has three states:

- **spanning-tree bpdufilter enable—**Unconditionally enables BPDU Filtering on the interface.
- **spanning-tree bpdufilter disable—**Unconditionally disables BPDU Filtering on the interface.
- **no spanning-tree bpdufilter—**Enables BPDU Filtering on the interface if the interface is in operational edge port and if you configure the **spanning-tree port type edge bpdufilter default** command.

Note When you enable BPDU Filtering locally on a port, this feature prevents the device from receiving or sending BPDUs on this port.

Before You Begin

Ensure that STP is configured.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **spanning-tree bpdufilter** {**enable** | **disable**}
4. (Optional) switch(config-if)# **no spanning-tree bpdufilter**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **spanning-tree bpdufilter** {**enable** | **disable**}	Enables or disables BPDU Filtering for the specified spanning tree edge interface. By default, BPDU Filtering is disabled.
Step 4	switch(config-if)# **no spanning-tree bpdufilter**	(Optional) Disables BPDU Filtering on the interface. **Note** Enables BPDU Filtering on the interface if the interface is an operational spanning tree edge port and if you enter the spanning-tree port type edge bpdufilter default command.

This example shows how to explicitly enable BPDU Filtering on the Ethernet spanning tree edge port 1/4:

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

Enabling Loop Guard Globally

You can enable Loop Guard globally by default on all point-to-point spanning tree normal and network ports. Loop Guard does not run on edge ports.

Loop Guard provides additional security in the bridge network. Loop Guard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link.

Note Entering the Loop Guard command for the specified interface overrides the global Loop Guard command.

Before You Begin

Ensure that STP is configured.

Ensure that you have spanning tree normal ports or have configured some network ports.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree loopguard default**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **spanning-tree loopguard default**	Enables Loop Guard by default on all spanning tree normal and network ports. By default, global Loop Guard is disabled.

This example shows how to enable Loop Guard on all spanning tree normal or network ports:

```
switch# configure terminal
switch(config)# spanning-tree loopguard default
```

Enabling Loop Guard or Root Guard on Specified Interfaces

You can enable either Loop Guard or Root Guard on specified interfaces.

Enabling Root Guard on a port means that port cannot become a root port, and LoopGuard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link.

Both Loop Guard and Root Guard enabled on an interface apply to all VLANs to which that interface belongs.

Note Entering the Loop Guard command for the specified interface overrides the global Loop Guard command.

Before You Begin

Ensure that STP is configured.

Ensure that you are configuring Loop Guard on spanning tree normal or network ports.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **spanning-tree guard** {**loop** | **root** | **none**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **spanning-tree guard** {**loop** | **root** | **none**}	Enables or disables either Loop Guard or Root Guard for the specified interface. By default, Root Guard is disabled by default, and Loop Guard on specified ports is also disabled. **Note** Loop Guard runs only on spanning tree normal and network interfaces.

