Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter NX-OS Configuration Guide

Release 4.1(2)E1(1)
October 2009

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-19953-01

Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter NX-OS Configuration Guide

CONTENTS

Switch Security Features

Quality of Service

Preface

This preface describes the audience, organization, and conventions of the *Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter NX-OS Configuration Guide*. It also provides information on how to obtain related documentation.

Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

Organization

This guide is organized as follows:

Chapter	Title	Description
Chapter 1	Product Overview	Presents an overview of the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.
Part 1	Configuration Fundamentals	Contains chapters on using the CLI and initial switch configuration.
Part 2	LAN Switching	Contains chapters on how to configure Ethernet interfaces, VLANs, STP, Port Channels, trunks, the MAC address table, and IGMP snooping.
Part 3	Switch Security Features	Contains chapters on how to configure AAA, Radius, TACACS+, SSH/Telnet, and ACLs.
Part 4	System Management	Contains chapters on how to configure CFS, RBAC, System Message Logging, Call Home, SNMP, RMON, network management interfaces, storm control, and SPAN.
Part 5	FIP Snooping	Contains information about Fibre Channel Initialization Protocol and FIP snooping.
Part 6	Quality of Service	Contains chapters on how to configure QoS.

Chapter	Title	Description
Part 7	IBM BladeCenter-Specific Features	Contains information about features that are specific for the IBM BladeCenter.
Part 8	Troubleshooting	Contains chapters on how to perform basic troubleshooting.

Document Conventions

Command descriptions use these conventions:

boldface font	Commands and keywords are in **boldface**.
italic font	Arguments for which you supply values are in *italics*.
[ ]	Elements in square brackets are optional.
[**x** | **y** | **z**]	Optional alternative keywords are grouped in brackets and separated by vertical bars.

Screen examples use these conventions:

`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

- *Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter Hardware Installation Guide*
- *Regulatory Compliance and Safety Information for the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter*
- *Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter Getting Started Guide*

- *Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter NX-OS Command Reference*
- *Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter NX-OS Release Notes*
- *Cisco NX-OS System Messages Reference*

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

Product Overview

This chapter provides an overview of the Cisco NX-OS software and includes the following sections:

Cisco NX-OS Software for the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter

This section describes the Cisco NX-OS software for the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

The Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter (also referred in this document as the *switch*) is a Layer 2 device, which runs Cisco NX-OS. The Cisco NX-OS Release 4.1(2)1(1) software supports the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter including certain features that are specific to the product. The Cisco NX-OS software also interoperates with any networking operating system that conforms to the IEEE and RFC compliance standards.

The switch is a 10/1-Gb Ethernet switch for the IBM BladeCenter chassis. The switch offers a solution in high-end data centers where server virtualization and I/O consolidation are required.

This section includes the following topics:

Common Software Throughout the Data Center

The Cisco NX-OS software provides a unified operating system that is designed to run all areas of the data center network including the LAN and Layer 4 through Layer 7 network services.

Modular Software Design

The Cisco NX-OS software supports distributed multithreaded processing on symmetric multiprocessors (SMPs), multi-core CPUs, and distributed data module processors. The Cisco NX-OS software offloads computationally intensive tasks, such as hardware table programming, to dedicated processors distributed across the data modules. The modular processes are created on demand, each in a separate protected memory space. Processes are started and system resources are allocated only when you enable a feature. A real-time preemptive scheduler helps to ensure the timely processing of critical functions.

Serviceability

The Cisco NX-OS software has serviceability functions that allow the device to respond to network trends and events. These features help you with network planning and improving response times.

This section includes the following topics:

Switched Port Analyzer

The Switched Port Analyzer (SPAN) feature allows you to analyze all traffic between ports (called the SPAN source ports) by nonintrusively directing the SPAN session traffic to a SPAN destination port that has an external analyzer attached to it. For more information about SPAN, see Chapter 34, “Configuring SPAN.”

Ethanalyzer

Ethanalyzer is a Cisco NX-OS protocol analyzer tool based on the Wireshark (formerly Ethereal) open source code. Ethanalyzer is a command-line version of Wireshark for capturing and decoding packets. You can use Ethanalyzer to troubleshoot your network and analyze the control-plane traffic. For more information about Ethanalyzer, see Chapter 35, “Troubleshooting.”

Call Home

The Call Home feature continuously monitors hardware and software components to provide e-mail-based notification of critical system events. A versatile range of message formats is available for optimal compatibility with pager services, standard e-mail, and XML-based automated parsing applications. It offers alert grouping capabilities and customizable destination profiles.You can use this

feature, for example, to directly page a network support engineer, send an e-mail message to a network operations center (NOC), and employ Cisco AutoNotify services to directly generate a case with the Cisco Technical Assistance Center (TAC). For more information about Call Home, see Chapter 25, "Configuring Call Home."

Online Diagnostics

The Online Health Management System (OHMS) is a hardware fault detection and recovery feature. It ensures the general health of the switch. For more information about OHMS, see Chapter 24, "Configuring Online Diagnostics."

Manageability

This section includes the following topics:

Simple Network Management Protocol

The Cisco NX-OS software is compliant with Simple Network Management Protocol (SNMP) version 1, version 2, and version 3. A large number of MIBs is supported. For more information about SNMP, see the "System Management" section.

Role-Based Access Control

With role-based access control (RBAC), you can limit access to device operations by assigning roles to users. You can customize access and restrict it to the users who require it. For more information about RBAC, see the "Switch Security Features" section.

Cisco NX-OS Device Configuration Methods

You can configure devices using the CLI from a Secure Shell (SSH) session or a Telnet session. SSH provides a secure connection to the switch. For more information on SSH and Talent, see the "Switch Security Features" section.

You can also configure devices using the XML management interface, which is a programmatic method based on the NETCONF protocol that complements the CLI.

Traffic Routing, Forwarding, and Management

This section includes the following topics:

Ethernet Switching

The Cisco NX-OS software supports high-density, high-performance Ethernet systems and provides the following Ethernet switching features:

- IEEE 802.1D-2004 Rapid and Multiple Spanning Tree Protocols (802.1w and 802.1s)
- IEEE 802.1Q VLANs and trunks
- 512-subscriber VLANs
- IEEE 802.3ad link aggregation
- Private VLANs
- Unidirectional Link Detection (UDLD) in aggressive and standard modes

IP Multicast

The Cisco NX-OS includes the following multicast protocols and functions:

- Internet Group Management Protocol (IGMP) Versions 1, 2, and 3 router role
- IGMPv2 host mode
- IGMP snooping

FCoE Initialization Protocol

The Cisco NX-OS supports the FIP snooping bridge feature. The switch operates as a loss-less Ethernet bridge transparently forwarding FCoE packets.

Quality of Service

The Cisco NX-OS Quality of Service (QoS) support allows you to classify the network traffic, police and prioritize the traffic flow, and provide congestion avoidance. For more information, see the "Quality of Service" section.

Network Security Features

Cisco NX-OS includes the following security features:

- Authentication, authorization, and accounting (AAA)

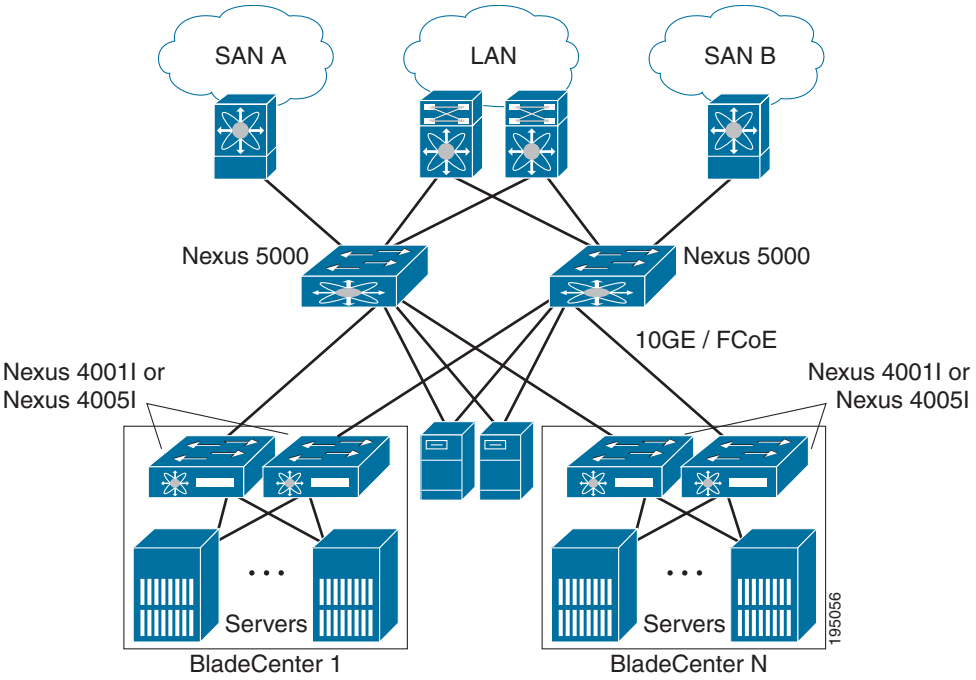
- RADIUS and TACACS+
- SSH Protocol Version 2
- SNMPv3
- Policies based on MAC and IPv4 addresses supported by named ACLs (port-based ACLs [PACLs], VLAN-based ACLs [VACLs])
- Traffic storm control (unicast, multicast, and broadcast)

For more information, see the "Switch Security Features".

Typical Deployment Topology

The switch is typically deployed in a topology that is shown in Figure 1-1:

Figure 1-1 Typical Deployment Topology



Supported Standards

Table 1-1 lists the IEEE standards supported by the switch.

Table 1-1 IEEE Compliance

Standard	Description
802.1D	MAC Bridges
802.1s	Multiple Spanning Tree Protocol
802.1w	Rapid Spanning Tree Protocol
802.3ad	Link aggregation with LACP
802.3ab	1000BaseT (10/100/1000 Ethernet over copper)
802.3ae	10-Gigabit Ethernet
802.1Q	VLAN Tagging
802.1p	Class of service Tagging for Ethernet frames
802.1x	Port-based network access control

PART 1

Configuration Fundamentals

CHAPTER 2

Configuring the Switch

This chapter describes basic switch configuration functions. This chapter includes the following sections:

Image Files on the Switch

The Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter have the following images:

- BIOS and loader images combined in one file
- Kickstart image
- System image that includes a BIOS image that can be upgraded

The switch has flash memory that consists of two separate flash parts:

- A 2 MB flash part holds two BIOS and loader images.
- A 1 GB flash part holds configuration files, kickstart images, systems images, and other files.

The upgradeable BIOS and the golden BIOS are programmed onto the 2 MB flash part. You cannot upgrade the golden BIOS.

When you download a new pair of kickstart and system images, you also get a new BIOS image because it is included in the system image. You can use the **install all** command to upgrade the kickstart, system, and upgradeable BIOS images.

This section includes the following topics:

Starting the Switch

A switch starts its boot process when it is completely inserted in the slot in the IBM BladeCenter. The switch does not have a power switch.

Booting Mechanism

The switch has a redundant boot BIOS mechanism.

When the switch boots, it checks for the location of the BIOS. If the primary bootflash contains a valid BIOS, the switch boots from there. If the primary bootflash is not there or is corrupted, the switch checks for the secondary BIOS and boots from there.

The BIOS runs a number of tests, and if any test fails, the boot process stops and provides the loader prompt to the user.

Console Settings

The loader, kickstart, and system images have the following factory default console settings:

- Speed—9600 baud
- Databits—8 bits per byte
- Stopbits—1 bit
- Parity—none

These settings are stored on the switch, and all three images use the stored console settings.

To change a console setting, use the **line console** command in configuration mode. The following example configures a line console and sets the options for that terminal line:

```
switch# configure terminal
switch(config)# line console
switch(config-console)# databits 7
switch(config-console)# exec-timeout 30
switch(config-console)# parity even
switch(config-console)# stopbits 2
```

You cannot change the BIOS console settings. These are the same as the default console settings.

Upgrading the Switch

Note Users with the network-administrator role can upgrade the software image on the switch.

This section includes the following topics:

Upgrade Procedure Summary

To upgrade the switch software, perform the following steps:

Step 1 Log in to the console port on the switch.

Step 2 Log in to Cisco.com and download the kickstart and system images to a server.

Step 3 Download the kickstart and system images to the switch using the **copy** command.

Step 4 Install the images using the **install all** command.

Caution While the switch performs the installation, all traffic through the switch is disrupted.

Detailed Upgrade Procedure

Caution Upgrading a switch disrupts all traffic flow.

To upgrade the software on the switch, perform the following steps:

Step 1 Log in to the switch on the console port connection.

Step 2 Log in to Cisco.com to access the Software Download Center. To log in to Cisco.com, go to the URL http://www.cisco.com/ and click **Log In** at the top of the page. Enter your Cisco username and password.

Note Unregistered Cisco.com users cannot access the links provided in this document.

Step 3 Access the Software Download Center using this URL:

http://www.cisco.com/kobayashi/sw-center/index.shtml

Step 4 Navigate to the software downloads for the switch.

You see links to the download images for the switch.

Step 5 Read the release notes for the related image file.

Step 6 Select and download the kickstart and system software files to a server.

Step 7 Ensure that the required space is available in the bootflash: directory for the image file(s) to be copied:

```
switch# dir bootflash:
         49     Jul 20 16:09:07 2009  ..tmp-kickstart
         26     Jul 20 16:09:08 2009  ..tmp-system
       1347     Jan 22 10:54:09 2009  StartupConfigFile
       4096     Jan 06 00:13:05 2009  TCLscipts/
       9244     Aug 12 07:17:06 2009  aclqosapi.log
   20834304     Aug 15 09:40:33 2009  block_kic.bin
   73822994     Aug 15 10:02:45 2009  block_sys.bin
   26001920     Feb 16 22:38:46 2009  diag-dce1ru-4.0.1a.bin
       4096     Dec 08 12:09:05 2008  electra/
    1537212     Aug 14 10:08:27 2009  fipsm.out
    4130990     Aug 15 14:09:11 2009  fwm_cm.bin
     399430     Sep 02 01:03:18 2009  klm_solm.klm
      78007     Sep 02 01:02:58 2009  libsolmcli.so
       1210     Mar 04 00:40:45 2009  linux-gdbserver.sh
       2155     Apr 01 00:13:32 2009  local.isan.init
      49152     Jul 20 16:09:07 2009  lost+found/
    3514473     Aug 31 18:52:18 2009  n4000_dplug.4.1.2.E1.0.175.gbin
   20768768     Aug 29 23:17:18 2009  n4000_kickstart.4.1.2.E1.0.174.gbin
   20272128     Aug 31 18:53:33 2009  n4000_kickstart.4.1.2.E1.0.175.gbin
   74497726     Aug 31 19:13:35 2009  n4000_system.4.1.2.E1.0.175.gbin
   73946627     Sep 09 11:36:46 2009  n4000_system.4.1.2.E1.0.189.bin
    6190714     Sep 02 06:56:42 2009  netstack.bin
       4096     Apr 14 03:01:17 2009  newer-fs/
   73844616     Aug 15 16:29:00 2009  pc_block.sys.bin
       4096     Mar 26 22:11:09 2009  plugin/
     493264     Sep 02 00:59:10 2009  solm.bin
       5446     Sep 02 00:59:31 2009  solm.cli
   20683264     Aug 14 11:18:28 2009  taishan_kickstart.4.1.2.E1.0.164.bin
   73837768     Aug 14 11:07:30 2009  taishan_system.4.1.2.E1.0.164.bin
       4096     Mar 19 21:26:29 2009  vdc_2/
       4096     Mar 19 21:26:29 2009  vdc_3/
       4096     Mar 19 21:26:29 2009  vdc_4/

Usage for bootflash://sup-local
  846000128 bytes used
     671744 bytes free
  846671872 bytes total
```

Tip We recommend that you keep the kickstart and system image files for at least one previous software release to use if the new image files do not load successfully.

Step 8 Install the new images, specifying the new image names that you downloaded:

```
switch(config)# install all kickstart bootflash:n4000-bk9-kickstart.4.1.2.E1.1.bin system
bootflash:n4000-bk9.4.1.2.E1.1.bin
```

The **install** command performs the following actions:

- Performs compatibility checks (equivalent to the **show incompatibility** command) for the images that you have specified. If there are compatibility issues, an error message is displayed and the installation does not proceed.
- Displays the compatibility check results and displays whether the installation is disruptive.
- Provides a prompt to allow you to continue or abort the installation.

Note A disruptive installation causes traffic disruption while the switch reboots.

- Updates the boot variables to reference the specified images and saves the configuration to the startup configuration file.

Step 9 After the switch completes the installation, log in and verify that the switch is running the required software version:

```
switch# show version
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2009, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained herein are owned by
other third parties and are used and distributed under license.
Some parts of this software are covered under the GNU Public
License. A copy of the license is available at
http://www.gnu.org/licenses/gpl.html.

Software
  BIOS:      version 1.0.28
  loader:    version N/A
  kickstart: version 4.1(2)E1(1) [build 4.1(2)E1(1)] [gdb]
  system:    version 4.1(2)E1(1) [build 4.1(2)E1(1)] [gdb]
  BIOS compile time:       06/17/09
  kickstart image file is: bootflash:///n4000_kickstart.4.1.2.E1.1.gbin
  kickstart compile time:  8/28/2009 23:00:00 [08/29/2009 10:06:33]
  system image file is:    bootflash:///n4000_system.4.1.2.E1.1.gbin
  system compile time:     8/28/2009 23:00:00 [08/29/2009 09:50:46]


Hardware
  cisco Nexus4010 Chassis ("20x10GE/supervisor")
  Motorola, e500v2  with 2076512 kB of memory.
  Processor Board ID JAB1303003F

  Device name: n4000
  bootflash:     589836 kB

Kernel uptime is 0 day(s), 7 hour(s), 52 minute(s), 10 second(s)

Last reset at 899227 usecs after  Thu Sep  3 07:27:36 2009

  Reason: Reset Requested by CLI command reload
  System version: 4.1(2)E1(0.170)
  Service:

plugin
  Core Plugin, Ethernet Plugin
```

Downgrading from a Higher Release

The procedure to downgrade the switch is identical to a switch upgrade, except that the image files to be loaded are for an earlier release than the image currently running on the switch.

Note Prior to downgrading to a specific release, check the release notes for the current release installed on the switch, to ensure that your hardware is compatible with the specific release.

To downgrade the software on the switch, perform the following steps:

Step 1 Locate the image files you will use for the downgrade by entering the **dir bootflash:** command.

If the image files are not stored on the bootflash memory, download the files from Cisco.com (using steps 1 through 9 of the software upgrade procedure).

Step 2 Install the new images.

```
switch(config)# install all kickstart bootflash:n4000-bk9-kickstart.4.1.2.E1.1.bin system
bootflash:n4000-bk9.4.1.2.E1.1.bin
```

The **install all** command performs the following actions:

- Performs compatibility checks (equivalent to the **show incompatibility** command) for the images that you have specified. If there are compatibility issues, an error message is displayed and the installation does not proceed.
- Displays the compatibility check results and displays whether the installation is disruptive.
- Provides a prompt to allow you to continue or abort the installation.

Note A disruptive installation causes traffic disruption while the switch reboots.

- Updates the boot variables to reference the specified images and saves the configuration to the startup configuration file.

Step 3 After the switch completes the installation, log in and verify that the switch is running the required software version:

```
switch# show version
```

Initial Configuration

The section includes the following topics:

Configuration Prerequisites

The following procedure is a review of the tasks you should have completed during hardware installation. These tasks must be completed before you can configure the switch.

Before you configure a switch, perform the following steps:

Step 1 Verify the following physical connections for the new switch:

- The console port is physically connected to a computer terminal (or terminal server).
- The management Ethernet port (mgmt0) is connected to an external hub, switch, or router.

See the *Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter Hardware Installation Guide* for more information.

Tip Save the host ID information for future use (for example, to enable licensed features). The host ID information is provided in the Proof of Purchase document that accompanies the switch.

Step 2 Verify that the default console port parameters are identical to those of the computer terminal (or terminal server) attached to the switch console port:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit

Initial Setup

The first time that you access a switch in the IBM BladeCenter, it runs a setup program that prompts you for the IP address and other configuration information necessary for the switch to communicate over the Ethernet interface. This information is required to configure and manage the switch.

Note The IP address can only be configured from the CLI. When the switch powers on for the first time, you should assign the IP address.

Preparing to Configure the Switch

Before you configure the switch for the first time, you need the following information:

- Administrator password.

Note If a password is weak (short, easy-to-decipher), your password configuration is rejected. Be sure to configure a strong password.

- If you are using an IPv4 address for the management interface, you need the following information:
  - IPv4 subnet mask for the switch management interface.
  - IPv4 address of the default gateway (optional).
- SSH service on the switch (optional).

  To enable this service, select the type of SSH key (dsa/rsa/rsa1) and number of SSH key bits (768 to 2048).
- NTP server IPv4 address (optional).
- SNMP community string (optional).
- switch name (optional).

  This is your switch prompt.
- An additional login account and password (optional).

Note If you are using IPv4, be sure to configure the IPv4 route, the IPv4 default network address, and the IPv4 default gateway address to enable SNMP access.

Default Login

The switch has the network administrator as a default user (admin). You cannot change the default user at any time.

There is no default password so you must explicitly configure a strong password. If a password is trivial (short, easy-to-decipher), your password configuration is rejected. Be sure to configure a strong password. If you configure and subsequently forget this new password, you have the option to recover this password.

Note If you enter a **write erase** command and reload the switch, you must reconfigure the default user (admin) password using the setup procedure.

Configuring the Switch

This section describes how to initially configure the switch.

Note Press **Ctrl-C** at any prompt to skip the remaining configuration options and proceed with what you have configured up to that point. Entering the new password for the administrator is a requirement and cannot be skipped.

Tip If you do not want to answer a previously configured question, or if you want to skip answers to any questions, press **Enter**. If a default answer is not available (for example, switch name), the switch uses what was previously configured and skips to the next question.

Once the switch boots up and displays the initial configuration dialog on the serial console connection, you can configure the switch.

To enter the basic configuration parameters, perform the following steps:

Step 1 Use a terminal emulator to access the console port of the switch.

You can now configure the switch.

Note Register the switch immediately with your supplier. Failure to register may affect response times for the initial service call. The device must be registered to receive entitled support services.

Step 2 Enter the basic configuration information.

The following example shows how to start the basic configuration setup:

```
switch# setup


         ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.

*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.
```

Step 3 Enter the setup mode by entering yes.

The following example shows how to enter the setup mode:

```
Would you like to enter the basic configuration dialog (yes/no): yes
```

Step 4 Create additional accounts by entering yes (no is the default).

The following example shows how to create additional accounts:

```
Create another login account (yes/no) [n]: y
```

a. Enter the user login ID:

```
Enter the User login Id : <ID>
```

b. Enter the user password:

```
    Enter the password for "qatest": <password>
Please enter a valid password.

    Confirm the password for "qatest":<password>
Please enter a valid password.
```

c. Enter the default user role:

```
Enter the user role [network-operator]:<role>
```

Step 5 Configure an SNMP community string by entering yes.

The following example shows how to configure an SNMP community string:

```
Configure read-only SNMP community string (yes/no) [n]: y

   SNMP community string : <string>
```

Step 6 Enter a name for the switch.

The following example shows how to enter the switch name:

```
Enter the switch name : ibm-switch-1
```

Step 7 Configure out-of-band management by entering yes.

The following example shows how to configure out-of-band management:

```
Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]:

   Mgmt0 IPv4 address : 10.10.10.1

   Mgmt0 IPv4 netmask : 255.255.255.0
```

Step 8 Configure the IPv4 default gateway (recommended) by entering yes. You can then enter its IP address.

The following example shows how to configure the default gateway:

```
Configure the default gateway? (yes/no) [y]:

   IPv4 address of the default gateway : 10.10.10.100
```

Step 9 Enable the Telnet service by entering yes.

The following example shows how to enable the Telnet service:

```
Enable the telnet service? (yes/no) [y]:
```

Step 10 Enable the SSH service by entering yes.

The following example shows how to enable the SSH service:

```
Enable the ssh service? (yes/no) [n]:
```

Step 11 Configure the NTP server by entering yes.

The following example shows how to configure the NTP server:

```
Configure the ntp server? (yes/no) [n]:

The following configuration will be applied:
  username qatest password <user-password> role network-operator
```

```
  snmp-server community topspin ro
  switchname ibm-switch-1
interface mgmt0
ip address 10.10.10.1 255.255.255.0
no shutdown
ip route 0.0.0.0/0 10.10.10.100
  telnet server enable
  no ssh server enable
```

Step 12 Configure the FCOE service by entering **yes** (or **y**), as in the following example (the default is no):

```
Enable FCOE service? (yes/no) [n]:
```

After the prompt for the FCOE service, the configuration displays.

```
The following configuration will be applied:
  username qatest password <user-password> role network-operator
  snmp-server community topspin ro
  switchname ibm-switch-1
interface mgmt0
ip address 10.10.10.1 255.255.255.0
no shutdown
ip route 0.0.0.0/0 10.10.10.100
  telnet server enable
  no ssh server enable

Would you like to edit the configuration? (yes/no) [n]:

Use this configuration and save it? (yes/no) [y]: y
```

Step 13 If you want to make changes to the displayed configuration, enter **yes** (or **y**); otherwise accept the default (no) by pressing **Enter**.

If you enter yes, the setup utility returns to the beginning of the setup, and repeats each step.

Step 14 Save this configuration by entering **yes** (or **y**), as in the following example (the default is no):.

If you do not save the configuration at this point, none of your changes are part of the configuration the next time the device reboots. Saving the configuration also automatically configures the boot variables for the kickstart and system images.

```
Use this configuration and save it? (yes/no) [y]: y

ibm-switch-1 #
```

Note The switch has two out-of-band management interfaces. The AMM configuration is mgmt1. Mgmt 0 must be placed on a different subnet than mgmt1.

Changing the Initial Configuration

To make changes to the initial configuration at a later time, enter the **setup** command in EXEC mode:

```
switch# setup

---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.

*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no): yes
```

The setup utility guides you through the basic configuration process.

Accessing the Switch

After the initial configuration, you can access the switch in a number of ways:

- Serial console access—You can use a serial port connection to access the CLI.
- Out-of-band access—You can use Telnet or SSH to access a switch.

Additional Switch Configuration

This section includes the following topics:

Assigning a Switch Name

Each switch in the network requires a unique name. You can assign names to easily identify the switch by its physical location, its network association, or the organization to which it is deployed. The assigned name is displayed in the command-line prompt. The switch name is limited to 20 alphanumeric characters.

Note This guide refers to the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter as the switch, and it uses the switch# prompt.

To change the name of the switch, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# switchname myswitch1` `myswitch1(config)#`	Changes the switch name prompt as specified (myswitch1).
Step 3	`myswitch1(config)# no switchname` `switch(config)#`	Reverts the switch name prompt to its default (switch#).

Configuring Date, Time, and Time Zone

The switch use Coordinated Universal Time (UTC), which is the same as Greenwich Mean Time (GMT). To change the default time on the switch, perform this task:

Command	Purpose
`switch# clock set` *HH:MM:SS DD Month YYYY*	Sets the default time on the switch. HH represents hours in 24-hour time (15 for 3 P.M.), MM is minutes (58), SS is seconds (09), DD is the date (29), Month is the month in words (February), and YYYY is the year (2008).

The following example sets the time for the switch:

```
switch# clock set 15:58:09 29 February 2009
Mon Feb 20 15:58:09 UTC 2009
```

Note The **clock** command changes are saved across system resets.

You can specify a time zone for the switch. To specify the local time without the daylight saving time feature, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# clock timezone` *timezone hours_offset minutes_offset*	Sets the time zone. timezone is the three letter time zone (PST for Pacific Standard), the hours offset from UTC (-8 for the PST offset), and minutes offset (needed for time zones such as Newfoundland Standard (NST) or India Standard (IST)).
Step 3	`switch(config)# exit`	Returns to EXEC mode.
Step 4	`switch# show clock`	Verifies the time zone configuration.
Step 5	`switch# show run`	Displays changes made to the time zone configuration along with other configuration information.

The following example sets the time zone to Pacific Standard Time (PST) and offsets the UTC time by negative eight hours and 0 minutes:

```
switch# configure terminal
switch(config)# clock timezone PST -8 0
```

To disable the local time setting, perform this task:

`switch(config)# no clock timezone`	Disables the time zone adjustment feature.

Adjusting for Daylight Saving Time or Summer Time

You can configure your switch to adjust for daylight saving time (or summer time). By default, Cisco NX-OS does not automatically adjust for daylight saving time. You must manually configure the switch to adjust to the daylight saving time.

For example, following U.S. standards (defined by the *Energy Policy Act* of 2005), you can have the switch advance the clock one hour at 2:00 a.m. on the second Sunday in March and move back the clock one hour at 2:00 a.m. on the first Sunday in November. You can also explicitly specify the start and end dates and times and whether or not the time adjustment recurs every year.

To enable the daylight saving time clock adjustment, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# clock summer-time` *timezone start_week start_day start_month start_time end_week end_day end_month end_time offset*	Sets the daylight savings time for a specified time zone. The start and end values are as follows: • Week ranging from 1 through 5 • Day ranging from Sunday through Saturday • Month ranging from January through December The daylight offset ranges from 1 through 1440 minutes, which are added to the start time and deleted time from the end time.
	`switch(config)# no clock summer-time`	Disables the daylight saving time adjustment feature.
Step 3	`switch(config)# exit`	Returns to EXEC mode.
Step 4	`switch# show running-config | include summer-time`	Verifies the time zone configuration.

The following example adjusts the daylight savings time for the U.S. Pacific daylight time by 60 minutes starting the second Sunday in March at 2 a.m. and ending the first Sunday in November at 2 a.m:

```
switch# configure terminal
switch(config)# clock summer-time PDT 1 Sunday March 02:00 5 Sunday November 02:00 60
```

NTP Configuration

A Network Time Protocol (NTP) server provides a precise time source (radio clock or atomic clock) to synchronize the system clocks of network devices. NTP is transported over User Datagram Protocol UDP/IP. All NTP communications use Coordinated Universal Time (UTC). An NTP server receives its time from a reference time source, such as a radio clock or atomic clock, attached to the time. NTP distributes this time across the network.

This section includes the following sections:

About NTP

In a large enterprise network, having one time standard for all network devices is critical for management reporting and event logging functions when trying to correlate interacting events logged across multiple devices. Many enterprise customers with extremely mission-critical networks maintain their own stratum-1 NTP source.

Time synchronization happens when several frames are exchanged between clients and servers. The switches in client mode know the address of one or more NTP servers. The servers act as the time source and receive client synchronization requests.

By configuring an IP address as a peer, the switch will obtain and provide time as required. The peer is capable of providing time on its own and is capable of having a server configured. If both these instances point to different time servers, your NTP service is more reliable. Even if the active server link is lost, you can still maintain the right time due to the presence of the peer.

Tip If an active server fails, a configured peer helps in providing the NTP time. Provide a direct NTP server association and configure a peer to ensure backup support if the active server fails.

If you only configure a peer, the most accurate peer takes on the role of the NTP server and the other peer(s) acts as a peer(s).

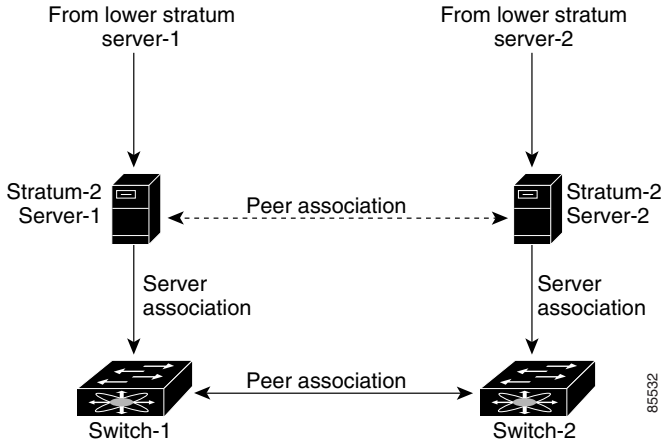
NTP Configuration Guidelines

The following guidelines apply to all NTP configurations:

- You should have a peer association with another switch only when you are sure that your clock is reliable (which means that you are a client of a reliable NTP server).
- A peer configured alone takes on the role of a server and should be used as backup. If you have two servers, then you can have several switches point to one server, and the remaining switches to the other server. You would configure peer association between these two sets, which forces the clock to be more reliable.
- If you only have one server, it is better for all the switches to have a client association with that server.

Not even a server down time will affect well-configured switches in the network. Figure 2-1 displays a network with two NTP stratum 2 servers and two switches.

Figure 2-1 NTP Peer and Server Association



In this configuration, the switches were configured as follows:

- Stratum 2 Server 1
  - IPv4 address–10.10.10.10
  - Stratum–2 Server-2
  - IPv4 address–10.10.10.9
- switch 1 IPv4 address–10.10.10.1
- switch 1 NTP configuration commands
  - **ntp server 10.10.10.10**
  - **ntp peer 10.10.10.2**
- switch 2 IPv4 address–10.10.10.2
- switch 2 NTP configuration commands
  - **ntp server 10.10.10.9**
  - **ntp peer 10.10.10.1**

Configuring NTP

You can configure NTP using either IPv4 addresses, IPv6 addresses, or Domain Name System (DNS) names. To configure NTP associations, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# ntp server {ip-address | ipv6-address | dns-name}`	Forms an association with a server.

	Command	Purpose
Step 3	`switch(config)# ntp peer {ip-address | ipv6-address | dns-name}`	Forms an association with a peer. You can specify multiple associations.
Step 4	`switch(config)# exit`	Returns to EXEC mode.
Step 5	`switch# copy running-config startup-config`	Saves your configuration changes to NVRAM. **Tip** This is one instance where you can save the configuration as a result of an NTP configuration change. You can enter this command at any time.
Step 6	`switch# show ntp peers`	Displays the configured server and peer associations.

Management Interface Configuration

The management interface on the switch allows multiple simultaneous Telnet or SNMP sessions. You can remotely configure the switch through the management interface (mgmt0), but first you must configure some IP parameters so that the switch is reachable. You can manually configure the management interface from the CLI.

This section includes the following topics:

About the mgmt Interface

The mgmt0 interface on Cisco NX-OS devices provides out-of-band management, which enables you to manage the device by its IPv4 or IPv6 address. The mgmt0 interface uses 10/100/1000 Ethernet.

The mgmt1 interface on Cisco NX-OS is provided for configuration using the Advanced Management Module (AMM) to manage the switch. To configure the mgmt1 interface using AMM, see the AMM documentation provided by IBM.

Note Before you begin to configure the management interface manually, obtain the switch IP address and subnet mask. Also make sure that the console cable is connected to the console port.

Configuring the Management Interface

To configure the management (mgmt0) Ethernet interface to connect over IP, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# interface mgmt 0`	Selects the management Ethernet interface on the switch and enters interface configuration submode.
Step 3	`switch(config-if)# ip address` *ipv4-address*`[/`*length*`]`	Configures the IPv4 address and its subnet mask.
	`switch(config-if)# ip address` *ipv4-address* `[`*subnet-mask*`]`	An alternative method that configures the IPv4 address and its subnet mask.
	`switch(config-if)# ipv6 address` *ipv6-address*`[/`*length*`]`	Configures the IPv6 address and its subnet mask.
Step 4	`switch(config-if)# no shutdown`	Enables the interface.
Step 5	`switch(config-if)# exit`	Returns to configuration mode.
Step 6	`switch(config)# vrf context management`	Enters VRF context management configuration mode.
Step 7	`switch(config-vrf)# ip route` *ipv4-prefix*`[/`*length*`]` *ipv4-nexthop-address*	Configures the IPv4 address of the next hop.
	`switch(config-vrf)# ipv6 route` *ipv6-prefix*`[/`*length*`]` *ipv6-nexthop-address*	Configures the IPv6 address of the next hop.
Step 8	`switch(config-if)# duplex {auto | full | half}`	(Optional) Configures the port duplex mode. The default is auto.
Step 9	`switch(config-if)#  speed {10 | 100 | 1000 | auto}`	(Optional) Configures the port speed. The default is auto.
Step 10	`switch(config-vrf)# exit`	Returns to EXEC mode.
Step 11	`switch# copy running-config startup-config`	(Optional) Saves your configuration changes to the file system.

The following example shows how to configure the management interface on mgmt 0:

```
switch# configure terminal
switch(config)# interface mgmt 0
switch(config-if)# ip address 10.65.122.252/24
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# vrf context management
switch(config-vrf)# ip route 0.0.0.0/0 10.65.122.2
switch(config-vrf)# exit
switch(config)# interface mgmt 0
switch(config-if)#
```

In some cases, a switch interface might be administratively shut down. You can check the status of an interface at any time by using the **show interface mgmt 0** command.

Displaying Management Interface Configuration

To display the management interface configuration, use the **show interface mgmt0** command:

```
switch# show interface mgmt0
mgmt0 is up
  Hardware: GigabitEthernet, address: 0005.ad00.36d8 (bia 0005.ad00.36d8)
  Internet Address is 172.29.231.220/23
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  full-duplex, 100 Mb/s
  1 minute input rate 1792 bits/sec, 2 packets/sec
  1 minute output rate 24 bits/sec, 0 packets/sec
  Rx
    136170 input packets 7896 unicast packets 119763 multicast packets
    8511 broadcast packets 10446815 bytes
  Tx
    793 output packets 14 unicast packets 723 multicast packets
    56 broadcast packets 187697 bytes
```

Shutting Down the Management Interface

To shut down the management interface (mgmt0), you use the **shutdown** command. A system prompt requests you confirm your action before it executes the command. You can use the **force** option to bypass this confirmation.

The following example shuts down the interface without using the **force** option:

```
switch# configure terminal
switch(config)# interface mgmt 0
switch(config-if)# shutdown
Shutting down this interface will drop all telnet sessions.
Do you wish to continue (y/n)? y
```

The following example shuts down the interface using the **force** option:

```
switch# configure terminal
switch(config)# interface mgmt 0
switch(config-if)# shutdown force
```

Managing the Switch Configuration

This section includes the following topics:

Displaying the Switch Configuration

You can view the ASCII form of the configuration file when required. To view the current configuration tree from the EXEC prompt, enter the **show running-config** command. If the running configuration is different from the startup configuration, enter the **show startup-config** command to view the ASCII version of the current startup configuration that was used to boot the switch if a **copy running-config startup-config** command was not entered after the reboot. Use the **show startup-config** command to view the contents of the current startup configuration.

You can also gather specific information about the entire switch configuration by entering the relevant **show** commands. Configurations are displayed based on a specified feature, interface, or module. Available **show** commands for each feature are briefly described in this section and listed at the end of each chapter.

Saving a Configuration

Use the **copy running-config startup-config** command to save the new configuration into nonvolatile storage. Once this command is entered, the running and the startup copies of the configuration are identical.

Clearing a Configuration

Use the **write erase** command to clear a startup configuration. Once this command is executed, the startup configuration of the switch reverts to factory defaults. The running configuration is not affected.

Caution The **write erase command** erases the entire startup configuration with the exception of any configuration that affects the loader functionality.

The **write erase boot** command only erases the configuration that affects the loader functionality. The loader functionality configuration includes the boot variables and the mgmt0 IP configuration information (IP address, netmask, and default gateway).

```
switch# write erase boot
```

This command will erase the boot variables and the IP configuration of interface mgmt 0.

Using Switch File Systems

This section includes the following topics:

Setting the Current Directory

The **cd** command changes the current directory level to a specified directory level. The CLI defaults to the volatile: file system. This command expects a directory name input.

Any file saved in the volatile: file system is erased when the switch reboots.

The syntax for this command is **cd** *directory name*.

This command exchanges the current directory to the root directory on the bootflash: file system:

```
switch# cd bootflash:
```

The following example changes the current directory to a mystorage directory that resides in the current directory:

```
switch# cd mystorage
```

Displaying the Current Directory

The **pwd** command displays the current directory location. The following example changes the directory and displays the current directory.

```
switch# cd bootflash:
switch# pwd
bootflash:
```

Listing the Files in a Directory

The **dir** command displays the contents of the current directory or the specified directory. The syntax for this command is **dir** *directory* or **dir** *filename*.

The following example shows how to list the files in the default volatile file system:

```
switch# dir volatile:

Usage for volatile://sup-local
          0 bytes used
   20971520 bytes free
   20971520 bytes total
```

Creating a Directory

The **mkdir** command creates a directory at the current directory level or at a specified directory level.

The syntax for this command is **mkdir** *name*.

The following example creates a directory called test in the bootflash directory:

```
switch# mkdir bootflash:test
```

The following example creates a directory called test in the current directory:

```
switch# mkdir test
```

Deleting an Existing Directory

The **rmdir** command deletes an existing directory at the current directory level or at a specified directory level. The directory must be empty to be deleted.

The syntax for this command is **rmdir** *name*.

The following example deletes the directory called test in the bootflash directory:

```
switch# rmdir bootflash:test
This is a directory.  Do you want to continue (y/n)?  [y] y
```

The **delete** command can also delete empty and nonempty directories. When you enter this command, a warning is displayed to confirm your intention to delete the directory.

The following example deletes the directory called test in the current directory:

```
switch# delete test
This is a directory.  Do you want to continue (y/n)?  [y] y
```

If the current directory is bootflash:mydir, this command deletes the bootflash:mydir/test directory.

Moving Files

The **move** command removes a file from the source directory and places it in the destination directory.

Caution If a file with the same name already exists in the destination directory, that file is overwritten by the moved file.

The following example moves the file called samplefile from the root directory to the mystorage directory:

```
switch# move bootflash:samplefile bootflash:mystorage/samplefile
```

The following example moves a file from the current directory level:

```
switch# move samplefile mystorage/samplefile
```

If the current directory is bootflash:mydir, this command moves bootflash:mydir/samplefile to bootflash:mydir/mystorage/samplefile.

Copying Files

The **copy** command copies a file between file systems within a switch.

Note Use the **dir** command to ensure that enough space is available in the target file system. If enough space is not available, use the **delete** command to remove unneeded files.

The following example copies the file called samplefile from the root directory to the mystorage directory:

```
switch# copy bootflash:samplefile bootflash:mystorage/samplefile
```

The following example copies a file from the current directory level:

```
switch# copy samplefile mystorage/samplefile
```

If the current directory is bootflash:mydir, this command copies bootflash:mydir/samplefile to bootflash:mydir/mystorage/samplefile.

Deleting Files

The **delete** command deletes a specified file or the specified directory and all its contents.

The following example shows how to delete a file from the current working directory:

```
switch# delete dns_config.cfg
```

The following example deletes the entire bootflash: directory and all its contents:

```
switch# delete bootflash:my-dir
```

Caution If you specify a directory, the **delete** command deletes the entire directory and all its contents.

Displaying File Contents

The **show file** command displays the contents of a specified file in the file system.

The following example displays the contents of a file residing in the current directory:

```
switch# show file myfile
```

Saving Command Output to a File

You can force all screen output to go to a file by appending > *filename* to any command. For example, enter **show interface > Samplefile** at the EXEC mode switch prompt to save the interface configuration to Samplefile which is a file created at the same directory level. At the EXEC mode switch prompt, enter a **dir** command to view all files in this directory, including the recently saved Samplefile.

Compressing and Uncompressing Files

The **gzip** command compresses (zips) the specified file using LZ77 coding.

The following example directs the output of the **show tech-support** command to a file (Samplefile), and then zips the file and displays the difference in the space used up in the volatile directory:

```
switch# show tech-support > Samplefile
Building Configuration ...
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
Usage for volatile://
    1527808 bytes used
   19443712 bytes free
   20971520 bytes total
switch# gzip volatile:Samplefile
switch# dir
     266069     Jul 04 00:51:03 2003 Samplefile.gz
Usage for volatile://
     266240 bytes used
   20705280 bytes free
   20971520 bytes total
```

The **gunzip** command uncompresses (unzips) LZ77 coded files.

The following example unzips the file that was compressed in the previous example:

```
switch# gunzip Samplefile
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
Usage for volatile://
    1527808 bytes used
   19443712 bytes free
   20971520 bytes total
```

CHAPTER 3

Using the Command-Line Interface

This chapter describes the command-line interface (CLI) and CLI command modes. It includes the following sections:

Accessing the Command Line Interface

You can connect to the switch using a terminal plugged into the console port. See "Console Settings" section on page 2-2 for information on how to set console port parameters.

You can also connect to the switch with Telnet or SSH. The switch supports up to eight simultaneous Telnet and SSH connections. To connect with Telnet or SSH, you need to know the hostname or IP address of the switch.

To establish a Telnet connection to the switch, perform this task:

	Command	Purpose
Step 1	**telnet** {*hostname* | *ip_addr*}	Establishes a Telnet connection from your host to the switch that you want to access.
Step 2	Login: **admin** Password: ***password***	Initiates authentication. **Note** If no password has been configured, press **Return**.
Step 3	switch# **exit**	Exits the session when finished.

Alternatively, to make an SSH connection to the switch, use the following command:

Command	Purpose
ssh {*hostname* | *ip_addr*}	Makes an SSH connection from your host to the switch that you want to access.

Using the CLI

This section includes the following topics:

Using CLI Command Modes

The switch supports two main command modes: user EXEC mode and configuration mode. The commands available to you depend on the mode you are in. To obtain a list of available commands in either mode, type a question mark (**?**) at the system prompt.

Table 3-1 lists and describes the two commonly used modes, how to enter the modes, and the resulting system prompts. The system prompt helps you identify which mode you are in and the commands that are available to you in that mode.

Table 3-1 Frequently Used switch Command Modes

Mode	Description	How to Access	Prompt
EXEC mode	Enables you to temporarily change terminal settings, perform basic tests, and display system information. **Note** Changes made in this mode are generally not saved across system resets.	At the switch prompt, enter the required EXEC mode command.	`switch#`
Configuration mode	Enables you to configure features that affect the system as a whole. **Note** Changes made in this mode are saved across system resets if you save your configuration.	From EXEC mode, enter the **configure terminal** command.	`switch(config)#`

You can abbreviate commands and keywords by entering just enough characters to make the command unique from other commands. For example, you can abbreviate the **configure terminal** command to **conf t**.

Changing Command Modes

Configuration mode, also known as terminal configuration mode, has several submodes. Each of these submodes places you further down in the prompt hierarchy. When you type **exit**, the switch backs out of the current level and returns you to the previous level. When you type **end**, the switch backs out to the user EXEC level. You can also press **Ctrl-Z** in configuration mode as an alternative to typing **end**.

Listing the Commands Used with Each Command Mode

You can display the commands available in any command mode by typing a question mark (**?**) at the switch prompt.

CLI Command Hierarchy

CLI commands are organized hierarchically, with commands that perform similar functions grouped under the same level. For example, all commands that display information about the system, configuration, or hardware are grouped under the **show** command, and all commands that allow you to configure the switch are grouped under the **configure terminal** command.

To execute a command, you enter the command by starting at the top level of the hierarchy. For example, to configure an interface, use the **config terminal** command. Once you are in configuration mode, enter the **interface** command. When you are in the interface submode, you can query the available commands.