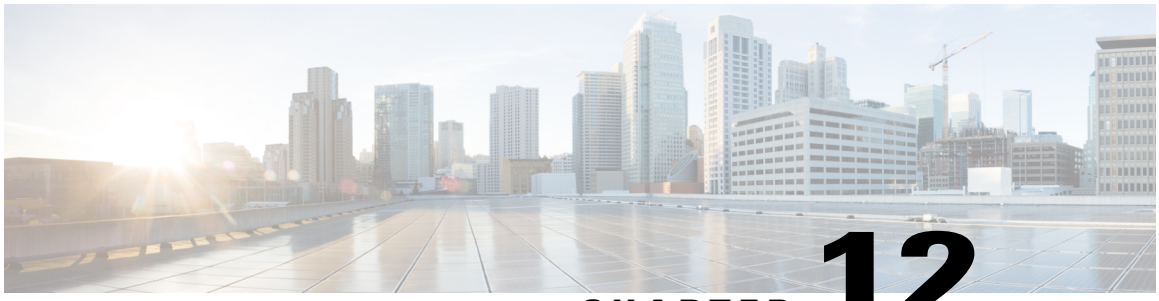
This example shows how to enable Root Guard on Ethernet port 1/4:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree guard root
```

Verifying STP Extension Configuration

To display the configuration information for the STP extensions, perform one of the following tasks:

Command	Purpose
switch# **show running-config spanning-tree** [**all**]	Displays the current status of spanning tree on the switch
switch# **show spanning-tree** [*options*]	Displays selected detailed information for the current spanning tree configuration.



CHAPTER 12

Configuring LLDP

This chapter contains the following sections:

- Configuring Global LLDP Commands, page 173
- Configuring Interface LLDP Commands, page 175

Configuring Global LLDP Commands

You can set global LLDP settings. These settings include the length of time before discarding LLDP information received from peers, the length of time to wait before performing LLDP initialization on any interface, the rate at which LLDP packets are sent, the port desription, system capabilities, system description, and system name.

LLDP supports a set of attributes that it uses to discover neighbor devices. These attributes contain type, length, and value descriptions and are referred to as TLVs. LLDP supported devices can use TLVs to receive and send information to their neighbors. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

The switch supports the following required management LLDP TLVs:

- Data Center Ethernet Parameter Exchange (DCBXP) TLV
- Management address TLV
- Port description TLV
- Port VLAN ID TLV ((IEEE 802.1 organizationally specific TLVs)
- System capabilities TLV
- System description TLV
- System name TLV

The Data Center Bridging Exchange Protocol (DCBXP) is an extension of LLDP. It is used to announce, exchange, and negotiate node parameters between peers. DCBXP parameters are packaged into a specific DCBXP TLV. This TLV is designed to provide an acknowledgement to the received LLDP packet.

DCBXP is enabled by default, provided LLDP is enabled. When LLDP is enabled, DCBXP can be enabled or disabled using the [**no**] **lldp tlv-select dcbxp** command. DCBXP is disabled on ports where LLDP transmit or receive is disabled.

To configure LLDP settings, perform this task:

Before You Begin

Ensure that the LLDP feature is enabled on the switch.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **lldp** {**holdtime** *seconds* | **reinit** *seconds* | **timer** *seconds* | **tlv-select** {**dcbxp** | **management-address** | **port-description** | **port-vlan** | **system-capabilities** | **system-description** | **system-name**}}
3. switch(config)# **no lldp** {**holdtime** | **reinit** | **timer**}
4. (Optional)switch#**show lldp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **lldp** {**holdtime** *seconds* | **reinit** *seconds* | **timer** *seconds* | **tlv-select** {**dcbxp** | **management-address** | **port-description** | **port-vlan** | **system-capabilities** | **system-description** | **system-name**}}	Configures LLDP options. Use the **holdtime** option to set the length of time (10 to 255 seconds) that a device should save LLDP information received before discarding it. The default value is 120 seconds. Use the **reinit** option to set the length of time (1 to 10 seconds) to wait before performing LLDP initialization on any interface. The default value is 2 seconds. Use the **timer** option to set the rate (5 to 254 seconds) at which LLDP packets are sent. The default value is 30 seconds. Use the **tlv-select** option to specify the type length value (TLV). The default is enabled to send and receive all TLVs. Use the **dcbxp** option to specify the Data Center Ethernet Parameter Exchange (DCBXP) TLV messages. Use the **managment-address** option to specify the management address TLV messages. Use the **port-description** option to specify the port description TLV messages. Use the **port-vlan** option to specify the port VLAN ID TLV messages. Use the **system-capabilities** option to specify the system capabilities TLV messages. Use the **system-description** option to specify the system description TLV messages. Use the **system-name** option to specify the system name TLV messages.

	Command or Action	Purpose
Step 3	switch(config)# **no lldp** {**holdtime** | **reinit** | **timer**}	Reset the LLDP values to their defaults.
Step 4	(Optional)switch#**show lldp**	Displays LLDP configurations.

This example shows how to configure the global LLDP hold time to 200 seconds:

```
switch# configure terminal
switch(config)# lldp holdtime 200
switch(config)#
```

This example shows how to to enable LLDP to send or receive the management address TLVs:

```
switch# configure terminal
switch(config)# lldp tlv-select management-address
switch(config)#
```

Configuring Interface LLDP Commands

To configure the LLDP feature for a physical Ethernet interface, perform this task:

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# [**no**] **lldp** {**receive** | **transmit**}
4. (Optional)switch#**show lldp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Selects the interface to change.
Step 3	switch(config-if)# [**no**] **lldp** {**receive** | **transmit**}	Sets the selected interface to either receive or transmit. The **no** form of the command disables the LLDP transmit or receive.
Step 4	(Optional)switch#**show lldp**	Displays LLDP configurations.

This example shows how to set an interface to transmit LLDP packets:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# lldp transmit
```

This example shows how to configure an interface to disable LLDP:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
```

This example shows how to display LLDP interface information:

```
switch# show lldp interface ethernet 1/2
tx_enabled: TRUE
rx_enabled: TRUE
dcbx_enabled: TRUE
Port MAC address:    00:0d:ec:a3:5f:48
Remote Peers Information
No remote peers exist
```

This example shows how to display LLDP neighbor information:

```
switch# show lldp neighbors
LLDP Neighbors

Remote Peers Information on interface Eth1/40
Remote peer's MSAP: length 12 Bytes:
00    c0    dd    0e    5f    3a    00    c0    dd    0e    5f    3a

LLDP TLV's
LLDP TLV type:Chassis ID  LLDP TLV Length: 7
LLDP TLV type:Port ID  LLDP TLV Length: 7
LLDP TLV type:Time to Live  LLDP TLV Length: 2
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 55
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 5
LLDP TLV type:END of LLDPDU  LLDP TLV Length: 0

Remote Peers Information on interface Eth1/34
Remote peer's MSAP: length 12 Bytes:
00    0d    ec    a3    27    40    00    0d    ec    a3    27    69

LLDP TLV's
LLDP TLV type:Chassis ID  LLDP TLV Length: 7
LLDP TLV type:Port ID  LLDP TLV Length: 7
LLDP TLV type:Time to Live  LLDP TLV Length: 2
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 55
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 5
LLDP TLV type:END of LLDPDU  LLDP TLV Length: 0

Remote Peers Information on interface Eth1/33
Remote peer's MSAP: length 12 Bytes:
00    0d    ec    a3    27    40    00    0d    ec    a3    27    68