The following example shows how to query the available command in the interface submode:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# ?
  bandwidth               Set bandwidth informational parameter
  cdp                     Configure CDP interface parameters
  channel-group           Configure port channel parameters
  delay                   Specify interface throughput delay
  description             Enter description of maximum 80 characters
  duplex                  Enter the port duplex mode
  end                     Go to exec mode
  errdisable              Configure error disable parameters
  exit                    Exit from command interpreter
  flowcontrol             Configure interface flowcontrol
  ip                      Configure IP features
  link                    Configure link
  lldp                    Configure Interface LLDP parameters
  logging                 Configure logging for interface
  mac                     MAC configuration commands
  no                      Negate a command or set its defaults
  pop                     Pop mode from stack of restore from name
  priority-flow-control   Enable/Disable PFC
  push                    Push current mode to stack or save it under name
  service-policy          Configure service policy for an interface
  shutdown                Enable/disable an interface
  snmp                    Modify SNMP interface parameters
  spanning-tree           Spanning Tree Subsystem
  speed                   Enter the port speed
  storm-control           Configure Interface storm control
```

```
  switchport            Configure switchport parameters
  udld                  UDLD protocol
  where                 Shows the cli context you are in
```

EXEC Mode Commands

When you start a session on the switch, you begin in EXEC mode. From EXEC mode, you can enter configuration mode. Most of the EXEC commands are one-time commands, such as **show** commands, which display the current configuration status.

The following commands are available in EXEC mode:

```
switch# ?
  attach            Connect to a specific linecard
  blink             Blink locator led
  callhome          Callhome commands
  cd                Change current directory
  checkpoint        Create configuration rollback checkpoint
  clear             Reset functions
  cli               CLI commands
  clock             Manage the system clock
  configure         Enter configuration mode
  copy              Copy from one file to another
  debug             Debugging functions
  debug-filter      Enable filtering for debugging functions
  delete            Delete a file or directory
  dir               List files in a directory
  echo              Echo argument back to screen (useful for scripts)
  end               Go to exec mode
  ethanalyzer       Configure cisco fabric analyzer
  event             Event Manager commands
  exit              Exit from command interpreter
  find              Find a file below the current directory
  format            Format disks
  gunzip            Uncompresses LZ77 coded files
  gzip              Compresses file using LZ77 coding
  install           Upgrade software
  mkdir             Create new directory
  modem             Modem commands
  move              Move files
  no                Negate a command or set its defaults
  ntp               Execute NTP commands
  ping              Test network reachability
  ping6             Test IPv6 network reachability
  pop               Pop mode from stack of restore from name
  push              Push current mode to stack or save it under name
  pwd               View current directory
  reload            Reboot the entire box
  rmdir             Delete a directory
  rollback          Rollback configuration
  routing-context   Set the routing context
  run-script        Run shell scripts
  send              Send message to open sessions
  setup             Run the basic SETUP command facility
  show              Show running system informationn
  show              Show running system information
  sleep             Sleep for the specified number of seconds
  ssh               SSH to another system
  ssh6              SSH to another system using IPv6 addressing
  system            System management commands
  tac-pac           Save tac info in a compressed .gz file at specific location
  tail              Display the last part of a file
```

```
  tar               Archiving operations
  telnet            Telnet to another system
  telnet6           Telnet6 to another system using IPv6 addressing
  terminal          Set terminal line parameters
  test              Test command
  traceroute        Traceroute to destination
  traceroute6       Traceroute6 to destination
  undebug           Disable Debugging functions (See also debug)
  update            Update license
  where             Shows the cli context you are in
  write             Write current configuration
  xml               Xml agent
```

Configuration Mode Commands

Configuration mode allows you to make changes to the existing configuration. When you save the configuration, these commands are saved across switch reboots. Once you are in configuration mode, you can enter interface configuration mode, zone configuration mode, and a variety of protocol-specific modes. Configuration mode is the starting point for all configuration commands.

The following commands are available in configuration mode:

```
switch# configure terminal
switch(config)# ?
  aaa                 Configure aaa functions
  access-list         Configure access control list parameters
  banner              Configure banner message
  boot                Configure boot variables
  callhome            Enter the callhome configuration mode
  cdp                 Configure CDP parameters
  class-map           Configure a class map
  cli                 Configure CLI commands
  clock               Configure time-of-day clock
  end                 Go to exec mode
  errdisable          Error disable
  exit                Exit from command interpreter
  fcoe_mgr            Config commands for Fcoe_mgr
  feature             Command to enable/disable features
  hostname            Configure system's host name
  hw-module           Enable/Disable OBFL information
  interface           Configure interfaces
  ip                  Configure IP features
  ipv6                Configure IPv6 features
  key                 Key Management
  lacp                Config commands for LACP
  license             Modify license features
  line                Configure a terminal line
  lldp                Configure global LLDP parameters
  logging             Modify message logging facilities
  lst                 Lst configuration commands
  mac                 MAC configuration commands
  mac-address-table   MAC Address Table
  monitor             Configure Ethernet SPAN sessions
  no                  Negate a command or set its defaults
  ntp                 NTP Configuration
  object-group        Configure ACL object groups
  password            Password for the user
  policy-map          Configure a policy map
  pop                 Pop mode from stack of restore from name
  port-channel        Configure port channel parameters
  push                Push current mode to stack or save it under name
  qos                 QoS Global Commands
```

```
radius              Configure RADIUS configuration
radius-server       Configure RADIUS related parameters
resequence          Resequence a list with sequence numbers
rmon                Remote Monitoring
role                Configure roles
route-map           Create route-map or enter route-map command mode
snmp-server         Configure snmp server
solm                SOL Manager
spanning-tree       Spanning Tree Subsystem
ssh                 Configure SSH parameters
switchname          Configure system's host name
system              System config command
system              System management commands
table-map           Configure a table map
udld                UDLD protocol
username            Configure user information.
vlan                Vlan commands
vrf                 Configure VRF parameters
where               Shows the cli context you are in
xml                 Xml agent
```

Using Commands

You can configure the CLI to function in two ways: configure it interactively by entering commands at the CLI prompt or create an ASCII file containing switch configuration information (use the CLI to edit and activate the file).

Listing Commands and Syntax

In any command mode, you can obtain a list of available commands by entering a question mark (?).

```
switch# ?
```

To see a list of commands that begin with a particular character sequence, type those characters followed by a question mark (?). Do not include a space before the question mark.

```
switch# co?
configure copy
```

To list keywords or arguments, enter a question mark in place of a keyword or argument. Include a space before the question mark. This form of help is called command syntax help because it reminds you which keywords or arguments are applicable based on the commands, keywords, and arguments you have already entered.

```
switch# configure ?
  <CR>
  terminal  Configure the system from terminal input
```

Tip If you are having trouble entering a command, check the system prompt and enter the question mark (**?**) for a list of available commands. You might be in the wrong command mode or using incorrect syntax.

Entering Command Sequences

In any command mode, you can begin a particular command sequence, then immediately press the **Tab** key to complete the rest of the command:

```
switch (config)# ro<Tab>
switch (config)# role <Tab>
switch (config)# role name
```

This form of help is called command completion because it completes a word for you. If several options are available for the typed letters, all options that match those letters are displayed.

Undoing or Reverting to Default Values or Conditions

You can enter the **no** form of any command to perform the following actions:

- Disable a feature

  If you want to disable a feature that was enabled:

  ```
  switch # configure terminal
  switch(config)# feature fip-snooping
  switch(config)# no feature fip-snooping
  switch(config)#
  ```

- Use the **no** form of a command in EXEC mode

  If you enter a command, you can undo the results:

  ```
  switch # blink interface ethernet 1/20
  switch# no blink interface ethernet 1/20
  ```

- Revert to the default value

  If you want to revert to the default value, you can undo the results:

  ```
  switch# configure terminal
  switch(config)# banner motd #Welcome to the switch#
  switch(config)# show banner motd
  Welcome to the switch
  switch(config)# no banner motd
  switch(config)# show banner motd
  Nexus 4000 Switch
  ```

Using Keyboard Shortcuts

You can execute an EXEC mode command from a configuration mode or submode prompt. You can enter this command from any submode within the configuration mode. The command is executed at the EXEC level, and the prompt resumes its current mode level, as in the following example.

```
switch(config)# terminal session-timeout 0
switch(config)#
```

In the preceding example, **terminal session-timeout** is an EXEC mode command:

Table 3-2 lists some useful command keys that can be used in both EXEC and configuration modes.

Table 3-2 Useful Command Keys

Command	Description
Ctrl-P	Up history
Ctrl-N	Down history
Ctrl-X-H	List history
Alt-P	History search backwards **Note** The difference between **Tab** completion and **Alt-P** or **Alt-N** is that pressing **Tab** completes the current word, while **Alt-P** and **Alt-N** completes a previously entered command.
Alt-N	History search forwards
Ctrl-G	Exit
Ctrl-Z	End
Ctrl-L	Clear session

Table 3-3 describes the commonly used configuration submodes.

Table 3-3 Common Configuration Submodes

Submode Name	From Configuration Mode, Enter:	Submode Prompt
Call home	**callhome**	switch(config-callhome)#
Interface configuration	**interface** *type slot/port*	switch(config-if)#
Line console	**line console**	switch(config-console)
Virtual terminal line	**line vty**	switch(config-line)#
Role	**role name**	switch(config-role)#
VLAN	**vlan**	switch(config-vlan)#

Using CLI Variables

The switch CLI parser supports the definition and use of variables in CLI commands. CLI variables can be used as follows:

- Entered directly on the command line.
- Passed to the child script initiated using the **run-script** command.

  The variables defined in the parent shell are available for use in the child **run-script** command process (see the "Executing Commands Specified in a Script" section on page 3-11).
- Passed as command line arguments to the **run-script** command (see the "Executing Commands Specified in a Script" section on page 3-11).

CLI variables have the following characteristics:

- You cannot reference a variable through another variable using nested references.

- You can define persistent variables that are available across switch reloads.
- You can reference only one predefined system variable, which is the TIMESTAMP variable.

User-Defined Persistent CLI Variables

You can define CLI session variables to persist only for the duration of your CLI session using the **cli var name** command in EXEC mode. CLI session variables are useful for scripts that you execute periodically.

The following example shows how to create a user-defined CLI session variable:

```
switch# cli var name testinterface ethernet 1/20
```

You can reference a variable using the syntax **$(variable)**. The following example shows how to reference a user-defined CLI session variable.

```
switch# show interface $(testinterface)
Ethernet1/20 is down (SFP not inserted)
  Hardware: 1000/10000 Ethernet, address: 0005.ad00.37b7 (bia 0005.ad00.37b7)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  Port mode is access
  unknown enum 3-duplex, 10 Gb/s
  Input flow-control is off, output flow-control is off
  Switchport monitor is off
  Last link flapped never
  Last clearing of "show interface" counters never
  1 minute input rate 0 bits/sec, 0 packets/sec
  1 minute output rate 0 bits/sec, 0 packets/sec
  Rx
    0 input packets 1773583844 unicast packets 775822475 multicast packets
    1375018549 broadcast packets 10904942378 jumbo packets 3782510632 storm supp
ression packets
    0 bytes
  Tx
    0 output packets 0 multicast packets
    0 broadcast packets 8644443928 jumbo packets
    0 bytes
    0 input error 0 short frame 0 watchdog
    0 no buffer 0 runt 2743974619 CRC 0 ecc
    0 overrun  0 underrun 0 ignored 0 bad etype drop
    0 bad proto drop 0 if down drop 0 input with dribble
    0 input discard
    0 output error 0 collision 0 deferred
    0 late collision 0 lost carrier 0 no carrier
    0 babble
    0 Rx pause 0 Tx pause
  0 interface resets
```

Use the **show cli variables** command to display user-defined CLI session variables. The following example displays user-defined CLI session variables.

```
switch# show cli variables
VSH Variable List
-----------------
TIMESTAMP="2009-08-03-21.18.38"
testinterface="ethernet 1/20"
```

Use the **cli no var name** command to remove user-defined CLI session variables. The following example removes a user-defined CLI session variable.

```
switch# cli no var name testinterface
```

Using Command Aliases

Command alias support has the following characteristics:

- Command aliases are global for all user sessions.
- Command aliases are saved across reboots.
- Commands being aliased must be typed in full without abbreviation.
- Command alias translation always takes precedence over any keyword in any configuration mode or submode.
- Command alias configuration takes effect for other user sessions immediately.
- You cannot override the default command alias **alias**, which aliases the **show cli alias** command.
- Nesting of command aliases is permitted to a maximum depth of 1. One command alias can refer to another command alias that must refer to a valid command, not to another command alias.
- A command alias always replaces the first command keyword on the command line.
- You can define command aliases for commands in any configuration submode or the EXEC mode.

Defining Command Aliases

You can define command aliases using the **cli alias name** command in configuration mode.

This following example shows how to define command aliases:

```
switch# configure terminal
switch(config)# cli alias name eth interface ethernet
switch(config)# cli alias name shintbr show interface brief
switch(config)# cli alias name shethintup shintbr | include up | include ethernet
```

You can display the command aliases defined on the switch using the **alias** default command alias.

The following example shows how to display the command aliases defined on the switch:

```
switch# alias
CLI alias commands
==================
alias      :show cli alias
gigint     :interface gigabitethernet
shintbr    :show interface brief
shethintup  :shintbr | include up | include ethernet
```

Command Scripts

This section includes the following topics:

Executing Commands Specified in a Script

The **run-script** command executes the commands specified in a file. To use this command, be sure to create the file and specify commands in the required order.

Note You cannot create the script file at the switch prompt. You can create the script file on an external machine and copy it to the bootflash: directory. This section assumes that the script file resides in the bootflash: directory.

The syntax for this command is **run-scrip**t *filename*.

The following example displays the CLI commands specified in a test file that resides in the bootflash: directory:

```
switch# show file bootflash:testfile
configure terminal
interface ethernet 1/20
no shutdown
end
show interface ethernet 1/20
```

This file output is in response to the **run-script** command executing the contents in the test file:

```
switch# show file bootflash:testfile
configure terminal
interface ethernet 1/20
no shutdown
end
show interface ethernet 1/20

switch# run-script bootflash:testfile
`configure terminal`
`interface ethernet 1/20`
`no shutdown`
`end`
`show interface ethernet 1/20`
Ethernet1/20 is up
  Hardware: 1000/10000 Ethernet, address: 0005.ad00.3e4f (bia 0005.ad00.3e4f)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  Port mode is access
  full-duplex, 10 Gb/s, media type is 10g
  Input flow-control is off, output flow-control is off
  Rate mode is dedicated
  Switchport monitor is off
  Last link flapped 00:29:47
  Last clearing of "show interface" counters never
  1 minute input rate 0 bits/sec, 0 packets/sec
  1 minute output rate 0 bits/sec, 0 packets/sec
  Rx
```

```
  188 input packets 0 unicast packets 190 multicast packets
  0 broadcast packets 0 jumbo packets 0 storm suppression packets
  20139 bytes
Tx
  188 output packets 188 multicast packets
  0 broadcast packets 0 jumbo packets
  20139 bytes
  0 input error 0 short frame 0 watchdog
  0 no buffer 0 runt 0 CRC 0 ecc
  0 overrun  0 underrun 0 ignored 0 bad etype drop
  0 bad proto drop 0 if down drop 0 input with dribble
  0 input discard
  0 output error 0 collision 0 deferred
  0 late collision 0 lost carrier 0 no carrier
  0 babble
  0 Rx pause 0 Tx pause
1 interface resets
```

Setting the Delay Time

The **sleep** command delays an action by a specified number of seconds.

The syntax for this command is **sleep** *seconds*:

```
switch# sleep 30
```

You will see the switch prompt return after 30 seconds. This command is useful within scripts. For example, if you create a command script called test-script.

```
switch# show file bootflash:test-script
discover scsi-target remote
sleep 10
show scsi-target disk
switch# run-script bootflash:test-script
```

When you execute the test-script command script, the switch software executes the **discover scsi-target remote** command, and then waits for 10 seconds before executing the **show scsi-target disk** command.

CHAPTER 4

Managing Licenses

This chapter describes how to manage licenses for a Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

Licensing allows you to access specified premium features on the switch after you install the appropriate license for that feature. This chapter contains information related to licensing types, options, procedures, installation, and management for the Cisco NX-OS software.

This chapter includes the following sections:

Licensing Terminology

The following terms are used in this chapter:

- Licensed feature—Permission to use a particular feature through a license file, a hardware object, or a legal contract. This permission is limited to the number of users, number of instances, time span, and the implemented switch.
- Licensed application—A software feature that requires a license to be used.
- License enforcement—A mechanism that prevents a feature from being used without first obtaining a license.
- Node-locked license—A license that can only be used on a particular switch using the unique host ID of the switch.
- Host IDs—A unique chassis serial number that is specific to each switch.

- Proof of purchase—A document entitling its rightful owner to use licensed features on one switch as described in that document. The proof of purchase document is also known as the claim certificate.
- Product Authorization Key (PAK)—The PAK allows you to obtain a license key from one of the sites listed in the proof of purchase document. After registering at the specified website, you will receive your license key file and installation instructions through e-mail.
- License key file—A switch-specific unique file that specifies the licensed features. Each file contains digital signatures to prevent tampering and modification. License keys are required to use a licensed feature. License keys are enforced within a specified time span.
- Missing license—If the bootflash has been corrupted after you have installed a license, that license shows as "missing." The feature still works, but the license count is inaccurate. You should reinstall the license as soon as possible.
- Incremental license—An additional licensed feature that was not in the initial license file. License keys are incremental. If you purchase some features now and others later, the license file and the software detect the sum of all features for the specified switch.
- Evaluation license—A temporary license. Evaluation licenses are time bound (valid for a specified number of days) and are not tied to a host ID (switch serial number).
- Permanent license—A license that is not time bound is called a permanent license.
- Grace period—The amount of time the features in a license package can continue functioning without a license.
- Support—If you purchased Cisco support through a Cisco reseller, contact the reseller directly. If you purchased support directly from Cisco Systems, contact Cisco TAC at the following URL:

  http://www.cisco.com/en/US/support/tsd_cisco_worldwide_contacts.html

Licensing Model

The licensing model for the Cisco NX-OS software is feature-based. Feature-based licenses make features available to the entire physical switch.

If asked by the tool to generate the license, the feature is BASIC_STORAGE_SERVICES_PKG. The BASIC_STORAGE_SERVICES_PKG includes the FIP snooping feature license.

License Installation

You can either obtain a factory-installed license (only applies to new orders) or perform a manual license installation of the license (applies to an existing switch in your network).

This section includes the following topics:

Obtaining a Factory-Installed License

You can obtain factory-installed licenses for a new switch.

To obtain a factory-installed license, perform the following steps:

Step 1 Contact your reseller or Cisco representative and request this service.

Note If you purchased Cisco support through a Cisco reseller, contact the reseller directly. If you purchased support directly from Cisco Systems, contact Cisco TAC at this URL:

http://www.cisco.com/en/US/support/tsd_cisco_worldwide_contacts.html

Your switch is shipped with the required licenses installed in the system. The proof of purchase document is sent along with the switch.

Step 2 Obtain the host ID from the proof of purchase document for future use.

You can now start to use the switch and the licensed features.

Performing a Manual Installation

All switch licenses are factory-installed. Manual installation is not required.

Obtaining the License Key File

To obtain new or updated license key files, perform the following steps:

Step 1 Use the **show license host-id** command to obtain the serial number for your switch. The host ID is also referred to as the switch serial number.

```
switch# show license host-id
License hostid: VDH=JAB1309001K
```

Tip Use the entire ID that appears after the colon (:) sign. In this example, the host ID is VDH=JAB1309001K.

Step 2 Obtain either your claim certificate or your proof of purchase document. This document accompanies every switch.

Step 3 Get the product authorization key (PAK) from either the claim certificate or the proof of purchase document.

Step 4 Locate the website URL from either the claim certificate or the proof of purchase document.

Step 5 Access the specified URL that applies to your switch and enter the switch serial number and the PAK.

The license key file is sent to you by e-mail. The license key file is digitally signed to only authorize use on the requested switch. The requested features are also enabled once the Cisco NX-OS software on the specified switch accesses the license key file.

Caution Install the license key file in the specified switch without making any modifications.

A license is either permanent or it expires on a fixed date. If you do not have a license, the grace period for using that feature starts from the first time you start using a feature offered by that license (see the "Grace Period Alerts" section on page 4-8).

Step 6 Use the **copy licenses** command in EXEC mode to save your license file to one of two locations; either the bootflash: or the volatile: directory (see the "Backing Up License Files" section on page 4-5).

Installing the License Key File

Tip If you need to install multiple licenses in any switch, be sure to provide unique file names for each license key file.

To install a license key file in any switch, perform the following steps:

Step 1 Log in to the switch.

Step 2 Copy the license file to bootflash.

```
switch# copy scp://root@10.0.0.0/root/nexus4000-licenses.lic
bootflash:nexus4000-licenses.lic vrf management
```

Step 3 Perform the installation by entering the **install license** command:

```
switch# install license bootflash:nexus4000-licenses.lic
Installing license ......done
```

Note If you provide a target name for the license key file, the file is installed with the specified name. Otherwise, the filename specified in the license key file is used to install the license.

Step 4 Back up the license file to a .tar file on bootflash: using the **copy licenses** command:

```
switch# copy licenses bootflash:/basic_license.tar
Backing up license done
```

Step 5 Exit the switch console and open a new terminal session to view the license file installed on the switch using the **show license usage** command:

```
switch# show license usage
Feature                      Ins  Lic   Status Expiry Date Comments
                                 Count
--------------------------------------------------------------------------------
BASIC_STORAGE_SERVICES_PKG    Yes   -    In use Never       -
--------------------------------------------------------------------------------
```

Note If the license meets all guidelines when the **install license** command is entered, all features continue functioning as configured.

Backing Up License Files

All installed license files can be backed up as a .tar file in the user specified location. Use the **copy licenses** command in EXEC mode to save your license file to one of two locations; bootflash: or volatile:. The following example saves all licenses to a file named basic_license.tar.

```
switch# copy licenses bootflash:/basic_license.tar
Backing up license done
```

Tip We recommend backing up your license files immediately after installing them and just before running a **write erase** command.

Caution If you erase any existing licenses, you can only install them using the **install license** command.

Identifying License Features in Use

When a Cisco NX-OS software feature is enabled, it can activate a license grace period. To identify the features active for a specific license, use the **show license usage** *license-name* command.

Use the **show license usage** command to identify all of the active features on your switch:

```
switch# show license usage

Feature                      Ins  Lic   Status Expiry Date Comments
                                 Count
--------------------------------------------------------------------------------
BASIC_STORAGE_SERVICES_PKG    Yes   -    In use Never       -
--------------------------------------------------------------------------------
```

Uninstalling Licenses

You can only uninstall a permanent license that is not in use. If you try to delete a permanent license that is currently being used, the software rejects the request with an error message. Uninstalling an unused license initiates the grace period. The grace period is measured from the first use of the feature without a license and is reset when a valid license file is installed.

Note Permanent licenses cannot be uninstalled if they are currently being used. Features turned on by permanent licenses must first be disabled, before that license is uninstalled.

Tip If you are using an evaluation license and would like to install a new permanent license, you can do so without service disruption and before the evaluation license expires. Removing an evaluation license immediately triggers a grace period without service disruption.

Caution Disable related features before uninstalling a license. The delete procedure fails if the license is in use.

To uninstall a license, perform the following steps:

Step 1 Save your running configuration to a remote server using the **copy** command (see Chapter 2, "Configuring the Switch").

Step 2 Disable the features provided by the license to be uninstalled. Enter the **show license usage** command to view the enabled features.