LLDP TLV's
LLDP TLV type:Chassis ID  LLDP TLV Length: 7
LLDP TLV type:Port ID  LLDP TLV Length: 7
LLDP TLV type:Time to Live  LLDP TLV Length: 2
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 55
LLDP TLV type:LLDP Organizationally Specific  LLDP TLV Length: 5
LLDP TLV type:END of LLDPDU  LLDP TLV Length: 0
```

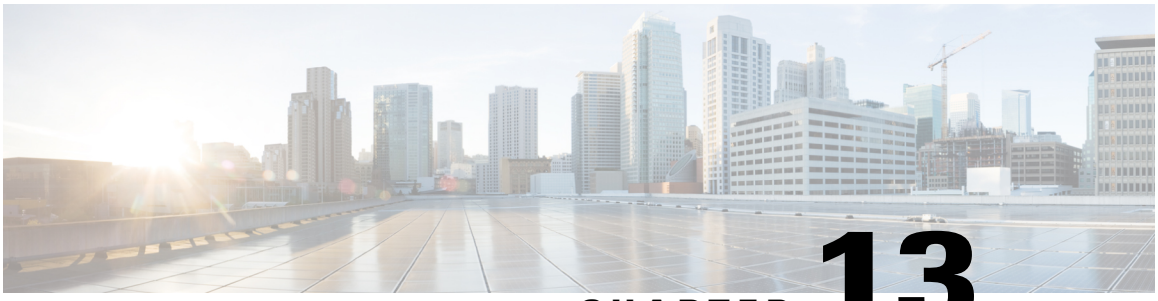
This example shows how to display LLDP timer information:

```
switch# show lldp timers
LLDP Timers
holdtime 120 seconds
reinit 2 seconds
msg_tx_interval 30 seconds
```

This example shows how to display LLDP counters:

```
switch# show lldp traffic
LLDP traffic statistics:
```

```
Total frames out: 8464
Total Entries aged: 6
Total frames in: 6342
Total frames received in error: 2
Total frames discarded: 2
Total TLVs unrecognized: 0
```



CHAPTER 13

Configuring the MAC Address Table

This chapter contains the following sections:

- Information About MAC Addresses, page 179
- Configuring MAC Addresses, page 179
- Verifying the MAC Address Configuration, page 181

Information About MAC Addresses

To switch frames between LAN ports, the switch maintains an address table. When the switch receives a frame, it associates the media access control (MAcC) address of the sending network device with the LAN port on which it was received.

The switch dynamically builds the address table by using the MAC source address of the frames received. When the switch receives a frame for a MAC destination address not listed in its address table, it floods the frame to all LAN ports of the same VLAN except the port that received the frame. When the destination station replies, the switch adds its relevant MAC source address and port ID to the address table. The switch then forwards subsequent frames to a single LAN port without flooding all LAN ports.

You can also enter a MAC address, which is termed a static MAC address, into the table. These static MAC entries are retained across a reboot of the switch.

In addition, you can enter a multicast address as a statically configured MAC address. A multicast address can accept more than one interface as its destination.

The address table can store a number of unicast and multicast address entries without flooding any frames. The switch uses an aging mechanism, defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.

Configuring MAC Addresses

Configuring a Static MAC Address

You can configure MAC addresses for the switch. These addresses are static MAC addresses.

Note You can also configure a static MAC address in interface configuration mode or VLAN configuration mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config-)# **mac-address-table static** *mac_address* **vlan** *vlan-id* {**drop** | **interface** {*type slot*/*port*} | **port-channel** *number*} [**auto-learn**]
3. (Optional) switch(config-)# **no mac-address-table static** *mac_address* **vlan** *vlan-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config-)# **mac-address-table static** *mac_address* **vlan** *vlan-id* {**drop** | **interface** {*type slot*/*port*} | **port-channel** *number*} [**auto-learn**]	Specifies a static address to add to the MAC address table. If you enable the auto-learn option, the switch will update the entry if the same MAC address is seen on a different port.
Step 3	switch(config-)# **no mac-address-table static** *mac_address* **vlan** *vlan-id*	(Optional) Deletes the static entry from the MAC address table.

This example shows how to put a static entry in the MAC address table:

```
switch# configure terminal
switch(config)# mac-address-table static 12ab.47dd.ff89 vlan 3 interface ethernet 2/1
```

You can use the **mac-address-table static** command to assign a static MAC address to a virtual interface.

Configuring the Aging Time for the MAC Table

You can configure the amount of time that an entry (the packet source MAC address and port that packet ingresses) remain in the MAC table. MAC aging time can be configured in either interface configuration mode or in VLAN configuration mode.

Note If the Cisco Nexus 3000 switch is used as a Layer 2 or Layer 3 termination switch, Cisco recommends that you set the **mac-address-table aging-time** to 1800 (higher than the default ARP aging time of 1500 seconds) on all VLANs.

Cisco Nexus 3000 switch does not support per-VLAN cam aging timers.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **mac-address-table aging-time** *seconds* [**vlan** *vlan_id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac-address-table aging-time** *seconds* [**vlan** *vlan_id*]	Specifies the time before an entry ages out and is discarded from the MAC address table. The *seconds* range is from 0 to 1000000. The default is 1800 seconds. Entering the value 0 disables the MAC aging. If a VLAN is not specified, the aging specification applies to all VLANs.

This example shows how to set the aging time for entries in the MAC address table to 1800 seconds (30 minutes):