```
switch# show license usage

Feature                      Ins  Lic   Status Expiry Date Comments

                                 Count

--------------------------------------------------------------------------------

BASIC_STORAGE_SERVICES_PKG    Yes   -   In use Never       -

--------------------------------------------------------------------------------
```

Step 3 Uninstall the license file using the **clear license** *filename* command, where *filename* is the name of the installed license key file:

```
switch# clear license nexus4000-licenses.lic

Clearing license nexus4000-licenses.lic:

SERVER this_host ANY

VENDOR cisco

FEATURE BASIC_STORAGE_SERVICES_PKG cisco 1.0 permanent uncounted \

        HOSTID=VDH=JAB1309001K \

        NOTICE="<LicFileID>dc3-licenses.lic</LicFileID><LicLineID>0</LicLineID> \
```

```
        <PAK>dummyPak</PAK>" SIGN=5E8D227654E2
```

Step 4 Enter **yes** (yes is the default) to continue with the license update:

```
Do you want to continue? (y/n) y
Clearing license failed: License is in use
```

The license is in use in the example.

Step 5 Disable all license features before you uninstall:

```
switch# no feature fip-snooping

switch(config)# show license usage

Feature                      Ins  Lic   Status Expiry Date Comments

                                 Count

--------------------------------------------------------------------------------

BASIC_STORAGE_SERVICES_PKG    Yes   -   Unused Never       -

--------------------------------------------------------------------------------

switch(config)# clear license nexus4000-licenses.lic
Clearing license nexus4000-licenses.lic:

SERVER this_host ANY

VENDOR cisco

FEATURE BASIC_STORAGE_SERVICES_PKG cisco 1.0 permanent uncounted \

        HOSTID=VDH=JAB1309001K \

        NOTICE="<LicFileID>dc3-licenses.lic</LicFileID><LicLineID>0</LicLineID> \

        <PAK>dummyPak</PAK>" SIGN=5E8D227654E2


Do you want to continue? (y/n) y

Clearing license .....done
```

The license file is now uninstalled.

Grace Period Alerts

Cisco NX-OS gives you a 120-day grace period. This grace period starts or continues when you are evaluating a feature for which you have not installed a license.

The grace period stops if you disable a feature you are evaluating, but if you enable that feature again without a valid license, the grace period countdown continues from when it had stopped.

The grace period operates across all features in a license package. License packages can contain several features. If you disable a feature during the grace period and there are other features in that license package that are still enabled, the countdown does not stop for that license package. To suspend the grace period countdown for a license package, you must disable every feature in that license package.

Use the **show license usage** command to display grace period information for a switch:

```
switch# show license usage
Feature                      Ins  Lic   Status Expiry Date Comments
                                 Count
--------------------------------------------------------------------------------
BASIC_STORAGE_SERVICES_PKG    No    -   In use             Grace 119D 22H
--------------------------------------------------------------------------------
```

The Cisco NX-OS license counter keeps track of all licenses on a switch. If you are evaluating a feature and the grace period has started, you will receive console messages, SNMP traps, system messages, and Call Home messages on a daily basis.

The frequency of these messages become hourly during the last seven days of the grace period.

Note You cannot modify the frequency of the grace period messages.

Caution After the final seven days of the grace period, the feature is turned off and your network traffic may be disrupted. Any future upgrade will enforce license requirements and the 120-day grace period.

License Transfers Between Switches

A license is specific to the switch for which it is issued and is not valid on any other switch. If you need to transfer a license from one switch to another, contact your customer service representative.

Note If you purchased Cisco support through a Cisco reseller, contact the reseller directly. If you purchased support directly from Cisco Systems, contact Cisco TAC at this URL:

http://www.cisco.com/en/US/support/tsd_cisco_worldwide_contacts.html

Verifying the License Configuration

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

Command	Purpose
`switch# show license [brief]`	Displays information for all installed license files.
`switch# show license file`	Displays information for a specific license file.
`switch# show license host-id`	Displays the host ID for the physical switch.
`switch# show license usage`	Displays the usage information for installed licenses.

PART 2

LAN Switching

CHAPTER 5

Configuring Ethernet Interfaces

This chapter describes the configuration of the Ethernet interfaces on the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter. It includes the following sections:

Information About Ethernet Interfaces

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

On the switch, the Ethernet interfaces are enabled by default.

This section includes the following topics:

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 up to 20 ports.
- Port number
  - Port number within the group.

About the Unidirectional Link Detection Parameter

This section includes the following topics:

UDLD Overview

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.

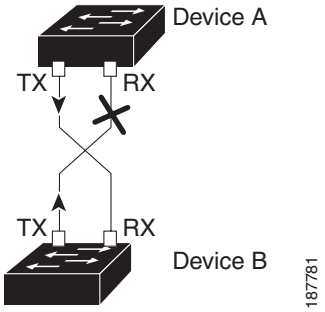
A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does 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.

The 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.

Figure 5-1 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 5-1 Unidirectional Link



Default UDLD Configuration

Table 5-1 shows the default UDLD configuration.

Table 5-1 UDLD Default Configuration

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

For information on configuring the UDLD for the device and its port, see the "Configuring the UDLD Mode" section on page 5-5.

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

The switch has external and internal switchable 10-Gigabit and 1-Gigabit Ethernet ports. Each port is equipped with SFP+ interface adapters.

About 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

Table 5-2 shows the default CDP configuration.

Table 5-2 Default CDP Configuration

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

About the Debounce Timer Parameters

The port debounce time is the amount of time that an interface waits to notify that a link is going down. During this time, the interface waits to see if the link comes back up. The wait period is a time when traffic is stopped.

You can enable the debounce timer for each interface and specify the delay time in milliseconds.

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

About MTU Configuration

A per-physical Ethernet interface maximum transmission unit (MTU) is not supported. Instead, MTU is set according to the QoS classes. You modify MTU by setting Policy and Class maps. See Chapter 30, "Configuring Quality of Service" for more details.

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

Configuring Ethernet Interfaces

This section shows how to configure Ethernet interfaces. It 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:

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

The following example shows how to enable the UDLD for the switch:

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

The following 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
```

The following 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
```

The following example shows how to disable UDLD for an Ethernet port:

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

The following example shows how to disable UDLD for the switch:

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

Configuring Interface Speed

The switch supports 6 switchable 1-Gigabit and 10-Gigabit ports. The default interface speed is 10-Gigabit. To configure these ports for 1-Gigabit Ethernet, insert a 1-Gigabit Ethernet SFP transceiver into the applicable port and then set its speed with the speed command.

To configure a 1-Gigabit Ethernet port, perform this task:

	Command	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.

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
```

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

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.

Configuring the Cisco Discovery Protocol

This section shows how to configure the Cisco Discovery Protocol (CDP). It includes the following topics:

Configuring the CDP Characteristics

You can configure the frequency of 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:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# cdp advertise {v1 | v2 }`	(Optional) Configures the version to use to send CDP advertisements. Version-2 is the default state.
Step 3	`switch(config)# 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.
Step 4	`switch(config)# 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.
Step 5	`switch(config)# 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 CDP commands to return to the default settings.

The following 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:

	Command	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.
	`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.

Configuring the Debounce Timer

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

You can show the debounce times for all of the Ethernet ports by using the **show interface debounce** command.

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

	Command	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)# link debounce time` *milliseconds*	Enables the debounce timer for the amount of time (1 to 5000 milliseconds) specified. Disables the debounce timer if you specify 0 milliseconds.

The following example shows how to enable the debounce timer and set the debounce time to 1000 milliseconds for an Ethernet interface:

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

The following example shows how to disable the debounce timer for an Ethernet interface:

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

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

Configuring the Description Parameter

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

	Command	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.

The following 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:

	Command	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.

The following example shows how to disable an Ethernet port:

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

To restart an interface, perform this task:

Command	Purpose
`switch(config-if)# no shutdown`	Restarts the interface.

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 the following 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 debounce`	Displays the debounce 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: 1000/10000 Ethernet, address: 0005.ad00.31c6 (bia 0005.ad00.31c6)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  Port mode is access
  full-duplex, 10 Gb/s, media type is 10g
  Input flow-control is off, output flow-control is off
  Rate mode is dedicated
  Switchport monitor is off
  Last link flapped 09:00:19
  Last clearing of "show interface" counters never
  1 minute input rate 104 bits/sec, 0 packets/sec
  1 minute output rate 0 bits/sec, 0 packets/sec
  Rx
    0 input packets 0 unicast packets 16740 multicast packets
    0 broadcast packets 0 jumbo packets 0 storm suppression packets
    13 bytes
  Tx
    0 output packets 1225 multicast packets
    0 broadcast packets 0 jumbo packets
    0 bytes
```

```
    0 input error 0 short frame 0 watchdog
    0 no buffer 0 runt 0 CRC 0 ecc
    5 overrun  0 underrun 5 ignored 0 bad etype drop
    0 bad proto drop 0 if down drop 0 input with dribble
    0 input discard
    0 output error 0 collision 0 deferred
    0 late collision 0 lost carrier 0 no carrier
    0 babble
    0 Rx pause 0 Tx pause
  1 interface resets
```

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

```
switch# show interface ethernet 1/1 capabilities
Ethernet1/1
  Model:                DS-C9134-K9-SUP
  Type (SFP capable):   1000base-(unknown)
  Speed:                1000,10000,auto
  Duplex:               full
  Trunk encap. type:    802.1Q
  Channel:              no
  Broadcast suppression: percentage(0-100)
  Flowcontrol:          rx-(on),tx-(on)
  Rate mode:            none
  QOS scheduling:       rx-(8q2t),tx-(1p7q4t)
  CoS rewrite:          no
  ToS rewrite:          no
  SPAN:                 yes
  UDLD:                 yes
  Link Debounce:        yes
  Link Debounce Time:   yes
  MDIX:                 no
```

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-AVAGO
    part number is SFBR-7700SDZ
    revision is B4
    serial number is AGD121020S2
    nominal bitrate is 10300 MBits/sec
    Link length supported for 50/125um fiber is 82 m(s)
    Link length supported for 62.5/125um 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        10      eth  access up      none                       10G(D) --
Eth1/2        1       eth  access down    SFP not inserted           10G(D) --
Eth1/3        10      eth  access up      none                       10G(D) --
Eth1/4        1       eth  access down    SFP not inserted           10G(D) --
Eth1/5        1       eth  access down    SFP not inserted           10G(D) --
Eth1/6        1       eth  access down    SFP not inserted           10G(D) --
Eth1/7        1       eth  access down    SFP not inserted           10G(D) --
```

```
Eth1/8        1       eth  access down    SFP not inserted           10G(D) --
Eth1/9        1       eth  access up      none                       10G(D) --
Eth1/10       1       eth  access down    SFP not inserted           10G(D) --
Eth1/11       1       eth  access down    SFP not inserted           10G(D) --
Eth1/12       1       eth  access down    SFP not inserted           10G(D) --
Eth1/13       1       eth  access up      none                       10G(D) --
Eth1/14       1       eth  access down    SFP not inserted           10G(D) --
Eth1/15       1       eth  access down    SFP not inserted           10G(D) --
Eth1/16       1       eth  access down    SFP not inserted           10G(D) --
Eth1/17       1       eth  access up      none                       10G(D) --
Eth1/18       1       eth  access down    SFP not inserted           10G(D) --
Eth1/19       1       eth  access down    SFP not inserted           10G(D) --
Eth1/20       1       eth  access down    Link not connected         10G(D) --

--------------------------------------------------------------------------------
Port   VRF          Status IP Address                              Speed    MTU
--------------------------------------------------------------------------------
mgmt0  --           up     10.65.122.252                           1000     1500
```

The following example shows how to display the link debounce status:

```
switch# show interface debounce

--------------------------------------------------------------------------------
Port          Debounce time   Value(ms)
--------------------------------------------------------------------------------
Eth1/1        enable               100
Eth1/2        enable               100
Eth1/3        enable               100
Eth1/4        enable               100
Eth1/5        enable               100
Eth1/6        enable               100
Eth1/7        enable               100
Eth1/8        enable               100
Eth1/9        enable               100
Eth1/10       enable               100
Eth1/11       enable               100
Eth1/12       enable               100
Eth1/13       enable               100
Eth1/14       enable               100
Eth1/15       enable               100
Eth1/16       enable               100
Eth1/17       enable               100
Eth1/18       enable               100
Eth1/19       enable               100
Eth1/20       enable               100
```

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

yourname-c4948         mgmt0         179    R S I     WS-C4948-10GE Gig1/4
BladeSwitch-N5K-1(SSI130205W1)Eth1/1        169    R S I s   N5K-C5020P-BF Eth1/
1
switch(JAF1251BEES)   Eth1/9        152    S I s     DS-C9134-K9   Eth1/9
switch(JAF1251BEES)   Eth1/13       154    S I s     DS-C9134-K9   Eth1/13
switch(JAF1251BEES)   Eth1/17       135    S I s     DS-C9134-K9   Eth1/17
```

Default Physical Ethernet Settings

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

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

1. MTU cannot be changed per-physical Ethernet interface. You modify MTU by selecting maps of QoS classes. See Chapter 30, "Configuring Quality of Service," for additional information.

CHAPTER 6

Configuring VLANs

You can use virtual LANs (VLANs) to divide the network into separate logical areas. VLANs can also be considered as broadcast domains.

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 a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router.

This chapter includes the following sections:

Information About VLANs

This section includes the following topics:

Understanding VLANs

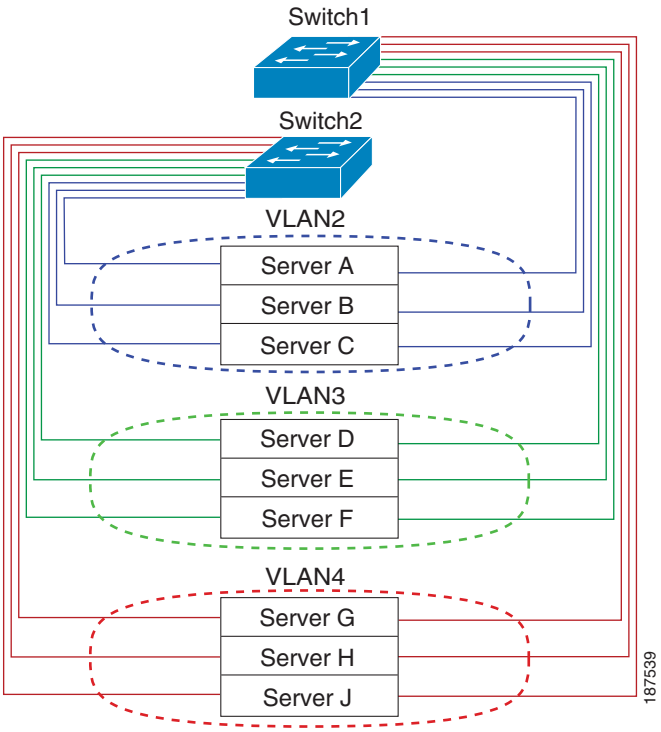
Note VLAN Trunking Protocol (VTP) mode is OFF. VTP BPDUs are dropped on all interfaces of the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter, which partitions VTP domains if other switches have VTP turned on.

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.

Figure 6-1 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 6-1 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; that is, the VLAN is in the no shutdown condition. 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 switch supports VLAN numbers 1 to 4094 in accordance with the IEEE 802.1Q standard. These VLANs are organized into ranges. You use each range slightly differently. The switch is physically limited in the number of VLANs it can support. For details of the number of supported VLANs, see the "Configuration Limits" section on page 36-1.

Table 6-1 describes the details of the VLAN ranges.

Table 6-1 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, and 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 moving 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.

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

Configuring a VLAN

This section includes the following topics:

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.

To create a VLAN, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# vlan {vlan-id | vlan-range}`	Creates a VLAN or a range or 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 default VLAN and cannot be created or deleted. You cannot create or delete those VLANs that are reserved for internal use.

The following 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.

To delete a VLAN, perform this task:

Command	Purpose
`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.

Entering the VLAN Submode and Configuring the 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; see the "Understanding VLAN Ranges" section on page 6-2 for complete information.

To enter the submode and configure the VLAN, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
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)# no shutdown`	Enables the VLAN. The default value is **no shutdown** (or enabled). You cannot shut down the default VLAN, VLAN1, or VLANs 1006 to 4094.

The following 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:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# interface {type 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 a port channel.
Step 3	`switch(config-if)# switchport access vlan vlan-id`	Sets the access mode of the interface to the specified VLAN.

The following 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 20:

```
switch# show running-config vlan 1-20
version 4.1(2)E1(1)
vlan 1
vlan 5
  name accounting
vlan 10
```

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

```
switch# show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Eth1/2, Eth1/4, Eth1/5, Eth1/6
                                                Eth1/7, Eth1/8, Eth1/9, Eth1/10
                                                Eth1/11, Eth1/12, Eth1/13
                                                Eth1/14, Eth1/15, Eth1/16
                                                Eth1/17, Eth1/18, Eth1/19
5    accounting                       active    Eth1/20
10   VLAN0010                         active    Eth1/1, Eth1/3
```

```
Remote SPAN VLANs
------------------------------------------------------------------------------

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

The following example shows the details of VLAN 5 including its member ports:

```
switch# show vlan id 5

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
5    accounting                       active

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

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

The following example shows the VLAN settings summary:

```
switch# show vlan summary

Number of existing VLANs           : 3
 Number of existing user VLANs     : 3
 Number of existing extended VLANs : 0
```

CHAPTER 7

Configuring Private VLANs

This chapter shows you how to configure private VLANs.

Note You must enable the private VLAN feature before you can perform any of the configurations in this chapter.

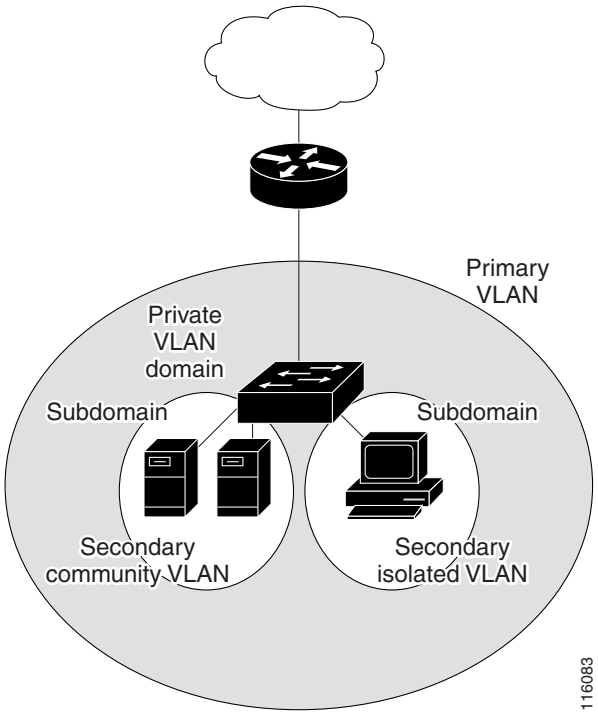
This chapter includes the following sections:

About Private VLANs

A private VLAN partitions the Layer 2 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 Figure 7-1). 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.

Note A PVLAN isolated port on the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter running the current release of Cisco NX-OS does not support IEEE 802.1q encapsulation and cannot be used as a trunk port.

Figure 7-1 Private VLAN Domain



Note You must first create the VLAN before you can convert it to a private VLAN, either primary or secondary. See Chapter 6, "Configuring VLANs" for information on creating VLANs.

This section includes the following topics:

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.

Understanding Private VLAN Ports

The types of private VLAN ports are as follows:

- Promiscuous—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, 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.
- Isolated—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 private VLAN domain, except that it can communicate with associated promiscuous ports. Private VLANs 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.
- Community—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 private VLAN domain.

Note Because trunks can support the VLANs carrying traffic between promiscuous, isolated, and community ports, the isolated and community port traffic might enter or leave the switch through a trunk interface.

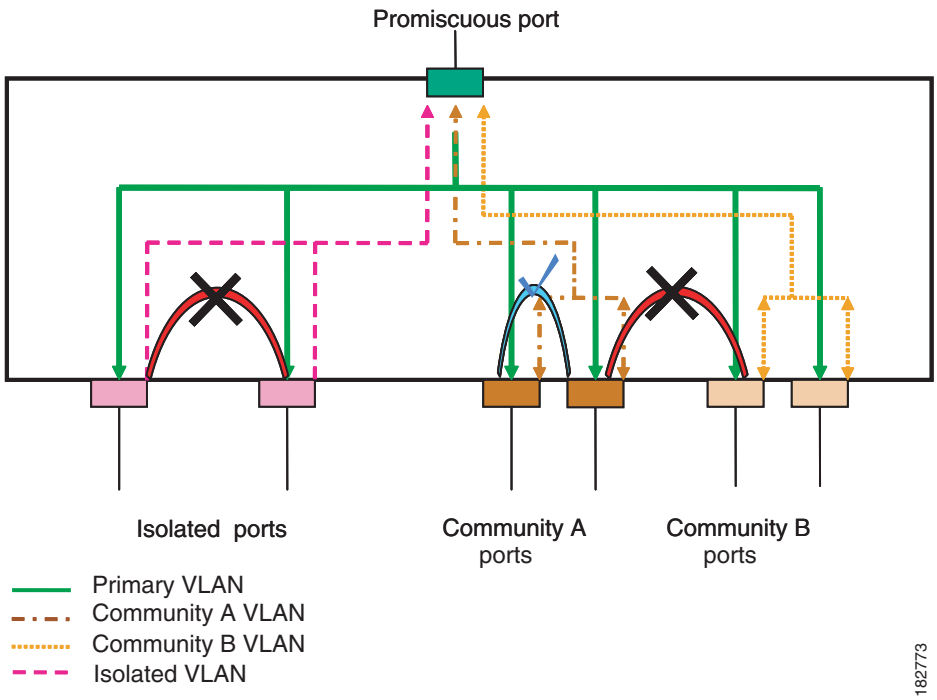
Understanding Primary, Isolated, and Community Private VLANs

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 multiple isolated VLANs in a private VLAN domain; all the traffic remains isolated within each one. Each isolated VLAN can have several isolated ports, and 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 private VLAN 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.

Figure 7-2 shows the traffic flows within a private VLAN, along with the types of VLANs and types of ports.

Figure 7-2 Private VLAN Traffic Flows



Note The private VLAN 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 normal VLAN.

A promiscuous 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 private VLAN. For example, you can use a promiscuous port to monitor or back up all the private VLAN servers from an administration workstation.

In a switched environment, you can assign an individual private VLAN 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

For 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** command to verify that the association is operational. The switch does not display an error message when the association is non operational. (See the "Verifying Private VLAN Configuration" section on page 7-10 for information on configuration verification.)

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

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

Understanding 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 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.

Understanding 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.

Configuring a Private VLAN

Note You must have already created the VLAN before you can assign the specified VLAN as a private VLAN.

This section includes the following topics:

Configuration Guidelines for Private VLANs

When configuring private VLANs, follow these guidelines:

- You must enable private VLANs before the switch can apply the private VLAN functionality.
- You cannot disable private VLANs if the switch has any operational ports in a private VLAN mode.
- Enter the **private-vlan synchronize** command to map the secondary VLANs to the same Multiple Spanning Tree (MST) instance as the primary VLAN. See the "Mapping Secondary VLANs to Same MSTI as Primary VLANs for Private VLANs" section on page 9-16 for more details.

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.

To enable private VLAN functionality on the switch, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# feature private-vlan`	Enables the private VLAN feature on the switch.

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

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

To disable private VLAN functionality, perform this task:

Command	Purpose
`switch(config)# no feature private-vlan`	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.

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. Ensure that the private VLAN feature is enabled.

To create a private VLAN, perform this task:

	Command	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.

The following 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
```

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

```
switch(config-vlan)# exit
switch(config)# vlan 100
switch(config-vlan)# private-vlan community
```

The following example shows how to assign VLAN 109 to a private VLAN as an isolated VLAN:

```
switch(config-vlan)# exit
switch(config)# vlan 109
switch(config-vlan)# private-vlan isolated
```

To disable a private VLAN, perform this task:

Command	Purpose
`switch(config-vlan)# no private-vlan {community | isolated | primary}`	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.

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 and isolated VLAN IDs.

- 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 ports that are associated with the VLAN become inactive. 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 return when you recreate the specified VLAN and configure it as the previous secondary VLAN.

Ensure that the private VLAN feature is enabled.

To associate secondary VLANs with a primary VLAN, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# vlan primary-vlan-id`	Enter 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.

The following example shows how to associate community VLANs 100 through 103 and isolated VLAN 109 with primary VLAN 5:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# private-vlan association 100-103, 109
```

To remove all associations from the private VLAN, perform this task:

Command	Purpose
`switch(config-vlan)# no private-vlan association`	Removes all associations from the primary VLAN and returns it to normal VLAN mode.

Configuring an Interface as a Private VLAN Host Port

You can configure 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. You then associate the host port with both the primary and secondary VLANs.

Note We recommend that you enable BPDU Guard on all interfaces configured as a host port. See Chapter 10, "Configuring STP Extensions" for information on configuring BPDU Guard.

Ensure that the private VLAN feature is enabled.

To configure an interface as a private VLAN host port, perform this task:

	Command	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 host port. The interface can be either a physical Ethernet port.
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.

The following example shows how to configure the 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
```

To remove the private VLAN association from an interface, perform this task:

Command	Purpose
`switch(config-if)# no switchport private-vlan host-association`	Removes the private VLAN association from the port.

Configuring an Interface as a Private VLAN Promiscuous Port

You can configure an interface as a private VLAN promiscuous port, and then you can associate that promiscuous port with the primary and secondary VLANs.

Ensure that the private VLAN feature is enabled.

To configure an interface as a private VLAN promiscuous port, perform this task:

	Command	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.

	Command	Purpose
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.