```
switch# configure terminal
switch(config) # mac-address-table aging-time 1800
switch(config) #
```

Clearing Dynamic Addresses from the MAC Table

You can clear all dynamic entries in the MAC address table.

Command	Purpose
switch(config)# **clear mac-address-table dynamic** {**address** *mac-addr*} {**interface** [*type slot*/*port* | **port-channel** *number*} {**vlan** *vlan-id*}	Clears the dynamic address entries from the MAC address table.

This example shows how to clear the dynamic entries in the MAC address table:

```
switch# clear mac-address-table dynamic
```

Verifying the MAC Address Configuration

To display MAC address configuration information, perform one of these tasks:

Command	Purpose
switch# **show mac-address-table aging-time**	Displays the MAC address aging time for all VLANs defined in the switch.

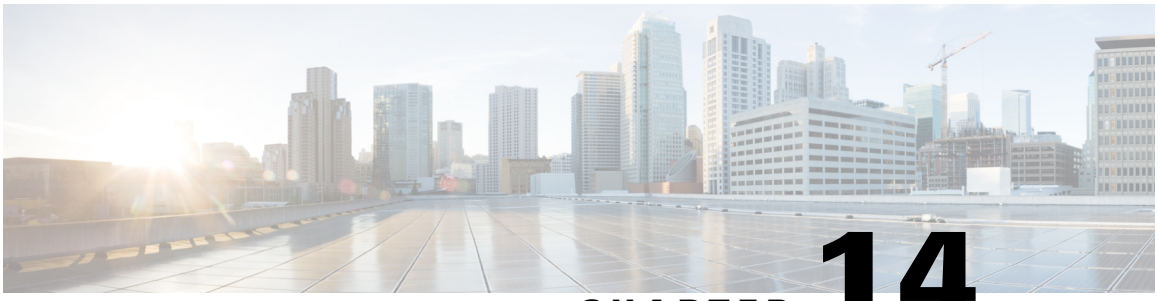
Command	Purpose
switch# **show mac-address-table**	Displays the contents of the MAC address table.

This example shows how to display the MAC address table:

```
switch# show mac-address-table
VLAN      MAC Address       Type    Age       Port
---------+-----------------+-------+---------+------------------------------
1         0018.b967.3cd0    dynamic 10        Eth1/3
1         001c.b05a.5380    dynamic 200       Eth1/3
Total MAC Addresses: 2
```

This example shows how to display the current aging time:

```
switch# show mac-address-table aging-time
Vlan  Aging Time
----- ----------
1     300
13    300
42    300
```



CHAPTER 14

Configuring IGMP Snooping

This chapter contains the following sections:

- Information About IGMP Snooping, page 183
- Configuring IGMP Snooping Parameters, page 186
- Verifying IGMP Snooping Configuration, page 188

Information About IGMP Snooping

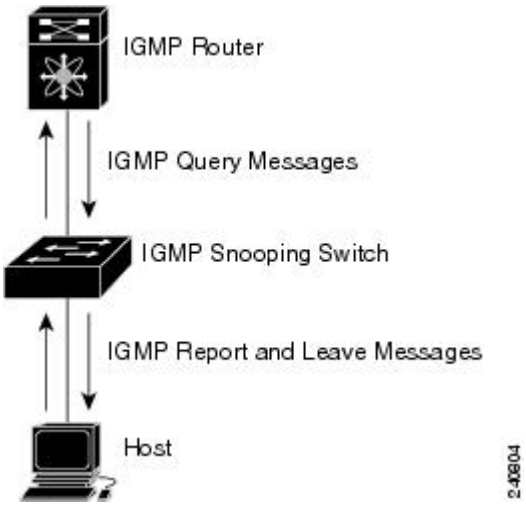
The IGMP snooping software examines IGMP protocol messages within a VLAN to discover which interfaces are connected to hosts or other devices interested in receiving this traffic. Using the interface information, IGMP snooping can reduce bandwidth consumption in a multi-access LAN environment to avoid flooding the entire VLAN. The IGMP snooping feature tracks which ports are attached to multicast-capable routers to help it manage the forwarding of IGMP membership reports. The IGMP snooping software responds to topology change notifications.

Note IGMP snooping is supported on all Ethernet interfaces. The term *snooping* is used because Layer 3 control plane packets are intercepted and influence Layer 2 forwarding decisions.

Cisco NX-OS supports IGMPv2 and IGMPv3. IGMPv2 supports IGMPv1, and IGMPv3 supports IGMPv2. Although not all features of an earlier version of IGMP are supported, the features related to membership query and membership report messages are supported for all IGMP versions.

The following figure shows an IGMP snooping switch that is located between the host and the IGMP router. The IGMP snooping switch snoops the IGMP membership reports and leave messages and forwards them only when necessary to the connected IGMP routers.

Figure 20: IGMP Snooping Switch



Note The switch supports IGMPv3 snooping based only on the destination multicast MAC address. It does not support snooping based on the source MAC address or on proxy reports.

The Cisco NX-OS IGMP snooping software supports optimized multicast flooding (OMF) that forwards unknown traffic to routers only and performs no data driven state creation. For more information about IGMP snooping, see http://tools.ietf.org/wg/magma/draft-ietf-magma-snoop/rfc4541.txt.

IGMPv1 and IGMPv2

Both IGMPv1 and IGMPv2 support membership report suppression, which means that if two hosts on the same subnet want to receive multicast data for the same group, then the host that receives a member report from the other host suppresses sending its report. Membership report suppression occurs for hosts that share a port.

If no more than one host is attached to each VLAN switch port, then you can configure the fast leave feature in IGMPv2. The fast leave feature does not send last member query messages to hosts. As soon as the software receives an IGMP leave message, the software stops forwarding multicast data to that port.

IGMPv1 does not provide an explicit IGMP leave message, so the software must rely on the membership message timeout to indicate that no hosts remain that want to receive multicast data for a particular group.

Note Cisco NX-OS ignores the configuration of last member query interval when you enable the fast leave feature because it does not check for remaining hosts.

IGMPv3

The IGMPv3 snooping implementation on the switch forwards IGMPv3 reports to allow the upstream multicast router do source-based filtering.

By default, the software tracks hosts on each VLAN port. The explicit tracking feature provides a fast leave mechanism. Because every IGMPv3 host sends membership reports, a report suppression feature limits the amount of traffic the switch sends to other multicast capable routers. When report suppression is enabled, and no IGMPv1 or IGMPv2 hosts requested the same group, the software provides proxy reporting. The proxy feature builds group state from membership reports from the downstream hosts and generates membership reports in response to queries from upstream queriers.

Even though the IGMPv3 membership reports provide a full accounting of group members on a LAN segment, when the last host leaves, the software sends a membership query. You can configure the parameter last member query interval. If no host responds before the timeout, the software removes the group state.

IGMP Snooping Querier

When there is no multicast router in the VLAN to originate the queries, you must configure an IGMP snooping querier to send membership queries.

When an IGMP snooping querier is enabled, it sends out periodic IGMP queries that trigger IGMP report messages from hosts that want to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

IGMP Forwarding

The control plane of the Cisco Nexus 3000 Series switch is able to detect IP addresses but forwarding occurs using the MAC address only.

When a host connected to the switch wants to join an IP multicast group, it sends an unsolicited IGMP join message, specifying the IP multicast group to join. Alternatively, when the switch receives a general query from a connected router, it forwards the query to all interfaces, physical and virtual, in the VLAN. Hosts wanting to join the multicast group respond by sending a join message to the switch. The switch CPU creates a multicast forwarding table entry for the group if it is not already present. The CPU also adds the interface where the join message was received to the forwarding table entry. The host associated with that interface receives multicast traffic for that multicast group.

The router sends periodic multicast general queries and the switch forwards these queries through all ports in the VLAN. Interested hosts respond to the queries. If at least one host in the VLAN wants to receive multicast traffic, the router continues forwarding the multicast traffic to the VLAN. The switch forwards multicast group traffic to only those hosts listed in the forwarding table for that multicast group.

When hosts want to leave a multicast group, they can either silently leave, or they can send a leave message. When the switch receives a leave message from a host, it sends a group-specific query to determine if any other devices connected to that interface are interested in traffic for the specific multicast group. The switch then updates the forwarding table for that MAC group so that only those hosts interested in receiving multicast traffic for the group are listed in the forwarding table. If the router receives no reports from a VLAN, it removes the group for the VLAN from its IGMP cache.

Configuring IGMP Snooping Parameters

To manage the operation of the IGMP snooping process, you can configure the optional IGMP snooping parameters described in the following table.

Table 14: IGMP Snooping Parameters

Parameter	Description
IGMP snooping	Enables IGMP snooping on a per-VLAN basis. The default is enabled. **Note** If the global setting is disabled, then all VLANs are treated as disabled, whether they are enabled or not.
Explicit tracking	Tracks IGMPv3 membership reports from individual hosts for each port on a per-VLAN basis. The default is enabled.
Fast leave	Enables the software to remove the group state when it receives an IGMP Leave report without sending an IGMP query message. This parameter is used for IGMPv2 hosts when no more than one host is present on each VLAN port. The default is disabled.
Last member query interval	Sets the interval that the software waits after sending an IGMP query to verify that no hosts that want to receive a particular multicast group remain on a network segment. If no hosts respond before the last member query interval expires, the software removes the group from the associated VLAN port. Values range from 1 to 25 seconds. The default is 1 second.
Snooping querier	Configures a snooping querier on an interface when there is no multicast router in the VLAN to generate queries. The default is disabled.
Report suppression	Limits the membership report traffic sent to multicast-capable routers. When you disable report suppression, all IGMP reports are sent as is to multicast-capable routers. The default is enabled.
Multicast router	Configures a static connection to a multicast router. The interface to the router must be in the selected VLAN.
Static group	Configures an interface belonging to a VLAN as a static member of a multicast group.

You can disable IGMP snooping either globally or for a specific VLAN.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip igmp snooping**
3. switch(config)# **vlan** *vlan-id*
4. switch(config-vlan)# **ip igmp snooping**
5. switch(config-vlan)# **ip igmp snooping explicit-tracking**
6. switch(config-vlan)# **ip igmp snooping fast-leave**
7. switch(config-vlan)# **ip igmp snooping last-member-query-interval** *seconds*
8. switch(config-vlan)# **ip igmp snooping querier** *IP-address*
9. switch(config-vlan)# **ip igmp snooping report-suppression**
10. switch(config-vlan)# **ip igmp snooping mrouter interface** *interface*
11. switch(config-vlan)# **ip igmp snooping static-group** *group-ip-addr* [**source** *source-ip-addr*] **interface** *interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **ip igmp snooping**	Globally enables IGMP snooping. The default is enabled. **Note** If the global setting is disabled, then all VLANs are treated as disabled, whether they are enabled or not.
Step 3	switch(config)# **vlan** *vlan-id*	Enters VLAN configuration mode.
Step 4	switch(config-vlan)# **ip igmp snooping**	Enables IGMP snooping for the current VLAN. The default is enabled. **Note** If IGMP snooping is enabled globally, this command is not required.
Step 5	switch(config-vlan)# **ip igmp snooping explicit-tracking**	Tracks IGMPv3 membership reports from individual hosts for each port on a per-VLAN basis. The default is enabled on all VLANs.
Step 6	switch(config-vlan)# **ip igmp snooping fast-leave**	Supports IGMPv2 hosts that cannot be explicitly tracked because of the host report suppression mechanism of the IGMPv2 protocol. When you enable fast leave, the IGMP software assumes that no more than one host is present on each VLAN port. The default is disabled for all VLANs.
Step 7	switch(config-vlan)# **ip igmp snooping last-member-query-interval** *seconds*	Removes the group from the associated VLAN port if no hosts respond to an IGMP query message before the last member query interval expires. Values range from 1 to 25 seconds. The default is 1 second.
Step 8	switch(config-vlan)# **ip igmp snooping querier** *IP-address*	Configures a snooping querier when you do not enable PIM because multicast traffic does not need to be routed. The IP address is used as the source in messages. The default is disabled.

	Command or Action	Purpose
Step 9	switch(config-vlan)# **ip igmp snooping report-suppression**	Limits the membership report traffic sent to multicast-capable routers. When you disable report suppression, all IGMP reports are sent as is to multicast-capable routers. The default is enabled.
Step 10	switch(config-vlan)# **ip igmp snooping mrouter interface** *interface*	Configures a static connection to a multicast router. The interface to the router must be in the selected VLAN. You can specify the interface by type and number.
Step 11	switch(config-vlan)# **ip igmp snooping static-group** *group-ip-addr* [**source** *source-ip-addr*] **interface** *interface*	Configures an interface belonging to a VLAN as a static member of a multicast group. You can specify the interface by type and number.

The following example shows configuring IGMP snooping parameters for a VLAN:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# ip igmp snooping last-member-query-interval 3
switch(config-vlan)# ip igmp snooping querier 172.20.52.106
switch(config-vlan)# ip igmp snooping explicit-tracking
switch(config-vlan)# ip igmp snooping fast-leave
switch(config-vlan)# ip igmp snooping report-suppression
switch(config-vlan)# ip igmp snooping mrouter interface ethernet 1/10

switch(config-vlan)# ip igmp snooping static-group 230.0.0.1 interface ethernet 1/10
switch(config-vlan)# end
```