The following example shows how to configure port 1/2 as a promiscuous port associated with the primary VLAN 5 and the secondary isolated VLAN 109:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# switchport mode private-vlan promiscuous
switch(config-if)# switchport private-vlan mapping 5 109
```

You can only apply this command to a physical interface.

To clear the private VLAN mapping, perform this task:

Command	Purpose
`switch(config-if)# no switchport private-vlan mapping`	Clears the mapping from the private VLAN.

Verifying Private VLAN Configuration

To display private VLAN configuration information, perform this task:

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

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, veth1/1
5        102        community
5        103        community
5        109        isolated          Eth1/2
switch# show vlan private-vlan type
Vlan Type
---- -----------------
5    primary
```

```
100  community
101  community
102  community
103  community
109  isolated
```

The following example shows how to display enabled features:

```
switch# show system internal clis feature
 7 pvlan                           enabled
```

CHAPTER 8

Configuring Rapid PVST+

The Spanning Tree Protocol (STP) was implemented to provide a loop-free network. Rapid per VLAN Spanning Tree (Rapid PVST+) is an updated implementation of STP that allows you to create one spanning tree topology for each VLAN. Rapid PVST+ is the default STP mode on the switch.

This chapter includes the following sections:

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. See Chapter 9, "Configuring MST" for complete information on Multiple Spanning Tree (MST) and Chapter 10, "Configuring STP Extensions" for complete information on STP extensions.

Information About Rapid PVST+

This section describes the Rapid PVST+ protocol, which 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. (See the "Rapid PVST+ and IEEE 802.1Q Trunks" section on page 8-16).

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 (see the "Rapid PVST+ Interoperation with Legacy 802.1D STP" section on page 8-16).

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

This section includes an overview of Rapid PVST+ and consists of these topics:

Understanding STP

RSTP, Rapid PVST+, and MST are all extensions of the original IEEE 802.1D STP (see Chapter 9, "Configuring MST" for complete information on MST). STP is a Layer 2 loop prevention protocol that provides path redundancy while preventing undesirable loops in the network.

This section provides a basic understanding of STP in the following topics:

Overview

For an Ethernet network to function properly, only one active path can exist between any two stations. STP operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

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.

This section includes the following topics:

Bridge Priority Value

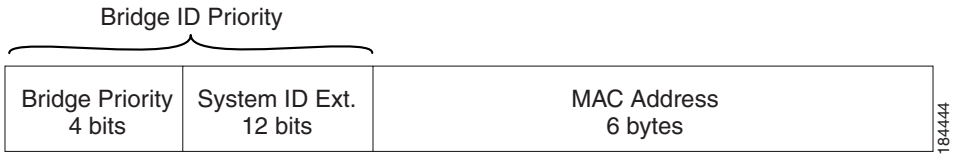
The bridge priority is a 4-bit value when the extended system ID is enabled (see "Configuring the Rapid PVST+ Bridge Priority of a VLAN" section on page 8-22).

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

Extended System ID

A 12-bit extended system ID field is part of the bridge ID (see Figure 8-1).

Figure 8-1 Bridge ID with Extended System ID



The switch always uses the 12-bit extended system ID.

Combined with the bridge ID, the system ID extension functions as the unique identifier for a VLAN (see Table 8-1).

Table 8-1 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.

See the "Rapid PVST+ BPDUs" section on page 8-8 for information about the fields that Rapid PVST+ adds to the BPDU.

Election of the Root Bridge

For each VLAN, the switch with the highest bridge ID (that is, the lowest numerical ID value) 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.

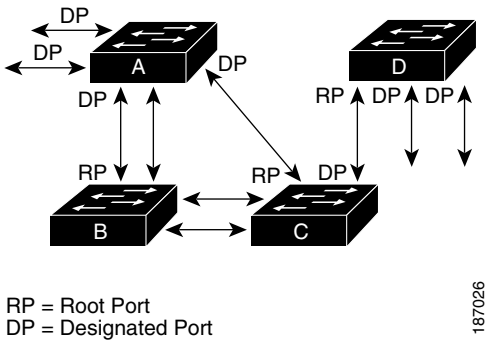
The STP root bridge is the logical center of each spanning tree topology in a network. All paths that are not needed to reach the root bridge from anywhere in the network are placed in STP blocking mode.

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 Figure 8-2, 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 8-2 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.

Understanding Rapid PVST+

This section includes the following Rapid PVST+ topics:

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 Port VLAN Identifier (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. See Chapter 10, “Configuring STP Extensions,” for more information on STP port types.

- 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.

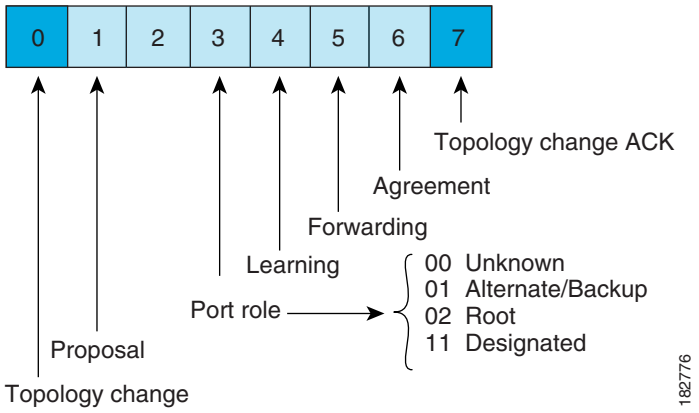
Note The TCA flag is used only when the switch is interacting with switches that are running legacy 802.1D STP. See the "Rapid PVST+ Interoperation with Legacy 802.1D STP" section on page 8-16 for information about Rapid PVST+ interaction with 802.1D STP.

The proposal and agreement sequence then quickly propagates toward the edge of the network and quickly restores connectivity after a topology change (see the "Synchronization of Port Roles" section on page 8-13).

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. Figure 8-3 shows the use of the BPDU flags in Rapid PVST+.

Figure 8-3 Rapid PVST+ Flag Byte in BPDU

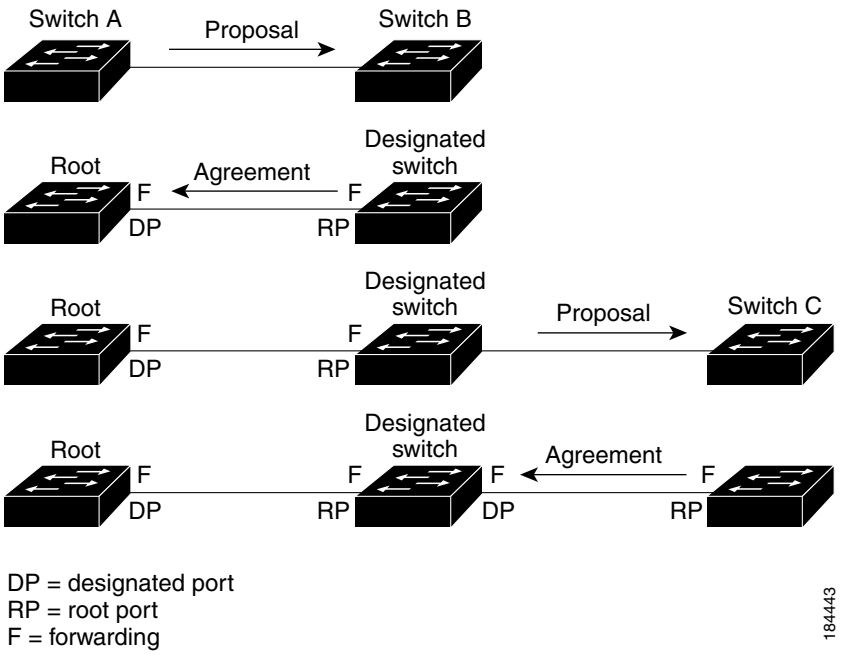


Another important change is that the Rapid PVST+ BPDU is type 2, version 2, which makes it possible for the switch to detect connected legacy (802.1D) bridges. The BPDU for 802.1D is version 0.

Proposal and Agreement Handshake

As shown in Figure 8-4, 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 8-4 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 (see Figure 8-4).

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. (See the "Port States" section on page 8-11 for information on port states.)

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.

Protocol Timers

Table 8-2 describes the protocol timers that affect the Rapid PVST+ performance.

Table 8-2 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 any port that 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

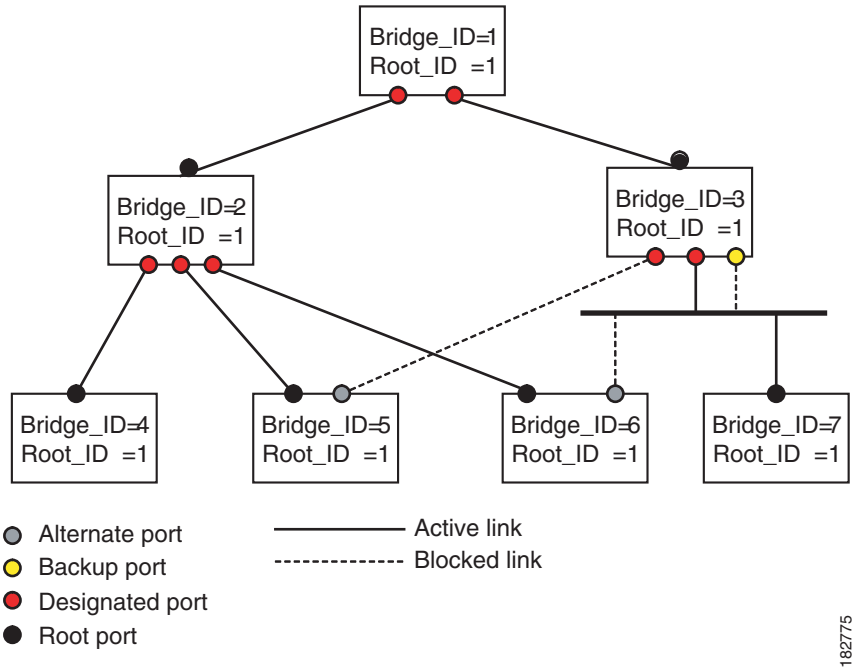
Rapid PVST+ provides rapid convergence of the spanning tree by assigning port roles and learning the active topology. Rapid PVST+ builds upon the 802.1D STP to select the switch with the highest priority (lowest numerical priority value) as the root bridge as described in the "Election of the Root Bridge" section on page 8-5. 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 Figure 8-5).

Figure 8-5 Sample Topology Demonstrating Port Roles



Port States

This section describes the Rapid PVST+ and MST port states and includes the following topics:

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:

1. The LAN port is put into the blocking state while it waits for protocol information that suggests it should go to the learning state.
2. 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.
3. In the learning state, the LAN port continues to block frame forwarding as it learns the end station location information for the forwarding database.
4. 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

Table 8-3 lists the possible operational and Rapid PVST+ states for ports and the corresponding inclusion in the active topology.

Table 8-3 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.

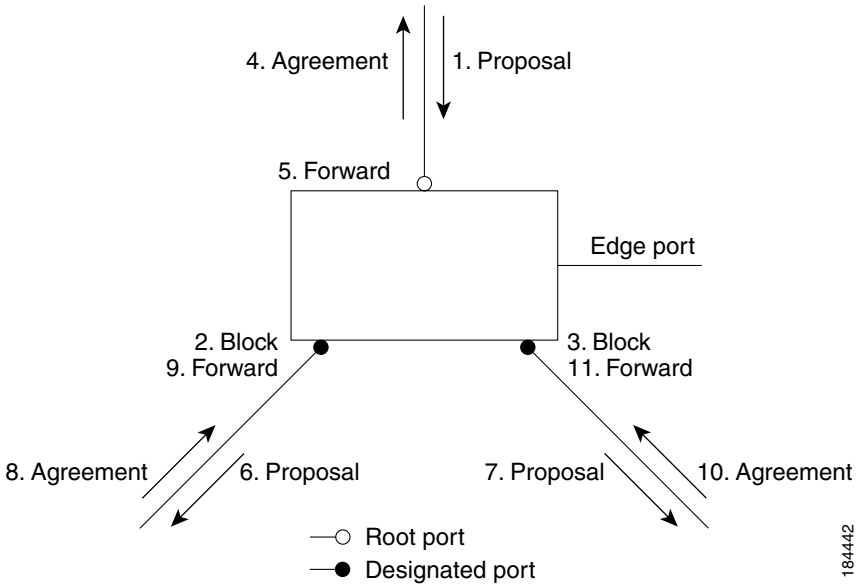
The switch is synchronized with superior root information received on the root port if all other ports are synchronized. An individual port on the switch is synchronized if either of the following applies:

- 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 Figure 8-6.

Figure 8-6 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.

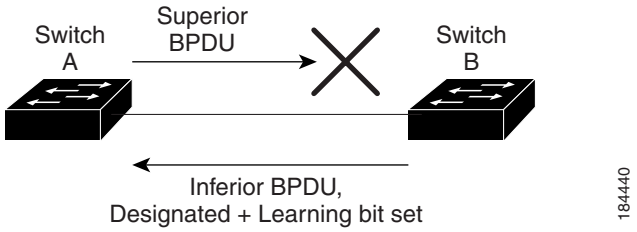
Detecting Unidirectional Link Failure

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.

Figure 8-7 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 8-7 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 (see Table 8-4). If a loop occurs, STP considers the port cost when selecting a LAN interface to put into the forwarding state.

Table 8-4 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

802.1Q trunks impose some limitations on the STP strategy for a network. 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+. See the "Restarting the Protocol" section on page 8-25 for more information.

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.

This section includes the following topics:

Enabling Rapid PVST+

Once you enable Rapid PVST+ on the switch, you must enable Rapid PVST+ on the specified VLANs (see "Enabling Rapid PVST+ per VLAN" section on page 8-18).

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.

To enable Rapid PVST+ on the switch, perform this task:

	Command	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.

The following 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** 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.

To enable Rapid PVST+ per VLAN, perform this task:

	Command	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. See Chapter 6, "Configuring VLANs."

The following example shows how to enable STP on VLAN 5:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5
```

To disable Rapid PVST+ per VLAN, perform this task:

Command	Purpose
`switch(config)# no spanning-tree vlan-range`	Disables Rapid PVST+ on the specified VLAN; see the following Caution for information regarding this command.

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 switches and bridges in a VLAN and leave it enabled on other switches and bridges in the VLAN. 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.

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** global configuration commands.

To configure a switch to become the primary root bridge for a VLAN in Rapid PVST+, perform this task:

	Command	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.

The following example shows how to configure the switch as the root bridge for VLAN 5 with a network diameter of 4:

```
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.

To configure a switch to become the secondary root bridge for a VLAN in Rapid PVST+, perform this task:

	Command	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.

The following example shows how to configure the switch as the secondary root bridge for VLAN 5 with a network diameter of 4:

```
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.

To assign Rapid PVST+ port priorities to individual ports, perform this task:

	Command	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 [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.

The following example shows how to configure the port priority of Ethernet access port 1/4 to 160:

```
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.

To set the Rapid PVST+ pathcost method and cost for a port, perform this task:

	Command	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.

	Command	Purpose
Step 3	`switch(config)# interface type slot/port`	Specifies the interface to configure, and enters the 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 port 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.

The following example shows how to configure the port cost of Ethernet access port 1/4 to 1000:

```
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.

To choose the bridge priority for a specific VLAN, perform this task:

	Command	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.

The following example shows how to configure the priority of VLAN 5 on Gigabit Ethernet port 1/4 to 8192:

```
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.

To configure the hello time for a VLAN in Rapid PVST+, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# spanning-tree vlan` *vlan-range* `hello-time value`	Configures the hello time of a VLAN. The hello time value can be from 1 to 10 seconds, and the default is 2 seconds.

The following example shows how to configure the hello time for VLAN 5 to 7 seconds:

```
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+. To configure the forward delay time per VLAN, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# spanning-tree vlan` *vlan-range* `forward-time` *value*	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.

The following example shows how to configure the forward delay time for VLAN 5 to 21 seconds:

```
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+. To configure the maximum age time for a VLAN in Rapid PVST+, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# spanning-tree vlan` *vlan-range* `max-age` *value*	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.

The following example shows how to configure the maximum aging time for VLAN 5 to 36 seconds:

```
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.

To specify the link type, perform this task:

	Command	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.

The following 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.

To restart the protocol negotiation, perform this task:

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 the Ethernet interface on slot 1, port 8:

```
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.

The following example shows how to display spanning tree status:

```
switch# show spanning-tree brief

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     0005.ad00.31d6
             This bridge is the root
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0005.ad00.31d6
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Eth1/9           Desg FWD 2         128.137  P2p
Eth1/13          Desg FWD 2         128.141  P2p
Eth1/17          Desg FWD 2         128.145  P2p


VLAN0010
  Spanning tree enabled protocol rstp
  Root ID    Priority    32768
```

```
              Address     000d.ecb2.2cbc
              Cost        2
              Port        129 (Ethernet1/1)
              Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32778  (priority 32768 sys-id-ext 10)
              Address     0005.ad00.31d6
              Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Eth1/1           Root FWD 2         128.129  P2p
Eth1/3           Desg FWD 2         128.131  P2p
```

CHAPTER 9

Configuring MST

Multiple Spanning Tree (MST), which is the IEEE 802.1s standard, allows you to assign two or more VLANs to a spanning tree instance. MST is not the default spanning tree mode; Rapid per VLAN Spanning Tree (Rapid PVST+) is the default mode. MST instances with the same name, revision number, and VLAN-to-instance mapping combine to form an MST region. The MST region appears as a single bridge to spanning tree configurations outside the region. MST fails over to IEEE 802.1D Spanning Tree Protocol (STP) when it receives an 802.1D message from a neighboring switch.

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.

This chapter includes the following sections:

Note See Chapter 8, "Configuring Rapid PVST+" for complete information on STP and Rapid PVST+ and Chapter 10, "Configuring STP Extensions" for complete information on STP extensions.

Information About MST

This section includes the following topics:

MST Overview

Note You must enable MST; Rapid PVST+ is the default spanning tree mode.

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 improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

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. (See Chapter 8, "Configuring Rapid PVST+" for complete information on the explicit handshake agreement.)

MAC address reduction is always enabled while you are using MST. (See Chapter 8, "Configuring Rapid PVST+" for complete information on MAC address reduction.) 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+)

Note
- 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.

MST Regions

To allow switches to participate in MST instances, you must consistently configure the switches with the same MST configuration information (see the "MST Configuration Information" section on page 9-3).

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. (See the "IST, CIST, and CST" section on page 9-4 for more information on the IST.)

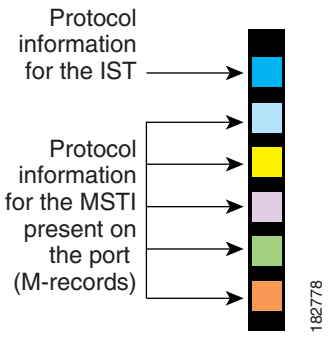
The MST region appears as a single bridge to adjacent MST regions and to other Rapid PVST+ regions and 802.1D spanning tree protocols.

Note We do not recommend that you 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 Figure 9-1). Only the IST sends BPDUs for the MST region; all M-records are encapsulated in that one BPDU that the IST sends (see the "IST, CIST, and CST Overview" section on page 9-4 for more information on IST). 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 9-1 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

These sections describe internal spanning tree (IST), common and internal spanning tree (CIST), and common spanning tree (CST):

IST, CIST, and CST Overview

Unlike Rapid PVST+ (see Chapter 8, "Configuring Rapid PVST+" for more information on this subject), 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.

For more information, see the "Spanning Tree Operation Within an MST Region" section on page 9-5 and the "Spanning Tree Operations Between MST Regions" section on page 9-5.

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 as shown in Figure 9-2. 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.

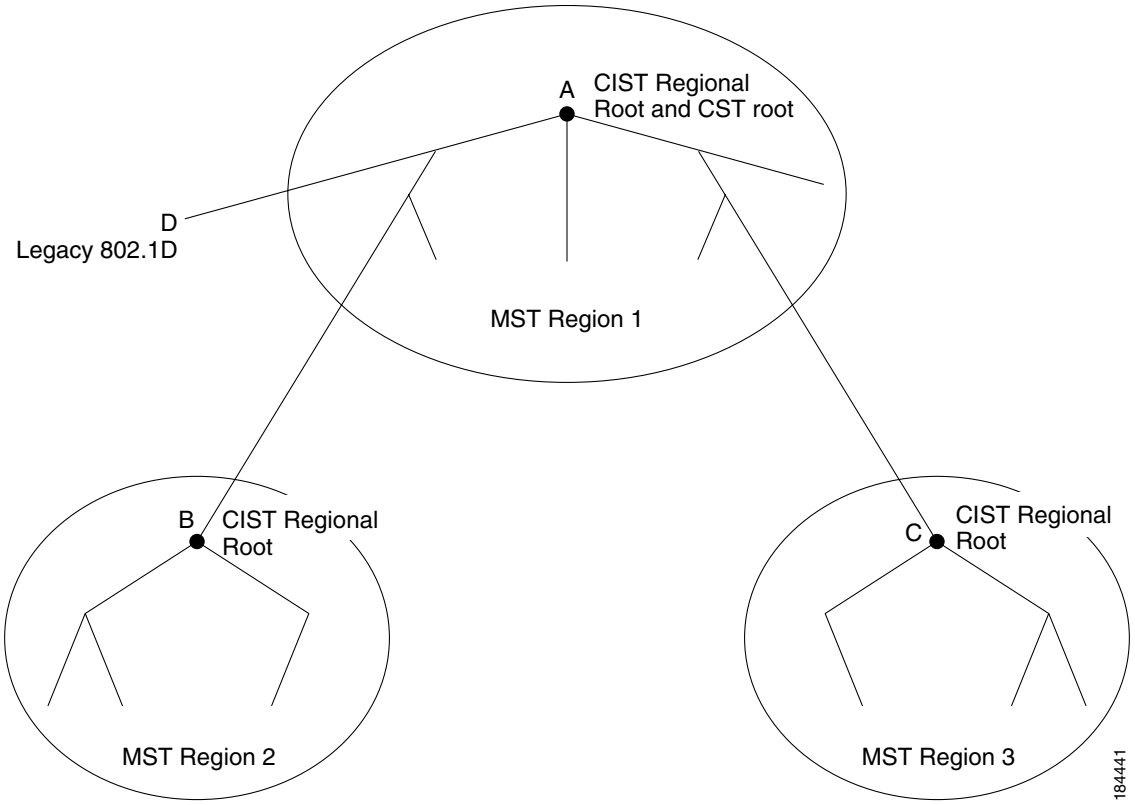
Spanning Tree Operations Between MST Regions

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.

Figure 9-2 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 9-2 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.

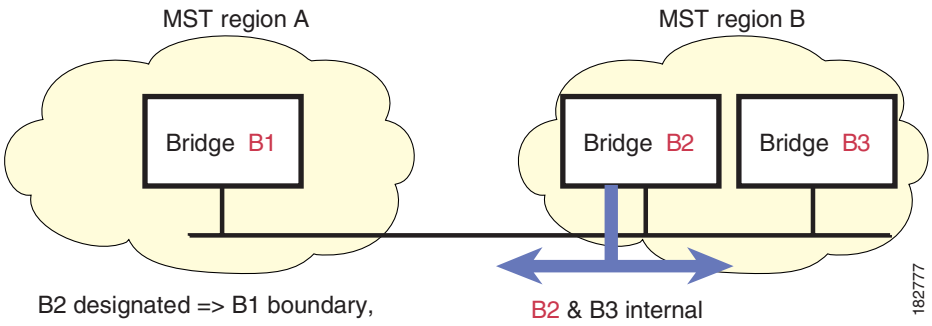
The message-age and maximum-age information in the 802.1w portion of the BPDU remain the same throughout the region (only on the IST), and the same values are propagated by the region-designated ports at the boundary.

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 to a LAN, the designated bridge of which is either a bridge with a different MST configuration (and so, a separate MST region) or a Rapid PVST+ or 802.1D STP bridge. 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 Figure 9-3).

Figure 9-3 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.

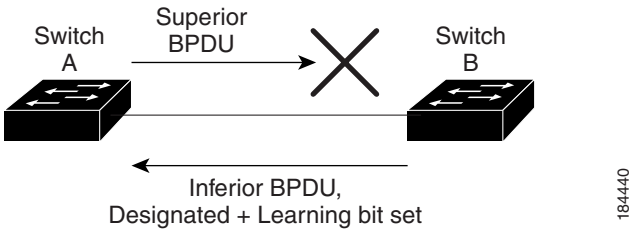
Detecting Unidirectional Link Failure

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.