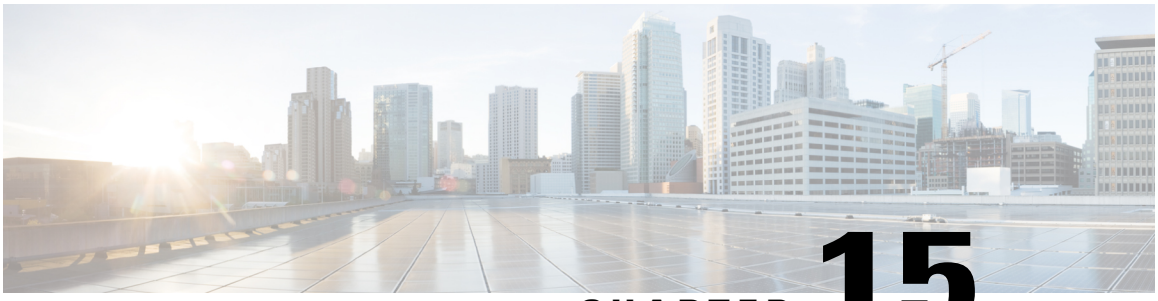
Verifying IGMP Snooping Configuration

To verify the IGMP snooping configuration, perform one of these tasks:

Command	Description
switch# **show ip igmp snooping** [[**vlan**] *vlan-id*]	Displays IGMP snooping configuration by VLAN.
switch# **show ip igmp snooping groups** [[**vlan**] *vlan-id*] [**detail**]	Displays IGMP snooping information about groups by VLAN.
switch# **show ip igmp snooping querier** [[**vlan**] *vlan-id*]	Displays IGMP snooping queriers by VLAN.
switch# **show ip igmp snooping mrouter** [[**vlan**] *vlan-id*]	Displays multicast router ports by VLAN.
switch# **show ip igmp snooping explicit-tracking vlan** *vlan-id*	Displays IGMP snooping explicit tracking information by VLAN.