Figure 9-4 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 9-4 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

This section includes the following topics:

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 submode, 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 submode without committing any changes, enter the **abort** command.
  - To leave the MST configuration submode and commit all the changes that you made before you left the submode, enter the **exit** command.

Enabling MST

You must enable MST; Rapid PVST+ is the default.

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.

To enable MST on the switch, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# spanning-tree mode mst`	Enables MST on the switch.

The following example shows how to enable MST on the switch:

```
switch# configure terminal
switch(config)# spanning-tree mode mst
```

To disable MST on the switch, perform this task:

Command	Purpose
`switch(config)# no spanning-tree mode mst`	Disables MST on the switch and returns you to Rapid PVST+.

Caution Changing the spanning tree mode can disrupt traffic because all spanning tree instances are stopped for the previous mode and restarted in the new mode.

Note Because STP is enabled by default, entering a **show running** 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.

To enter MST configuration mode, perform this task (note the difference between **exit** and **abort**):

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# spanning-tree mst configuration`	Enters MST configuration submode on the system. You must be in the MST configuration submode 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`	Commits all the changes and exits MST configuration submode.
	`switch(config-mst)# abort`	Exits the MST configuration submode without committing any of the changes.

The following example shows how to enter MST configuration submode on the switch:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
```

The following example shows how to commit the changes and leave MST configuration submode on the switch:

```
sswitch(config-mst)# exit
```

The following example shows how to leave MST-submode configuration on the switch without committing the changes:

```
sswitch(config-mst)# abort
```

To disable MST configuration mode, perform this task:

Command	Purpose
`switch(config-mst)# no spanning-tree mst configuration`	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.

To specify an MST name, perform this task:

	Command	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.

The following 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.

To specify an MST revision number, perform this task:

	Command	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.

The following 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.

To specify the configuration on an MST region, perform this task:

	Command	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.
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** global 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.

The following 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.

To map VLANs to MST instances, perform this task:

	Command	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.

The following 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
```

To unmap VLAN to MST instances, perform this task:

Command	Purpose
`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.

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.

To accomplish this synchronization automatically, perform this task:

	Command	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.

The following 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.

To enable the root bridge configuration, perform this task:

	Command	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.

The following 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
```

To disable the root bridge configuration, perform this task:

Command	Purpose
`switch(config)# no spanning-tree mst` *instance-id* **root**	Returns the switch priority, diameter, and hello time to default values.

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** global configuration command.

To enable a secondary root bridge, perform this task:

	Command	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 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.

The following 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
```

To disable the secondary root bridge configuration, perform this task:

Command	Purpose
`switch(config)# no spanning-tree mst` *instance-id* **root**	Returns the switch priority, diameter, and hello-time to default values.

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.

To configure the port priority, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.

	Command	Purpose
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: • 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.

The following example shows how to set the MST interface port priority for MSTI 3 on Ethernet port 1/1 to 64:

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

To configure the port cost, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.

	Command	Purpose
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. 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.

The following example shows how to set the MST interface port cost on Ethernet 1/1 for MSTI 4:

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

To configure the switch priority for an MST instance, perform this task:

	Command	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. • 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.

The following 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** global configuration commands to modify the hello time.

To configure the hello time, perform this task:

	Command	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.

The following 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.

To configure the forward delay timer, perform this task:

	Command	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.

The following 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).

To configure the maximum-aging timer, perform this task:

	Command	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.

The following 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).

To configure the maximum hop count, perform this task:

	Command	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.

The following 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.

To configure PVST simulation, perform this task:

	Command	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.

The following 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

Note PVST simulation is enabled by default; all interfaces on the switch interoperate between MST and Rapid PVST+.

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.

To disable PVST simulation, perform this task:

	Command	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.
	`switch(config-if)# spanning-tree mst simulate pvst`	Re-enables seamless operation between MST and Rapid PVST+ on specified interfaces.
	`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.

The following 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.

To specify the link type, perform this task:

	Command	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.

The following 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.

To restart the protocol, perform this task:

	Command	Purpose
Step 1	`switch# clear spanning-tree detected-protocol [interface interface [interface-num | port-channel]]`	Restarts MST on entire switch or specified interfaces.

The following 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 10

Configuring STP Extensions

Cisco has added extensions to the Spanning Tree Protocol (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.

This chapter includes the following sections:

Note See Chapter 8, "Configuring Rapid PVST+" for complete information on STP and Rapid PVST+ and Chapter 9, "Configuring MST" for complete information on MST.

Information About STP Extensions

This section includes the following topics:

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.

This section includes the following topics:

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 set as an edge port, you might create a bridging loop.

Spanning Tree Network Ports

Network ports are connected only to switches or bridges. Bridge Assurance is enabled only on network ports.

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 normal ports.

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. Table 10-1 lists all the BPDU Filtering combinations.

Table 10-1 BPDU Filtering Configurations

BPDU Filtering Per Port Configuration	BPDU Filtering Global Configuration	STP Edge Port Configuration	BPDU Filtering State
Default	Enable	Enable	Enable[1]
Default	Enable	Disable	Disable
Default	Disable	Not applicable	Disable
Disable	Not applicable	Not applicable	Disable
Enable	Not applicable	Not applicable	Enable

1. The 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.

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. (See Chapter 8, “Configuring Rapid PVST+” for information on STP port states.)

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

This section includes the following topics:

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 configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly for the type of device to which the interface is connected.

To configure the spanning tree port types globally, perform this task:

	Command	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.
	`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.

The following 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
```

The following 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 configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that the interface is connected to hosts.

To configure spanning tree edge ports on a specified interface, perform this task:

	Command	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.

The following 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 configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that the interface is connected to switches or routers.

To configure spanning tree network ports on a specified interface, perform this task:

	Command	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.

The following 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 configure this feature, you should do the following:

- Ensure that STP is configured.

- Ensure that you have configured some spanning tree edge ports.

To enable BPDU Guard globally, perform this task:

	Command	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.

The following 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 configure this feature, ensure that STP is configured.

To enable BPDU Guard on an interface, perform this task:

	Command	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.

The following 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
```

To disable BPDU Guard on an interface, perform this task:

Command	Purpose
`switch(config-if)# no spanning-tree bpduguard`	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.

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 resumes the regular STP transitions. However, this port maintains its configuration as an edge port.

Caution Be careful when using this command. Using this command incorrectly can cause bridging loops.

Before you configure this feature, you should do the following:

- Ensure that STP is configured.
- Ensure that you have configured some spanning tree edge ports.

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.

To enable BPDU Filtering globally, perform this task:

	Command	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.

The following 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.

Before you configure this feature, ensure that STP is configured.

Note When you enable BPDU Filtering locally on a port, this feature prevents the device from receiving or sending BPDUs on this port.

To enable BPDU Filtering on an interface, perform this task:

	Command	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.

The following 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
```

To disable BPDU Filtering on an interface, perform this task:

Command	Purpose
`switch(config-if)# no spanning-tree bpdufilter`	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.

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 configure this feature, you should do the following:

- Ensure that STP is configured.
- Ensure that you have spanning tree normal ports or have configured some network ports.

To enable Loop Guard globally, perform this task:

	Command	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.

The following 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

Note You can run Loop Guard on spanning tree normal or network ports. You can run Root Guard on all spanning tree ports: normal, edge, or network.

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 configure this feature, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring Loop Guard on spanning tree normal or network ports.

To enable Loop Guard or Root Guard on an interface, perform this task:

	Command	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.

The following 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 11

Configuring EtherChannels

This chapter describes how to configure EtherChannels and to apply and configure the Link Aggregation Control Protocol (LACP) for more efficient use of EtherChannels in Cisco NX-OS software.

This chapter includes the following sections:

Information About EtherChannels

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

You create an EtherChannel by bundling compatible interfaces. You can configure and run either static EtherChannels or EtherChannels running the Link Aggregation Control Protocol (LACP). (See "Understanding LACP" section on page 11-4 for information on LACP.)

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

You can use static EtherChannels, with no associated protocol, for a simplified configuration. For more efficient use of the EtherChannel, 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.

This section includes the following topics:

Understanding EtherChannels

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

You can collect up to eight ports into a static EtherChannel or you can enable the Link Aggregation Control Protocol (LACP). Configuring EtherChannels with LACP requires slightly different steps than configuring static EtherChannels (see the "Configuring EtherChannels" section on page 11-7).

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

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

Each port can be in only one EtherChannel. All the ports in an EtherChannel must be compatible; they must use the same speed and operate in full-duplex mode (see the "Compatibility Requirements" section on page 11-2). When you are running static EtherChannels, without LACP, the individual links are all in the on channel mode; you cannot change this mode without enabling LACP (see the "Port-Channel Modes" section on page 11-6).

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

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

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

Compatibility Requirements

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

The compatibility check includes the following operational attributes:

- Port mode
- Access VLAN
- Trunk native VLAN
- Allowed VLAN list
- Speed

- 802.3x flow control setting
- MTU

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

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

You can only add interfaces configured with the channel mode set to **on** to static EtherChannels. You can also only add interfaces configured with the channel mode as **active** or **passive** to EtherChannels that are running LACP. (See the "Port-Channel Modes" section on page 11-6 for information on port-channel modes.) You can configure these attributes on an individual member port.

When the interface joins an EtherChannel, the following individual parameters are replaced with the values on the EtherChannel:

- Bandwidth
- MAC address
- Spanning Tree Protocol

The following interface parameters remain unaffected when the interface joins an EtherChannel:

- Description
- CDP
- LACP port priority
- Debounce

Load Balancing Using EtherChannels

Cisco NX-OS load balances traffic across all operational interfaces in an EtherChannel 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. EtherChannels 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, the source and destination IP addresses, and the source and destination port number.

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

- 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 11-1 shows the criteria used for each configuration:

Table 11-1 EtherChannel Load-Balancing Criteria

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

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

Understanding LACP

LACP allows you to configure up to eight interfaces into an EtherChannel.

This section includes the following topics:

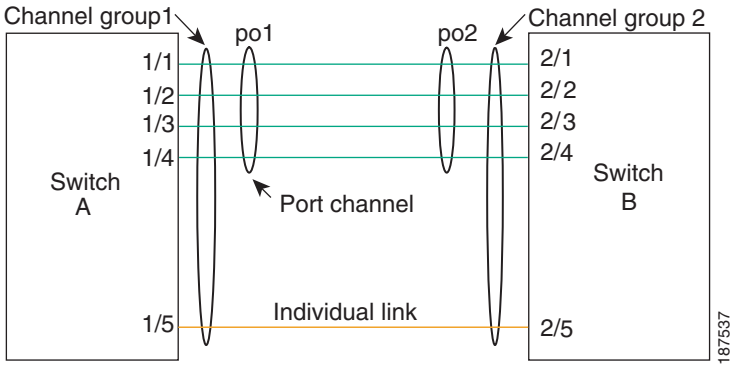
- LACP-Enabled and Static EtherChannels Differences, page 11-7

LACP Overview

Note You must enable LACP before the feature functions.

Figure 11-1 shows how individual links can be combined into LACP EtherChannels and channel groups as well as function as individual links.

Figure 11-1 Individual Links Combined into an EtherChannel



With LACP, you can bundle up to eight interfaces in a channel group.

Note When you delete the EtherChannel, 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. The ability of a port 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

Port-Channel Modes

Individual interfaces in EtherChannels are configured with channel modes. When you run static EtherChannels, 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.

Table 11-2 describes the channel modes.

Table 11-2 Channel Modes for Individual Links in an EtherChannel

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 EtherChannels, 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 an EtherChannel, 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 EtherChannel 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 an EtherChannel successfully with another port that is in active mode.
- A port in active mode can form an EtherChannel with another port in passive mode.

- A port in passive mode cannot form an EtherChannel 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 EtherChannels, 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 EtherChannels Differences

Table 11-3 provides a brief summary of major differences between EtherChannels with LACP enabled and static EtherChannels.

Table 11-3 EtherChannels with LACP Enabled and Static EtherChannels

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	6	6

Configuring EtherChannels

You can configure multiple EtherChannels on a device.

This section includes the following topics:

Creating an EtherChannel

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

Note If you want LACP-based EtherChannels, you need to enable LACP (see the "Enabling LACP" section on page 11-10).

To create an EtherChannel, perform this task:

	Command	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.

The following example shows how to create an EtherChannel:

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

To remove the EtherChannel and delete the associated channel group, perform this task:

Command	Purpose
`switch(config)# no interface port-channel` *channel-number*	Removes the EtherChannel and deletes the associated channel group. See the "Compatibility Requirements" section on page 11-2 for details on how the interface configuration changes when you delete the EtherChannel.

Adding a Port to an EtherChannel

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

Note If you want LACP-based EtherChannels, you need to enable LACP (see the "Enabling LACP" section on page 11-10).

To configure an EtherChannel, perform this task:

	Command	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.

	Command	Purpose
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 EtherChannel associated with this channel group if the EtherChannel does not already exist[1].

1. This is called implicit EtherChannel creation.

To remove the port from the channel group, perform this task:

Command	Purpose
`switch(config)# no channel-group`	Removes the port from the channel group. The port reverts to its original configuration.

The following 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 EtherChannels

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

Note If you want LACP-based EtherChannels, you need to enable LACP (see the "Enabling LACP" section on page 11-10).

To configure load balancing using EtherChannels, perform this task:

	Command	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}`	Specifies the load-balancing algorithm for the device. The range depends on the device. The default is source-dest-mac.
Step 3	`switch(config-router)# show port-channel load-balance`	(Optional) Displays the port-channel load-balancing algorithm.

The following example shows how to configure source IP load balancing for EtherChannels:

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

To restore the default load-balancing algorithm of source-dest-mac for non-IP traffic and source-dest-ip for IP traffic, perform this task:

Command	Purpose
`switch(config)# no port-channel load-balance ethernet`	Restores the default load-balancing algorithm.

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 EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

To enable LACP, perform this task:

	Command	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 system internal clis feature`	(Optional) Displays enabled features.

The following example shows how to enable LACP:

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

Configuring Port-Channel Port Modes

After you enable LACP, you can configure the channel mode for each individual link in the LACP EtherChannel as **active** or **passive**. This channel configuration mode allows the link to operate with LACP.

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

To configure the LACP link mode, perform this task:

	Command	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.

	Command	Purpose
Step 3	`switch(config-if)# channel-group` *number* **mode** {**active** | **on** | **passive**}	Specifies the port mode for the link in an EtherChannel. After LACP is enabled, you configure each link or the entire channel as active or passive. When you run EtherChannels with no associated protocol, the port-channel mode is always on. The default port-channel mode is on.
	`switch(config-if)# no channel-group` *number* **mode**	Returns the port mode to on for the specified interface.

The following 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
```

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.

To configure the LACP system priority, perform this task:

	Command	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(config-if)# show lacp system-identifier`	Displays the LACP system identifier.

The following 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

When you enable LACP, you can configure each link in the LACP EtherChannel for the port priority.

To configure the LACP link mode and port priority, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.

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

The following 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 system internal clis feature`	Displays enabled features.
`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 an EtherChannel.
`switch# show port-channel database [interface port-channel channel-number]`	Displays the aggregation state for one or more port-channel interfaces.
`switch# show port-channel load-balance`	Displays the type of load balancing in use for EtherChannels.
`switch# show port-channel summary`	Displays a summary for the port-channel interfaces.
`switch# show port-channel traffic`	Displays the traffic statistics for EtherChannels.
`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 EtherChannel feature.

CHAPTER 12

Configuring Access and Trunk Interfaces

Ethernet interfaces can be configured either as access ports or trunk ports. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across the network.

This chapter includes the following sections:

Information About Access and Trunk Interfaces

This section includes the following topics:

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

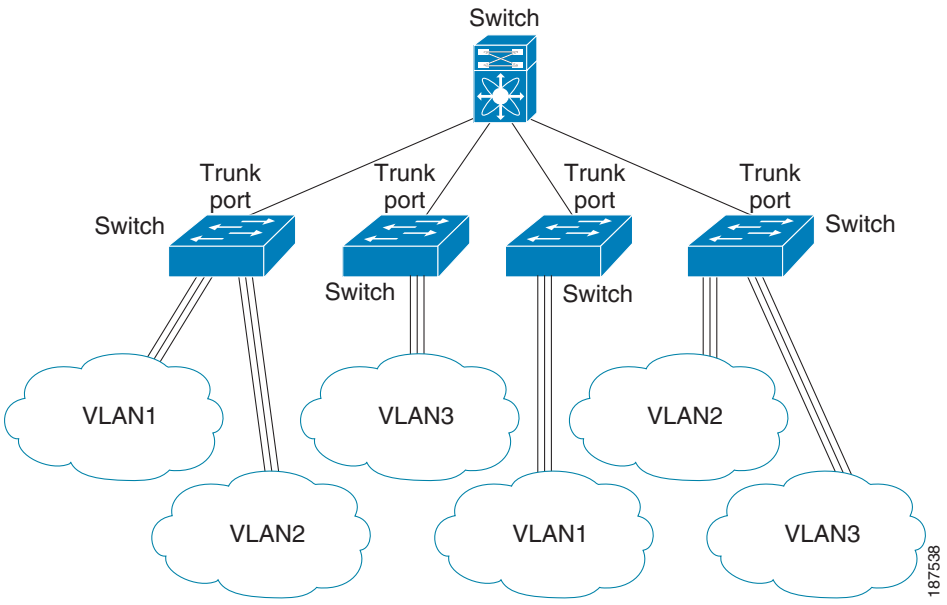
Understanding Access and Trunk Interfaces

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.

Figure 12-1 show how you can use trunk ports in the network. The trunk port carries traffic for two or more VLANs.

Figure 12-1 Devices in a Trunking Environment



To correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation or tagging method (see the "Understanding IEEE 802.1Q Encapsulation" section on page 12-2 for more information on this subject).

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.

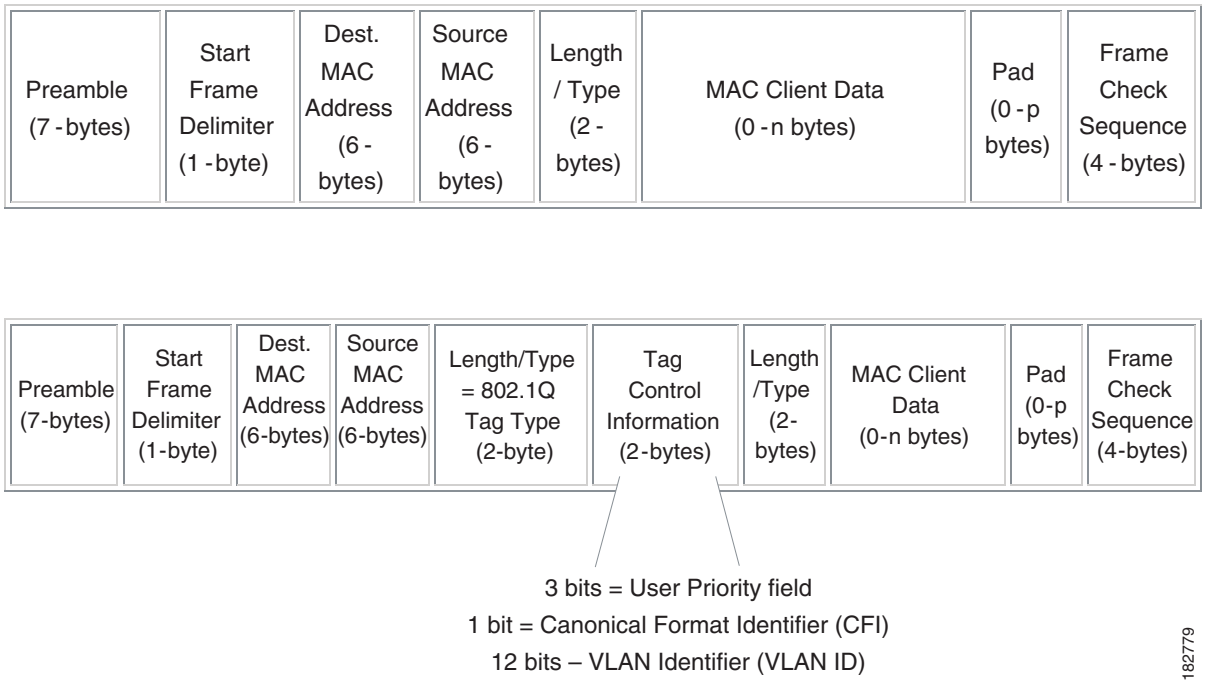
Note An Ethernet interface can function as either an access port or a trunk port; it cannot function as both port types simultaneously.

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 that uses a tag that is inserted into the frame header (see Figure 12-2). 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 12-2 Header Without and with 802.1Q Tag Included



Understanding Access VLANs

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.

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.

Understanding the Native VLAN ID for Trunk Ports

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

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.

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.

Configuring Access and Trunk Interfaces

This section includes the following topics:

Configuring a LAN Interface as an Ethernet Access Port

You can configure an Ethernet port 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.

To configure an Ethernet access port, perform this task:

	Command	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.

The following example shows how to set Ethernet 1/10 as an Ethernet access port that carries traffic for VLAN 5 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

Note You should apply the **switchport host** command only to interfaces connected to an end station.

You can optimize performance on access ports that are connected to end stations by simultaneously setting that port as an access port. An access host port handles the Spanning Tree Protocol (STP) like an edge port and immediately moves to the forwarding state without passing through the blocking and learning states. Configuring an interface as an access host port also disables port channeling on that interface.

Note See Chapter 11, "Configuring EtherChannels" for information on port channel interfaces and Chapter 8, "Configuring Rapid PVST+" for complete information on the Spanning Tree Protocol.

Ensure that you are configuring the correct interface to an interface that is an end station.

To configure an access host port, perform this task:

	Command	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 be an access host port, which immediately moves to the spanning tree forwarding state and disables port channeling on this interface. **Note** Apply this command only to end stations.

The following example shows how to set Ethernet 1/10 as an Ethernet access port with **PortFast enabled** and port channel 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. (See the "Understanding IEEE 802.1Q Encapsulation" section on page 12-2 for information about encapsulation.)

Note Cisco NX-OS supports only 802.1Q encapsulation.

To configure a trunk port, perform this task:

	Command	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.

The following example shows how to set Ethernet 1/1 as an Ethernet trunk port:

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

To configure native VLAN for a 802.1Q trunk port, perform this task:

	Command	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.

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

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

To configure the allowed VLAN for a trunk port, perform this task:

	Command	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. **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.

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

```
switch# configure terminal
```

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

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 13

Configuring the MAC Address Table

All Ethernet switching ports maintain media access control (MAC) address tables.

This chapter includes the following sections:

Information About MAC Addresses

To switch frames between LAN ports efficiently, the switch maintains an address table. When the switch receives a frame, it associates the media access control (MAC) 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 (for details, see the "Configuration Limits" section on page 36-1). 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

This section includes the following topics:

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.

To configure a static MAC address, perform this task:

	Command	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.

The following 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
```

To delete a static MAC address, perform this task:

Command	Purpose
`switch(config-if)# no mac-address-table static` *mac_address* `vlan` *vlan-id*	Deletes the static entry from the MAC address table.

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.

Note You can also configure MAC aging time in interface configuration mode or VLAN configuration mode.

To configure the aging time for all MAC addresses, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters 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 range is from 0 to 500; the default is 300 seconds. Entering the value 0 disables the MAC aging. If a VLAN is not specified, the aging specification applies to all VLANs.

The following example shows how to set the aging time for entries in the MAC address table to 400 seconds:

```
switch# configure terminal
switch(config)# mac-address-table aging-time 400
```

Clearing Dynamic Addresses from the MAC Table

You can clear all dynamic entries in the MAC address table.

To clear the MAC address table, perform this task:

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.

The following 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.
`switch# show mac-address-table`	Displays the contents of the MAC address table.

The following 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
```

The following 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

Internet Group Management Protocol (IGMP) snooping streamlines multicast traffic handling for VLANs. By examining (snooping) IGMP membership report messages from interested hosts, multicast traffic is limited to the subset of VLAN interfaces on which the hosts reside.

This chapter includes the following sections:

Information About IGMP Snooping

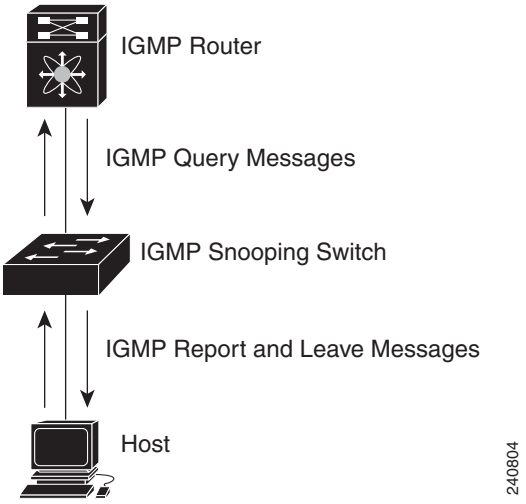
The IGMP snooping software examines IGMP protocol messages within a VLAN to discover which interfaces are connected to hosts or other devices interested in receiving this traffic. Using the interface information, IGMP snooping can reduce bandwidth consumption in a multi-access LAN environment to avoid 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.

Figure 14-1 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 14-1 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 RFC 4541.

This section includes the following topics:

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 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 Table 14-1.

Table 14-1 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.

To configure IGMP snooping, perform this task:

	Command	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.

	Command	Purpose
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.
	`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.
	`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.
	`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.
	`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.
	`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.
	`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.
	`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 VLAN 5:

```
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
switch#
```

You can disable IGMP snooping either globally or for a specific VLAN. To disable IGMP snooping globally, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# no ip igmp snooping`	Globally disables 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)# no ip igmp snooping`	Disables IGMP snooping for the current VLAN. The default is enabled.

The following example shows disabling IGMP snooping for VLAN 5 only:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# no ip igmp snooping
```

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]`	IGMP snooping configuration by VLAN.
`switch# show ip igmp snooping groups [[vlan] vlan-id] [detail]`	IGMP snooping information about groups by VLAN.
`switch# show ip igmp snooping querier [[vlan] vlan-id]`	IGMP snooping queriers by VLAN.
`switch# show ip igmp snooping mrouter [[vlan] vlan-id]`	Multicast router ports by VLAN.
`switch# show ip igmp snooping explicit-tracking vlan vlan-id`	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 describes how to configure traffic storm control on the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

This chapter includes the following sections:

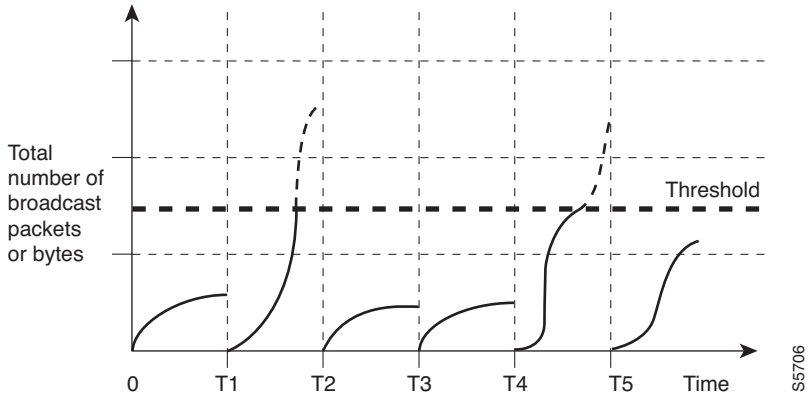
Information About Traffic Storm Control

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 Layer 2 ports by a broadcast, multicast, or unknown unicast traffic storm on physical interfaces.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming broadcast, multicast, and unknown unicast traffic over a 25-millisecond 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.

Figure 15-1 shows the broadcast traffic patterns on a Layer 2 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 15-1 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 switch is implemented in the hardware. The traffic storm control circuitry monitors packets that pass from a Layer 2 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 25-millisecond 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 specified 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 specified 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 specified 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 specified 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 specified interval, traffic storm control drops all multicast traffic until the end of the interval.
- If you set a unicast traffic storm control level it takes precedence over any other levels set for the broadcast and multicast traffic storm control levels.

By default, Cisco NX-OS takes no corrective action when the traffic exceeds the configured level.

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 1 to 100.
- The optional fraction of a level can be from 0 to 99.
- 100 percent means no traffic storm control.
- 1.0 percent means the port suppression level is 1% of the total multicast packets sent out from a port.

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.

To enable traffic storm control on an interface, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 1	`switch(config)# interface {ethernet slot/port | port-channel number}`	Enters interface configuration mode.
Step 2	`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.

The following 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.
`switch# show running-config interface`	Displays the traffic storm control configuration.

Displaying Traffic Storm Control Counters

You can display the counters the switch maintains for traffic storm control activity.

Note Traffic storm control uses a 10-microsecond interval that can affect the operation of traffic storm control.

To display traffic storm control counters on an interface, perform this task:

	Command	Purpose
Step 1	`switch# show interface [ethernet slot/port | port-channel number] counters storm-control`	Displays the traffic storm control counters.

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 40.5
switch(config-if)# storm-control unicast level 40
```

Default Settings

Table 15-1 lists the default settings for traffic storm control parameters.

Table 15-1 Default Traffic Storm Control Parameters

Parameters	Default
Traffic storm control	Disabled
Threshold percentage	100

CHAPTER 16

Configuring Link-State Tracking

This chapter describes how to configure link-state tracking and includes the following sections:

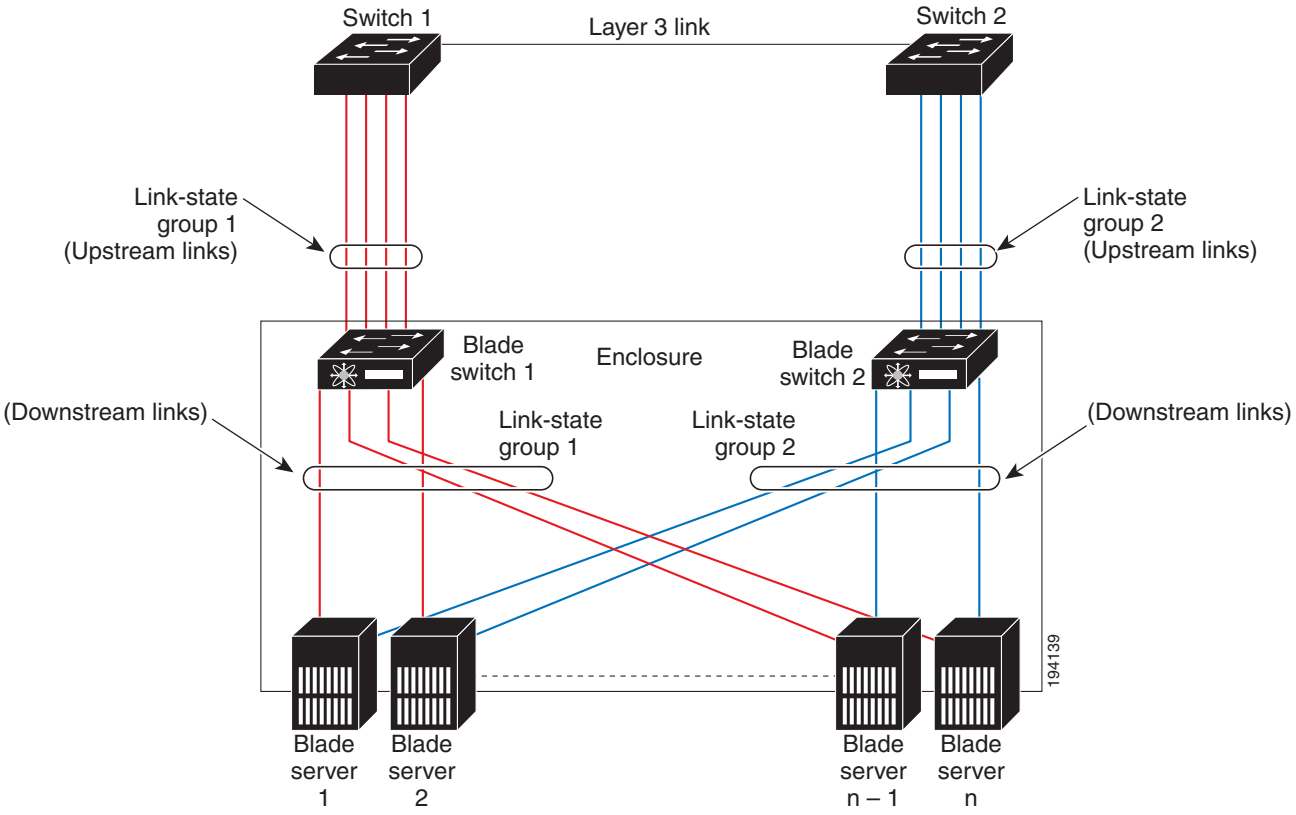
Understanding Link-State Tracking

Link-state tracking, also known as trunk failover, is a feature that binds the link state of multiple interfaces. Link-state tracking provides redundancy in the network when used with server network interface card (NIC) adapter teaming. When the server network adapters are configured in a primary or secondary relationship known as teaming and the link is lost on the primary interface, connectivity transparently changes to the secondary interface.

Figure 16-1 shows a network configured with link-state tracking. To enable link-state tracking, create a link-state group, and specify the interfaces that are assigned to the link-state group.

Interfaces connected to servers are referred to as downstream interfaces, and interfaces connected to distribution switches and network devices are referred to as upstream interfaces. When link-state tracking is enabled, the downstream interfaces are bound to the upstream interfaces. After a set of downstream ports are associated to a set of upstream ports, if all of the upstream ports become unavailable, link-state tracking automatically puts the associated downstream ports in an error-disabled state. This causes the primary interface of the server to failover to the secondary interface.

Figure 16-1 Typical Link-State Tracking Configuration



The configuration in Figure 16-1 ensures that when server NIC adapter teaming is used, the traffic flow continues uninterrupted when the uplink connection to a distribution switch is lost.

- The blade switches in the enclosure are connected to switch 1 and switch 2 through different switches.
- Link-state group 1 is the primary link from all the blade servers in the enclosure (blade server 1 through blade server n) to switch 1.
- Link-state group 2 is the secondary (backup) link from all the blade servers to switch 2.

In a link-state group, the upstream ports can become unavailable or lose connectivity because the switch or router fails, the cables are disconnected, or the link is lost. These are the interactions between the downstream and upstream interfaces when link-state tracking is enabled:

- If any of the upstream interfaces are in the link-up state, the downstream interfaces can change to or remain in the link-up state.
- If the server detects that the primary link is down, it redirects the traffic to the secondary (backup) link and the secondary link becomes the primary link.

  As an example of a connectivity change from link-state group 1 to link-state group 2, when the primary link from blade switch 1 to switch 1 is lost, blade server 1 connects through its secondary Ethernet server interface to blade switch 2 in link-state group 2.
- When link-state tracking is disabled, the entire feature is disabled. All the configuration for link-state tracking is removed. The downstream ports are reverted back to the state when no link-state tracking was set.

You can recover a downstream interface link-down condition by removing the failed downstream port from the link-state group. To recover for multiple downstream interfaces, disable the link-state group.

Configuring Link-State Tracking

The following section describes how to configure link-state tracking ports and includes the following topics:

Default Link-State Tracking Configuration

The link-state tracking feature is disabled. No link-state groups are defined.

Link-State Tracking Configuration Guidelines

Follow the listed guidelines when configuring link-state tracking:

- An interface that is defined as an upstream interface cannot also be defined as a downstream interface in the same or a different link-state group. The reverse is also true.
- An interface cannot be a member of more than one link-state group.
- Management interfaces cannot be members of a link-state group.
- An EtherChannel can be configured as upstream member in a link-state group. However, do not configure it as a downstream member.
- Only interfaces ethernet 1/1 through 1/6 can be configured as upstream ports in a specific link-state group.
- Only interfaces ethernet 1/7 through 1/20 can be configured as downstream ports in a specific link-state group.

Configuring Link-State Tracking

Beginning in privileged EXEC mode, and to configure a link-state group and to assign an interface to a group, perform this task:

	Command	Purpose
Step 1	`configure terminal`	Enters global configuration mode.
Step 1	`feature lst`	Enables link-state track feature.
Step 2	`link state track` *number*	Creates a link-state group, and enable link-state tracking. The group number can be 1 to 6. The default is 1.

	Command	Purpose
Step 3	**interface** *interface-id*	Specifies a physical interface or range of interfaces to configure, and enters interface configuration mode. Valid interfaces include switch ports in access or trunk mode (IEEE 802.1q), routed ports, or multiple ports bundled into an EtherChannel interface (static or LACP), that is also in trunk mode.
Step 4	**link state group** [*number*] {**upstream** | **downstream**}	Specifies a link-state group, and configures the interface as either an upstream or downstream interface in the group. The group number can be 1 to 6. The default is 1.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The following example shows how to create a link-state group and to configure the interface:

```
switch# configure terminal
switch(config)# feature lst
switch(config)# link state track 1
switch(config)# interface port-channel 1
switch(config-if)# link state group 1 upstream
switch(config-if)# end
```

The following example shows how to remove an interface from a link-state group:

```
switch# configure terminal
switch(config)# interface port-channel 1
switch(config-if)# no link state group 1 upstream
switch(config-if)# end
```

To disable the link state track feature, use the **no feature lst** command.

The link-state group cannot be disabled.

Displaying Link-State Tracking Status

Use the **show link state group** command to display the link-state group information. Enter this command without keywords to display information about all link-state groups. Enter the group number to display information specific to the group. Enter the detail keyword to display detailed information about the group.

The following is sample output from the **show link state group 1** command:

```
switch> show link state group 1
Link State Group: 1      Status: Enabled, Down
```

The following is sample output from the **show link state group detail** command:

```
switch> show link state group detail
(Up):Interface up   (Dwn):Interface Down   (Dis):Interface disabled
Link State Group: 1      Status: Enabled, Up
Upstream Interfaces   : Po1(Up)
Downstream Interfaces : Gi0/3(Up) Gi0/4(Up)
(Up):Interface up   (Dwn):Interface Down   (Dis):Interface disabled
```

PART 3

Switch Security Features

CHAPTER 17

Configuring AAA

This chapter describes how to configure authentication, authorization, and accounting (AAA) on the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

This chapter includes the following sections:

Information About AAA

This section includes the following topics:

AAA Security Services

The AAA feature allows you to verify the identity of, grant access to, and track the actions of users managing the switch. The switch supports Remote Access Dial-In User Service (RADIUS) or Terminal Access Controller Access Control System Plus (TACACS+) protocols.

Based on the user ID and password combination that you provide, the switch performs local authentication or authorization using the local database or remote authentication or authorization using one or more AAA servers. A preshared secret key provides security for communication between the switch and AAA servers. You can configure a common secret key for all AAA servers or for only a specific AAA server.

AAA security provides the following services:

- Authentication—Identifies users, including login and password dialog, challenge and response, messaging support, and, encryption depending on the security protocol that you select.

  Authentication is the process of verifying the identity of the person or device accessing the switch. This process is based on the user ID and password combination provided by the entity trying to access the switch. The switch allows you to perform local authentication (using the local lookup database) or remote authentication (using one or more RADIUS or TACACS+ servers).

- Authorization—Provides access control.

  AAA authorization is the process of assembling a set of attributes that describe what the user is authorized to perform. Authorization in switch is provided by attributes that are downloaded from AAA servers. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user.

- Accounting—Provides the method for collecting information, logging the information locally, and sending the information to the AAA server for billing, auditing, and reporting.

  The accounting feature tracks and maintains a log of every management session used to access the switch. You can use this information to generate reports for troubleshooting and auditing purposes. You can store accounting logs locally or send them to remote AAA servers.

Note The Cisco NX-OS software supports authentication, authorization, and accounting independently. For example, you can configure authentication and authorization without configuring accounting.

Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration
- Scalability
- Standardized authentication methods, such as RADIUS and TACACS+
- Multiple backup devices

Remote AAA Services

Remote AAA services provided through RADIUS and TACACS+ protocols have the following advantages over local AAA services:

- User password lists for each switch are easier to manage.
- AAA servers are already deployed widely across enterprises and can be easily used for AAA services.

- The accounting log for all switches in the fabric can be centrally managed.

AAA Server Groups

You can specify remote AAA servers for authentication, authorization, and accounting using server groups. A server group is a set of remote AAA servers that implement the same AAA protocol. The purpose of a server group is to provide for failover servers in case a remote AAA server fails to respond. If the first remote server in the group fails to respond, the next remote server in the group is tried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, then that server group option is considered a failure. If required, you can specify multiple server groups. If the switch encounters errors from the servers in the first group, it tries the servers in the next server group.

AAA Service Configuration Options

On the switch, you can have separate AAA configurations for the following services:

- User Telnet or Secure Shell (SSH) login authentication
- Console login authentication
- User management session accounting

Table 17-1 lists the CLI commands for each AAA service configuration option.

Table 17-1 AAA Service Configuration Commands

AAA Service Configuration Option	Related Command
Telnet or SSH login	**aaa authentication login default**
Console login	**aaa authentication login console**
User session accounting	**aaa accounting default**

You can specify the following authentication methods for the AAA services:

- RADIUS server groups—Uses the global pool of RADIUS servers for authentication.
- Specified server groups—Uses specified RADIUS or TACACS+ server groups for authentication.
- Local—Uses the local username or password database for authentication.
- None—Uses only the username.

Note If the method is for all RADIUS servers, instead of a specific server group, the switch chooses the RADIUS server from the global pool of configured RADIUS servers in the order of configuration. Servers from this global pool are the servers that can be selectively configured in a RADIUS server group on the switch.

Table 17-2 describes the AAA authentication methods that you can configure for the AAA services.

Table 17-2 AAA Authentication Methods for AAA Services

AAA Service	AAA Methods
Console login authentication	Server groups, local, and none
User login authentication	Server groups, local, and none
User management session accounting	Server groups and local

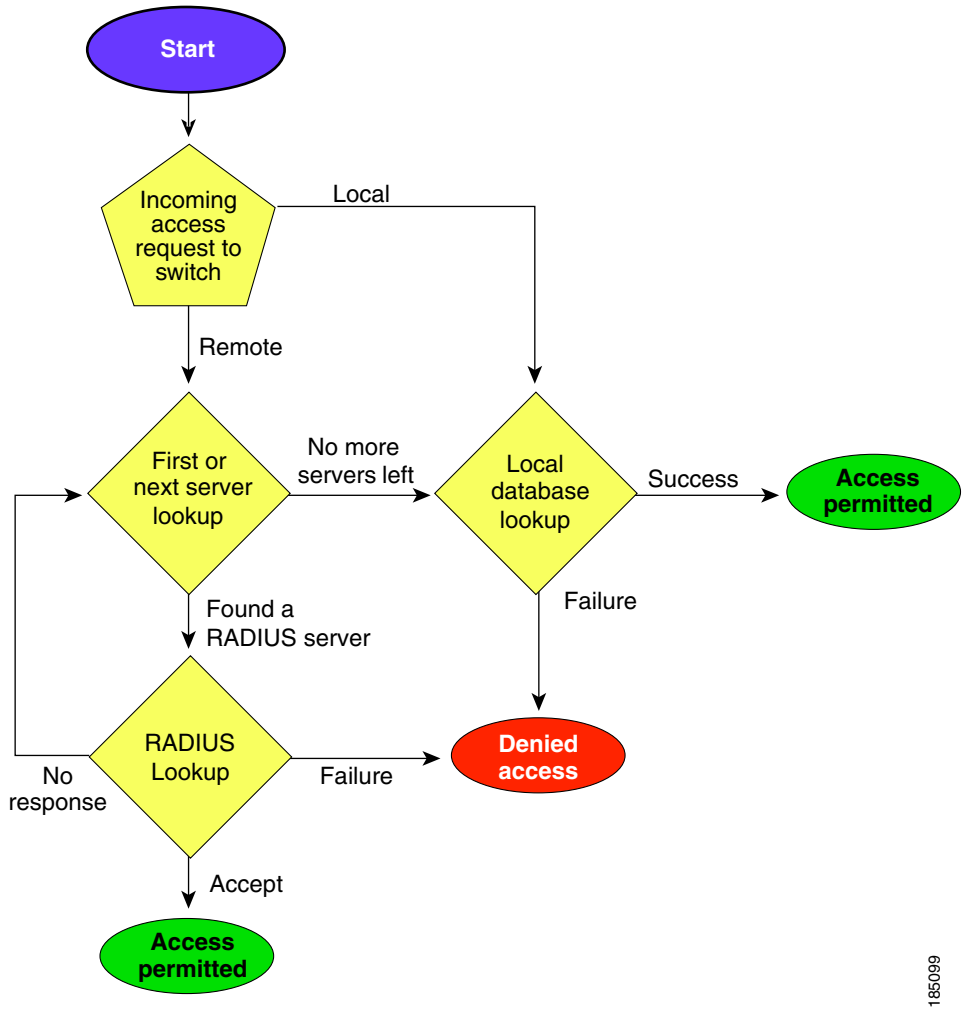
Note For console login authentication, user login authentication, and user management session accounting, the switch tries each option in the order specified. The local option is the default method when other configured options fail.

Authentication and Authorization Process for User Login

Figure 17-1 shows a flowchart of the authentication and authorization process for user login. The following process occurs:

1. When you log in to the required switch, you can use the Telnet, SSH, Fabric Manager or Device Manager, or console login options.
2. When you have configured the AAA server groups using the server group authentication method, the switch sends an authentication request to the first AAA server in the group as follows:
   a. If the AAA server fails to respond, then the next AAA server is tried and so on until the remote server responds to the authentication request.
   b. If all AAA servers in the server group fail to respond, then the servers in the next server group are tried.
   c. If all configured methods fail, then the local database is used for authentication.
3. If the switch successfully authenticate you through a remote AAA server, then the following possibilities apply:
   a. If the AAA server protocol is RADIUS, then user roles specified in the cisco-av-pair attribute are downloaded with an authentication response.
   b. If the AAA server protocol is TACACS+, then another request is sent to the same server to get the user roles specified as custom attributes for the shell.
4. If your username and password are successfully authenticated locally, the switch logs you in and assigns you the roles configured in the local database.

Figure 17-1 Authorization and Authentication Flow for User Login



Note "No more server groups left" means that there is no response from any server in all server groups. "No more servers left" means that there is no response from any server within this server group.

Prerequisites for Remote AAA

Remote AAA servers have the following prerequisites:

- At least one RADIUS or TACACS+ server must be IP reachable (see the "Configuring RADIUS Server Hosts" section on page 18-5 and the "Configuring TACACS+ Server Hosts" section on page 19-5).
- The switch is configured as a client of the AAA servers.
- The preshared secret key is configured on the switch and on the remote AAA servers.
- The remote server responds to AAA requests from the switch (see the "Manually Monitoring RADIUS Servers or Groups" section on page 18-13 and the "Manually Monitoring TACACS+ Servers or Groups" section on page 19-12).

AAA Guidelines and Limitations

The switch does not support all numeric usernames, whether created with TACACS+ or RADIUS, or created locally, and do not create local users with all numeric names. If an all numeric username exists on a AAA server and is entered during login, the switch will log in the user.

Configuring AAA

To configure AAA authentication and accounting, perform the following steps:

Step 1 If you want to use remote RADIUS or TACACS+ servers for authentication, configure the hosts on your switch. See Chapter 18, "Configuring RADIUS" and Chapter 19, "Configuring TACACS+."

Step 2 Configure console login authentication methods. See the "Configuring Console Login Authentication Methods" section on page 17-6.

Step 3 Configure default login authentication methods for user logins. See the "Configuring Default Login Authentication Methods" section on page 17-7

Step 4 Configure default AAA accounting default methods. See the "Configuring AAA Accounting Default Methods" section on page 17-9.

The following topics describe the AAA configuration procedure in more details:

Configuring Console Login Authentication Methods

This section describes how to configure the authentication methods for the console login.

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS or TACACS+ servers
- Local database on the switch
- Username only (**none**)

The default method is local.

Note The **group radius** and **group** *server-name* forms of the **aaa authentication** command are used for a set of previously defined RADIUS servers. Use the **radius server-host** command to configure the host servers. Use the **aaa group server radius** command to create a named group of servers.