The following example shows how to verify the IGMP snooping parameters:

```
switch# show ip igmp snooping
Global IGMP Snooping Information:
  IGMP Snooping enabled

IGMP Snooping information for vlan 1
  IGMP snooping enabled
  IGMP querier none
  Switch-querier disabled
  Explicit tracking enabled
  Fast leave disabled
  Report suppression enabled
  Router port detection using PIM Hellos, IGMP Queries
  Number of router-ports: 0
  Number of groups: 0
IGMP Snooping information for vlan 5
IGMP snooping enabled
  IGMP querier present, address: 172.16.24.1, version: 3
  Querier interval: 125 secs
  Querier last member query interval: 10 secs
  Querier robustness: 2
  Switch-querier enabled, address 172.16.24.1, currently running
  Explicit tracking enabled
  Fast leave enabled
  Report suppression enabled
  Router port detection using PIM Hellos, IGMP Queries
  Number of router-ports: 1
  Number of groups: 1
```



CHAPTER 15

Configuring Traffic Storm Control

This chapter contains the following sections:

- Information About Traffic Storm Control, page 191
- Traffic Storm Guidelines and Limitations, page 192
- Configuring Traffic Storm Control, page 193
- Traffic Storm Control Example Configuration, page 194
- Default Traffic Storm Settings, page 194

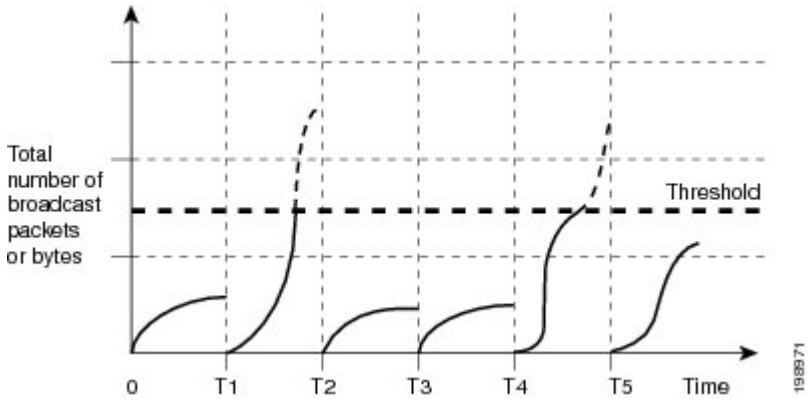
Information About Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. You can use the traffic storm control feature to prevent disruptions on Ethernet interfaces by a broadcast, multicast, or unknown unicast traffic storm.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming broadcast, multicast, and unicast traffic over a 10-microsecond interval. During this interval, the traffic level, which is a percentage of the total available bandwidth of the port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends.

The following figure shows the broadcast traffic patterns on an Ethernet interface during a specified time interval. In this example, traffic storm control occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

Figure 21: Broadcast Suppression



The traffic storm control threshold numbers and the time interval allow the traffic storm control algorithm to work with different levels of packet granularity. For example, a higher threshold allows more packets to pass through.

Traffic storm control on the Cisco Nexus 3000 Series switch is implemented in the hardware. The traffic storm control circuitry monitors packets that pass from an Ethernet interface to the switching bus. Using the Individual/Group bit in the packet destination address, the circuitry determines if the packet is unicast or broadcast, tracks the current count of packets within the 10-microsecond interval, and filters out subsequent packets when a threshold is reached.

Traffic storm control uses a bandwidth-based method to measure traffic. You set the percentage of total available bandwidth that the controlled traffic can use. Because packets do not arrive at uniform intervals, the 10-microsecond interval can affect the operation of traffic storm control.

The following are examples of how traffic storm control operation is affected:

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all broadcast traffic until the end of the interval.
- If you enable multicast traffic storm control, and the multicast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all multicast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all broadcast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all multicast traffic until the end of the interval.

By default, Cisco NX-OS takes no corrective action when the traffic exceeds the configured level.

Traffic Storm Guidelines and Limitations

When configuring the traffic storm control level, follow these guidelines and limitations:

- You can configure traffic storm control on a port-channel interface.
- Specify the level as a percentage of the total interface bandwidth:
  - The level can be from 0 to 100.
  - The optional fraction of a level can be from 0 to 99.
  - 100 percent means no traffic storm control.
  - 0.0 percent suppresses all traffic.
- There are local link and hardware limitations that prevent storm-control drops from being counted separately. Instead, storm-control drops are counted with other drops in the indiscards counter.
- Multicast-storm control configuration does not apply on mutlicast data packets with a link-local destination MAC address of 01-00-5e-00-00-xx.
- Because of hardware limitations and the method by which packets of different sizes are counted, the level percentage is an approximation. Depending on the sizes of the frames that make up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points.

Configuring Traffic Storm Control

You can set the percentage of total available bandwidth that the controlled traffic can use.

Note Traffic storm control uses a 10-microsecond interval that can affect the operation of traffic storm control.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {**ethernet** *slot*/*port* | **port-channel** *number*}
3. switch(config-if)# **storm-control** {**broadcast** | **multicast** | **unicast**} **level** *percentage*[.*fraction*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **interface** {**ethernet** *slot*/*port* | **port-channel** *number*}	Enters interface configuration mode.
Step 3	switch(config-if)# **storm-control** {**broadcast** | **multicast** | **unicast**} **level** *percentage*[.*fraction*]	Configures traffic storm control for traffic on the interface. The default state is disabled.

This example shows how to configure unicast traffic storm control for Ethernet interface 1/4:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# storm-control unicast level 40
```

Verifying Traffic Storm Control Configuration

To display traffic storm control configuration information, perform one of these tasks:

Command	Purpose
switch# **show interface** [**ethernet** *slot*/*port* | **port-channel** *number*] **counters storm-control**	Displays the traffic storm control configuration for the interfaces. **Note** Traffic storm control uses a 10-microsecond interval that can affect the operation of traffic storm control.
switch# **show running-config interface**	Displays the traffic storm control configuration.

Traffic Storm Control Example Configuration

The following example shows how to configure traffic storm control:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# storm-control broadcast level 40
switch(config-if)# storm-control multicast level 40
switch(config-if)# storm-control unicast level 40
```

Default Traffic Storm Settings

The following table lists the default settings for traffic storm control parameters.

Table 15: Default Traffic Storm Control Parameters

Parameters	Default
Traffic storm control	Disabled
Threshold percentage	100

INDEX

802.1Q VLANs 44, 53
  configuring 53
  private VLANs 44

A

adding ports 63
  port channels 63
aging time, configuring 180
  MAC table 180

B

blocking state, STP 115
BPDU guard 160
bridge ID 106
broadcast storms 191

C

changed information 1
  description 1
channel mode 67
  port channels 67
channel modes 61
  port channels 61
CIST regional root 134
CIST root 135
community ports 34
community VLANs 34, 35
configuring 29
  VLANs 29
configuring LACP 66

D

debounce timer 11
  parameters 11
debounce timer, configuring 20
  Ethernet interfaces 20

E

Ethernet interfaces 20
  debounce timer, configuring 20
extended range VLANs 26

G

guidelines and limitations 82
  vPCs 82

H

hardware hashing 65
  multicast traffic 65
host ports 34
  kinds of 34

I

ICMPv2 184
IEEE 802.1w 131
IGMP forwarding 185
  MAC address 185
IGMP snooping 185
  queries 185
IGMPv1 184
IGMPv3 185
interface information, displaying 21
  layer 2 21

interface speed 10, 13
configuring 13
interfaces 7, 8
chassis ID 7
options 7
UDLD 8
isolated port 34
isolated VLANs 34, 35

L

LACP 55, 59, 60, 62, 66, 69
configuring 66
marker responders 62
port channels 59
port priority 69
system ID 60
LACP-enabled vs static 62
port channels 62
layer 2 21
interface information, displaying 21
Layer 2 switching 3
Ethernet switching 3
Link Aggregation Control Protocol 55
Link Failure 118, 136
detecting unidirectional 118
load balancing 64
port channels 64
configuring 64

M

MAC table 180
aging time, configuring 180
MST 134, 143
CIST regional root 134
setting to default values 143
MSTP 131, 132, 133, 134, 135, 136, 143
boundary ports 136
described 136
CIST regional root 134
CIST root 135
CIST, described 133
CST 133, 134
defined 133
operations between regions 134
IEEE 802.1s 134
terminology 134
IST 133, 134
operations within a region 133
mapping VLANs to MST instance 143
MSTP *(continued)*
MST region 131, 132, 133, 135
CIST 133
described 131
hop-count mechanism 135
supported spanning-tree instances 132
multicast storms 191
multicast traffic 65
hardware hashing 65
port channels 65

N

native 802.1Q VLANs 53
configuring 53
new information 1
description 1

P

parameters, about 11
debounce timer 11
physical Ethernet settings 24
port channel 70
verifying configuration 70
port channeling 55
port channels 55, 56, 58, 59, 62, 63, 64, 65, 67
adding ports 63
channel mode 67
compatibility requirements 56
creating 62
hardware hashing 65
LACP 59
LACP-enabled vs static 62
load balancing 58, 64
port channels 58
STP 55
port profiles 11
about 11
guidelines and limitations 11
port profiles 11
PortFast BPDU filtering 161
ports 30
adding to VLANs 30
primary VLANs 34
private VLANs 34, 35, 37, 44
802.1Q VLANs 44
community VLANs 34, 35
end station access to 37
isolated trunk 37
isolated VLANs 34, 35

private VLANs *(continued)*
ports 34
community 34
isolated 34
promiscuous 34
primary VLANs 34
promiscuous trunk 37
secondary VLANs 34
promiscuous ports 34

R

rapid PVST priority 126
rapid PVST+ configurations 130
verifying 130
Rapid Spanning Tree Protocol 131
reduced MAC address 106
reserved-range VLANs 26
root guard 162
RSTP 110, 113, 117, 131
active topology 113
BPDU 117
processing 117
designated port, defined 113
designated switch, defined 113
proposal-agreement handshake process 110
rapid convergence 110
point-to-point links 110
root ports 110
root port, defined 113

S

secondary VLANs 34
SFP+ transceiver 10
Small form-factor pluggable (plus) transceiver 10
STP 55, 110, 115, 116, 159, 160
edge ports 110, 159
network ports 160
normal ports 160
port channel 55
STP *(continued)*
port types 159
PortFast 110, 159
understanding 115, 116
Blocking State 115
disabled state 116
forwarding state 116
learning state 115
STP bridge ID 106
STP root guard 162

T

traffic storms 192
control 192

U

UDLD 8, 9
aggressive mode 9
defined 8
nonaggressive mode 9
unicast storms 191
Unidirectional Link Detection 8

V

verifying 31, 130
rapid PVST+ configurations 130
VLAN configurations 31
VLAN configurations 31
verifying 31
VLANs 26, 29, 30, 44
adding ports to 30
configuring 29
extended range 26
private 44
reserved range 26
vPCs 82
guidelines and limitations 82