Before you configure console login authentication methods, configure RADIUS or TACACS+ server groups as needed. To configure console login authentication methods, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# aaa authentication login console {group group-list [none]| local | none}`	Configures login authentication methods for the console. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—Uses the global pool of RADIUS servers for authentication. • *named-group*—Uses a named subset of TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default console login method is **local**, which is used when no methods are configured or when all of the configured methods fail to respond.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show aaa authentication`	(Optional) Displays the configuration of the console login authentication methods.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure authentication methods for the console login:

```
switch# configure terminal
switch(config)# aaa authentication login console group radius
switch(config)# exit
switch# show aaa authentication
switch# copy running-config startup-config
```

Configuring Default Login Authentication Methods

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS or TACACS+ servers
- Local database on the switch
- Username only

The default method is local.

Before you configure default login authentication methods, configure RADIUS or TACACS+ server groups as needed. To configure default login authentication methods, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# aaa authentication login default {group group-list [none]| local | none}`	Configures the default authentication methods. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—Uses the global pool of RADIUS servers for authentication. • *named-group*—Uses a named subset of TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default login method is **local**, which is used when no methods are configured or when all of the configured methods do not respond.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show aaa authentication`	(Optional) Displays the configuration of the default login authentication methods.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling Login Authentication Failure Messages

When you log in, the login is processed by the local user database if the remote AAA servers do not respond. If you have enabled the displaying of login failure messages, the following message is displayed:

```
Remote AAA servers unreachable; local authentication done.
Remote AAA servers unreachable; local authentication failed.
```

To enable login authentication failure messages, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# aaa authentication login error-enable`	Enables login authentication failure messages. The default is disabled.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show aaa authentication`	(Optional) Displays the login failure message configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling MS-CHAP Authentication

Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) is the Microsoft version of CHAP. You can use MS-CHAP for user logins to a switch through a remote authentication server (RADIUS or TACACS+).

By default, the switch uses Password Authentication Protocol (PAP) authentication between the switch and the remote server. If you enable MS-CHAP, you need to configure your RADIUS server to recognize the MS-CHAP vendor-specific attributes (VSAs). See the "Using AAA Server VSAs with the Switch" section on page 17-10. Table 17-3 describes the RADIUS VSAs required for MS-CHAP.

Table 17-3 MS-CHAP RADIUS VSAs

Vendor-ID Number	Vendor-Type Number	VSA	Description
311	11	MS-CHAP-Challenge	Contains the challenge sent by a AAA server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MS-CHAP-Response	Contains the response value provided by an MS-CHAP user in response to the challenge. It is only used in Access-Request packets.

To enable MS-CHAP authentication, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# aaa authentication login mschap enable`	Enables MS-CHAP authentication. The default is disabled.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show aaa authentication login mschap`	(Optional) Displays the MS-CHAP configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring AAA Accounting Default Methods

The switch supports TACACS+ and RADIUS methods for accounting. The switches report user activity to TACACS+ or RADIUS security servers in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the AAA server.

When you activate AAA accounting, the switch reports these attributes as accounting records, which are then stored in an accounting log on the security server.

You can create default method lists defining specific accounting methods, which include the following:

- RADIUS server group—Uses the global pool of RADIUS servers for accounting.
- Specified server group—Uses a specified RADIUS or TACACS+ server group for accounting.
- Local—Uses the local username or password database for accounting.

Note If you have configured server groups and the server groups do not respond, by default the local database is used for authentication.

Before you configure AAA accounting default methods, configure RADIUS or TACACS+ server groups as needed.

To configure AAA accounting default methods, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# aaa accounting default {group group-list | local}`	Configures default accounting method. One or more server group names can be specified in a space separated list. The *group-list* argument consists of a space-delimited list of group names. The group names are of the following: • **radius**—Uses the global pool of RADIUS servers for accounting. • *named-group*—Uses a named subset of TACACS+ or RADIUS servers for accounting. The **local** method uses the local database for accounting. The default method is local, which is used when no server groups are configured or when all the configured server groups do not respond.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show aaa accounting`	(Optional) Displays the configuration AAA accounting default methods.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Using AAA Server VSAs with the Switch

You can use vendor-specific attributes (VSAs) to specify the switch user roles and SNMPv3 parameters on AAA servers.

This section includes the following topics:

About VSAs

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating VSAs between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute seperator value *
```

The protocol is a Cisco attribute for a particular type of authorization, separator is an equal sign (=) for mandatory attributes, and an asterisk (`*`) indicates optional attributes.

When you use RADIUS servers for authentication on the switch, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

VSA Format

The following VSA protocol options are supported by the switch:

- Shell—Used in access-accept packets to provide user profile information.
- Accounting—Used in accounting-request packets. If a value contains any white spaces, put it within double quotation marks.

The following attributes are supported by the switch:

- roles—Lists all the roles assigned to the user. The value field is a string that stores the list of group names delimited by white space.
- accountinginfo—Stores additional accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch, and it can only be used with the accounting protocol-related PDUs.

Specifying User Roles and SNMPv3 Parameters on AAA Servers

You can use the VSA cisco-av-pair on AAA servers to specify user role mapping for the switch using this format:

```
shell:roles="roleA roleB …"
```

If you do not specify the role option in the cisco-av-pair attribute, the default user role is network-operator.

You can also specify your SNMPv3 authentication and privacy protocol attributes as follows:

```
shell:roles="roleA roleB..." snmpv3:auth=SHA priv=AES-128
```

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If you do not specify these options in the cisco-av-pair attribute, MD5 and DES are the default authentication protocols.

For more information on user roles, see Chapter 22, “Configuring User Accounts and RBAC.”

Displaying and Clearing the Local AAA Accounting Log

The switch maintains a local log for the AAA accounting activity. To display this log and clear it, perform this task:

	Command	Purpose
Step 1	`switch# show accounting log [size] [start-time year month day hh:mm:ss]`	Displays the accounting log contents. By default, the command output contains up to 250,000 bytes of the accounting log. You can use the size argument to limit command output. The range is from 0 to 250000 bytes. You can also specify a start time for the log output.
Step 2	`switch# clear accounting log`	(Optional) Clears the accounting log contents.

Verifying AAA Configuration

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

Command	Purpose
show aaa accounting	Displays AAA accounting configuration.
show aaa authentication [**login** {**error-enable** | **mschap**}]	Displays AAA authentication information.
show aaa groups	Displays the AAA server group configuration.
show running-config aaa [all]	Displays the AAA configuration in the running configuration.
show startup-config aaa	Displays the AAA configuration in the startup configuration.

Example AAA Configuration

The following example shows how to configure AAA:

```
aaa authentication login default group radius
aaa authentication login console group radius
aaa accounting default group radius
```

Default Settings

Table 17-4 lists the default settings for AAA parameters.

Table 17-4 Default AAA Parameters

Parameters	Default
Console authentication method	local
Default authentication method	local
Login authentication failure messages	Disabled
MS-CHAP authentication	Disabled
Default accounting method	local
Accounting log display length	250 KB

CHAPTER 18

Configuring RADIUS

This chapter describes how to configure the Remote Access Dial-In User Service (RADIUS) protocol on the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

This chapter includes the following sections:

Information About RADIUS

The RADIUS distributed client/server system allows you to secure networks against unauthorized access. In the Cisco implementation, RADIUS clients run on the switch and send authentication and accounting requests to a central RADIUS server that contains all user authentication and network service access information.

This section includes the following topics:

RADIUS Network Environments

RADIUS can be implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

You can use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor network devices, each supporting RADIUS.

  For example, network devices from several vendors can use a single RADIUS server-based security database.

- Networks already using RADIUS.

  You can add a switch with RADIUS to the network. This action might be the first step when you make a transition to a AAA server.

- Networks that require resource accounting.

  You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of the RADIUS access control and accounting software to meet special security and billing needs.

- Networks that support authentication profiles.

  Using the RADIUS server in your network, you can configure AAA authentication and set up per-user profiles. Per-user profiles enable the switch to better manage ports using their existing RADIUS solutions and to efficiently manage shared resources to offer different service-level agreements.

RADIUS Operation

When a user attempts to log in and authenticate to a switch using RADIUS, the following process occurs:

1. The user is prompted for and enters a username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.
3. The user receives one of the following responses from the RADIUS server:
   - ACCEPT—The user is authenticated.
   - REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.
   - CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
   - CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.

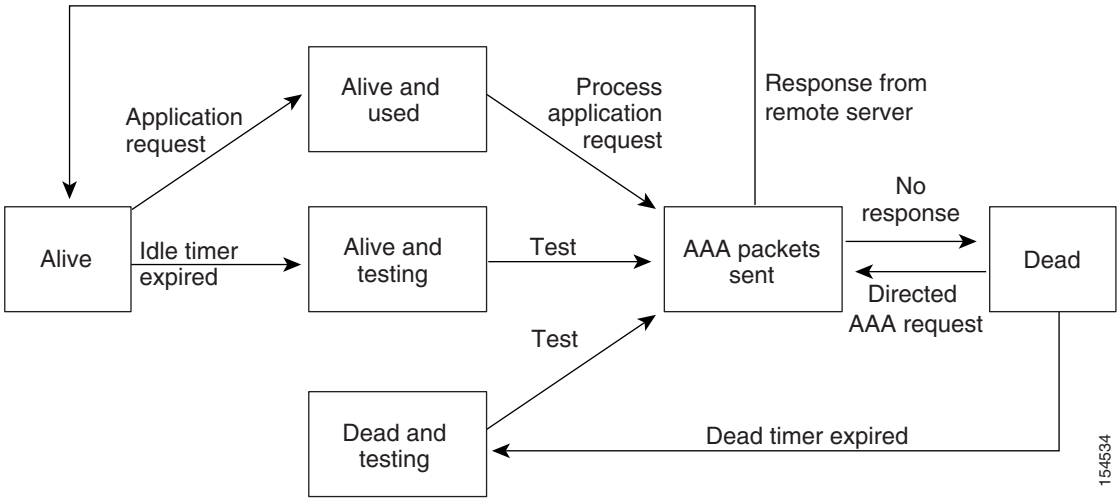
The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 or IPv6 address, access list, and user timeouts.

RADIUS Server Monitoring

An unresponsive RADIUS server can cause delay in processing of AAA requests. You can configure the switch to periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. The switch marks unresponsive RADIUS servers as dead and does not send AAA requests to any dead RADIUS servers. The switch periodically monitors the dead RADIUS servers and brings them to the alive state once they respond. This monitoring process verifies that a RADIUS server is in a working state before real AAA requests are sent its way. Whenever a RADIUS server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the switch displays an error message that a failure is taking place. See Figure 18-1.

Figure 18-1 RADIUS Server States



Note The monitoring interval for alive servers and dead servers are different and can be configured by the user. The RADIUS server monitoring is performed by sending a test authentication request to the RADIUS server.

Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific attributes (VSAs) between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is an equal sign (=) for mandatory attributes, and an asterisk (*) indicates optional attributes.

When you use RADIUS servers for authentication on a switch, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

The following VSA protocol options are supported by the switch:

- Shell—Used in access-accept packets to provide user profile information.
- Accounting—Used in accounting-request packets. If a value contains any white spaces, you should enclose the value within double quotation marks.

The switch supports the following attributes:

- roles—Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space.
- accountinginfo—Stores accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch. It can be used only with the accounting protocol data units (PDUs).

Prerequisites for RADIUS

RADIUS has the following prerequisites:

- Obtain IPv4 or IPv6 addresses or hostnames for the RADIUS servers.
- Obtain preshared keys from the RADIUS servers.
- Ensure that the switch is configured as a RADIUS client of the AAA servers.

Guidelines and Limitations

You can configure a maximum of 64 RADIUS servers on the switch.

Configuring RADIUS Servers

To configure RADIUS servers, perform the following steps:

Step 1 Establish the RADIUS server connections to the switch.

See the "Configuring RADIUS Server Hosts" section on page 18-5.

Step 2 Configure the preshared secret keys for the RADIUS servers.

See the "Configuring Global Preshared Keys" section on page 18-6.

Step 3 If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods.

See the "Allowing Users to Specify a RADIUS Server at Login" section on page 18-8 and the "Configuring AAA" section on page 17-6.

Step 4 If needed, configure any of the following optional parameters:

- Dead-time interval

  See the "The following example shows how to configure periodic RADIUS server monitoring:" section on page 18-12.

- Allow specification of a RADIUS server at login

See the "Allowing Users to Specify a RADIUS Server at Login" section on page 18-8).

- Transmission retry count and timeout interval

  See the "Configuring the Global RADIUS Transmission Retry Count and Timeout Interval" section on page 18-9.

- Accounting and authentication attributes

  See the "Configuring Accounting and Authentication Attributes for RADIUS Servers" section on page 18-10.

Step 5 If needed, configure periodic RADIUS server monitoring.

See the "Configuring Periodic RADIUS Server Monitoring" section on page 18-11.

The following topics describe the RADIUS configuration procedure in more details:

Configuring RADIUS Server Hosts

You must configure the IPv4 or IPv6 address or the hostname for each RADIUS server that you want to use for authentication. All RADIUS server hosts are added to the default RADIUS server group. You can configure up to 64 RADIUS servers.

To configure a RADIUS server host, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config) #radius-server host {ipv4-address | ipv6-address | host-name}`	Specifies the IPv4 or IPv6 address or hostname for a RADIUS server.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure a RADIUS server host:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1
switch(config)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Configuring Global Preshared Keys

You can configure preshared keys at the global level for all servers used by the switch. A preshared key is a shared secret text string between the switch and the RADIUS server hosts.

To configure global preshared keys, obtain the preshared key values for the remote RADIUS servers and perform this task:

:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# radius-server key [0 | 7] key-value`	Specifies a preshared key for all RADIUS servers. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. By default, no preshared key is configured.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show radius-server`	(Optional) Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to obtain the preshared key values for a remote RADIUS server:

```
switch# configure terminal
switch(config)# radius-server key 0 QsEfThUkO
switch(config)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Configuring RADIUS Server Preshared Keys

You can configure preshared keys for a RADIUS server. A preshared key is a shared secret text string between the switch and the RADIUS server host.

To configure radius server preshared keys, obtain the preshared key values for the remote RADIUS servers and perform this task:

:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# radius-server host {ipv4-address | ipv6-address | host-name} key [0 | 7] key-value`	Specifies a preshared key for a specific RADIUS server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show radius-server`	(Optional) Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure a preshared keys for a RADIUS server:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 key 0 PlIjUhYg
switch(config)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Configuring RADIUS Server Groups

You can specify one or more remote AAA servers for authentication using server groups. All members of a group must belong to the RADIUS protocol. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time but they only take effect when you apply them to a AAA service. For information on AAA services, see the "Remote AAA Services" section on page 17-2.

To configure radius server groups, perform this task:

:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# aaa group server radius group-name`	Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group. The *group-name* argument is a case-sensitive alphanumeric string with a maximum length of 127 characters.

	Command	Purpose
Step 3	`switch(config-radius)# server {ipv4-address |ipv6-address | server-name}`	Configures the RADIUS server as a member of the RADIUS server group. **Tip** If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.
Step 4	`switch(config-radius)# deadtime minutes`	(Optional) Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value. See the example that shows how to configure periodic RADIUS server monitoring.
Step 5	`switch(config-radius)# exit`	Exits configuration mode.
Step 6	`switch(config) #show radius-server group [GROUP-NAME]`	(Optional) Displays the RADIUS server group configuration.
Step 7	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure a RADIUS server group:

```
switch# configure terminal
switch(config)# aaa group server radius RadServer
switch(config-radius)# server 10.10.1.1
switch(config-radius)# deadtime 30
switch(config-radius)# use-vrf management
switch(config-radius)# exit
switch(config)# show radius-server group
switch(config)# copy running-config startup-config
```

Allowing Users to Specify a RADIUS Server at Login

Note By default, the switch forwards an authentication request based on the default AAA authentication method. You can configure the switch to allow the user to specify a VRF and RADIUS server to send the authenticate request by enabling the directed-request option. If you enable this option, the user can log in as *username@hostname,* where *hostname* is the name of a configured RADIUS server. User specified logins are only supported for Telnet sessions.

To allow users to specify a RADIUS server at login, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# radius-server directed-request`	Allows users to specify a RADIUS server to send the authentication request when logging in. The default is disabled.
Step 3	`switch(config)# exit`	Exits configuration mode.

	Command	Purpose
Step 4	`switch# show radius-server directed-request`	(Optional) Displays the directed request configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring the Global RADIUS Transmission Retry Count and Timeout Interval

You can configure a global retransmission retry count and timeout interval for all RADIUS servers. By default, a switch retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. The timeout interval determines how long the switch waits for responses from RADIUS servers before declaring a timeout failure.

To configure the global RADIUS transmission retry count and timeout interval, perform this task:

:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# radius-server retransmit` *count*	Specifies the retransmission count for all RADIUS servers. The default retransmission count is 1 and the range is from 0 to 5.
Step 3	`switch(config)# radius-server timeout` *seconds*	Specifies the transmission timeout interval for RADIUS servers. The default timeout interval is 5 seconds and the range is from 1 to 60 seconds.
Step 4	`switch(config)# exit`	Exits configuration mode.
Step 5	`switch# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 6	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring the RADIUS Transmission Retry Count and Timeout Interval for a Server

By default, a switch retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. You can also set a timeout interval that the switch waits for responses from RADIUS servers before declaring a timeout failure.

To configure RADIUS transmission retry count and timeout interval for a server, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# radius-server host {ipv4-address |ipv6-address | host-name} retransmit count`	Specifies the retransmission count for a specific server. The default is the global value. **Note** The retransmission count value specified for a RADIUS server overrides the count specified for all RADIUS servers.
Step 3	`switch(config)# switch(config)# radius-server host {ipv4-address |ipv6-address | host-name} timeout seconds`	Specifies the transmission timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for a RADIUS server overrides the interval value specified for all RADIUS servers.
Step 4	`switch(config)# exit`	Exits configuration mode.
Step 5	`switch# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 6	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure RADIUS transmission retry count and timeout interval for a server:

```
switch# configure terminal
switch(config)# radius-server host server1 retransmit 3
switch(config)# radius-server host server1 timeout 10
switch(config)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Configuring Accounting and Authentication Attributes for RADIUS Servers

You can specify that a RADIUS server is to be used only for accounting purposes or only for authentication purposes. By default, RADIUS servers are used for both accounting and authentication. You can also specify the destination UDP port numbers where RADIUS accounting and authentication messages should be sent.

To configure the accounting and authentication attributes for RADIUS servers, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config) #radius-server host {ipv4-address | ipv6-address | host-name} acct-port udp-port`	(Optional) Specifies a UDP port to use for RADIUS accounting messages.The default UDP port is 1812. The range is from 0 to 65535.
Step 3	`switch(config)# radius-server host {ipv4-address | ipv6-address | host-name} accounting`	(Optional) Specifies that the specified RADIUS server it to be used only for accounting purposes. The default is both accounting and authentication.

	Command	Purpose
Step 4	`switch(config)# radius-server host {ipv4-address | ipv6-address | host-name} auth-port udp-port`	(Optional) Specifies a UDP port to use for RADIUS authentication messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 5	`switch(config)# radius-server host {ipv4-address | ipv6-address | host-name} authentication`	(Optional) Specifies that the specified RADIUS server only be used for authentication purposes. The default is both accounting and authentication.
Step 6	`switch(config)# exit`	Exits configuration mode.
Step 7	`switch(config)# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 8	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure the accounting and authentication attributes for a RADIUS server:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 acct-port 2004
switch(config)# radius-server host 10.10.1.1 accounting
switch(config)# radius-server host 10.10.2.2 auth-port 2005
switch(config)# radius-server host 10.10.2.2 authentication
switch(config)# exit
switch(config)# show radius-server
switch# copy running-config startup-config
```

Configuring Periodic RADIUS Server Monitoring

You can monitor the availability of RADIUS servers. These parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval during which a RADIUS server receives no requests before the switch sends out a test packet. You can configure this option to test servers periodically.

Note For security reasons, we recommend that you do not configure a test username that is the same as an existing user in the RADIUS database.

The test idle timer specifies the interval during which a RADIUS server receives no requests before the switch sends out a test packet.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, the switch does not perform periodic RADIUS server monitoring.

To configure periodic RADIUS server monitoring, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# radius-server host {ipv4-address | ipv6-address | host-name} test {idle-time minutes | password password [idle-time minutes] | username name [password password [idle-time minutes]]}`	Specifies parameters for server monitoring. The default username is test and the default password is test. The default value for the idle timer is 0 minutes. The valid range is 0 to 1440 minutes. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.
Step 3	`switch(config)# radius-server deadtime minutes`	Specifies the number of minutes before the switch checks a RADIUS server that was previously unresponsive. The default value is 0 minutes. The valid range is 1 to 1440 minutes.
Step 4	`switch(config)# exit`	Exits configuration mode.
Step 5	`switch# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 6	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure periodic RADIUS server monitoring:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 test username user1 password Ur2Gd2BH
idle-time 3
switch(config)# radius-server deadtime 5
switch(config)# exit
switch# show radius-server
switch# copy running-config startup-config
```

Configuring the Dead-Time Interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time that the switch waits after declaring a RADIUS server is dead, before sending out a test packet to determine if the server is now alive. The default value is 0 minutes.

Note When the dead-time interval is 0 minutes, RADIUS servers are not marked as dead even if they are not responding. You can configure the dead-time interval for a RADIUS server group (see the "Configuring RADIUS Server Groups" section on page 18-7).

To configure dead time interval, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`#switch(config)# radius-server deadtime`	Configures the dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.

	Command	Purpose
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show radius-server`	(Optional) Displays the RADIUS server configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Manually Monitoring RADIUS Servers or Groups

To manually send a test message to a RADIUS server or to a server group, perform this task:

	Command	Purpose
Step 1	`switch# test aaa server radius {ipv4-address | ipv6-address | server-name} [vrf vrf-name] username password`	Sends a test message to a RADIUS server to confirm availability.
Step 1	`switch# test aaa group group-name username password`	Sends a test message to a RADIUS server group to confirm availability.

The following example shows how to manually send a test message to a RADIUS server:

```
switch# test aaa server radius 10.10.1.1 user1 Ur2Gd2BH
switch# test aaa group RadGroup user2 As3He3CI
```

Verifying RADIUS Configuration

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

Command	Purpose
show running-config radius [**all**]	Displays the RADIUS configuration in the running configuration.
show startup-config radius	Displays the RADIUS configuration in the startup configuration.
show radius-server [*server-name* | *ipv4-address* | *ipv6-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured RADIUS server parameters.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 4000I and 4005I Switch Module for IBM BladeCenter NX-OS Command Reference*.

Displaying RADIUS Server Statistics

To display the statistics the switch maintains for RADIUS server activity, perform this task:

Command	Purpose
switch# switch# **show radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}	Displays the RADIUS statistics.

The following example shows how to display statistics:

```
switch# show radius-server statistics 10.10.1.1
```

Example RADIUS Configuration

The following example shows how to configure RADIUS:

```
radius-server key 7 "ToIkLhPpG"
radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication accounting
aaa group server radius RadServer
    server 10.10.1.1
use-vrf management
```

Default Settings

Table 18-1 lists the default settings for RADIUS parameters.

Table 18-1 Default RADIUS Parameters

Parameters	Default
Server roles	Authentication and accounting
Dead timer interval	0 minutes
Retransmission count	1
Retransmission timer interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

CHAPTER 19

Configuring TACACS+

This chapter describes how to configure the Terminal Access Controller Access Control System Plus (TACACS+) protocol on the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

This chapter includes the following sections:

Information About TACACS+

The TACACS+ security protocol provides centralized validation of users attempting to gain access to the switch. TACACS+ services are maintained in a database on a TACACS+ daemon typically running on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your switch are available.

TACACS+ provides for separate authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service (authentication, authorization, and accounting) independently. Each service is associated with its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The TACACS+ client/server protocol uses TCP (TCP port 49) for transport requirements. The switch provide centralized authentication using the TACACS+ protocol.

This section includes the following topics:

- TACACS+ Server Monitoring, page 19-3

TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the switch can authorize access without authenticating.
- Uses the TCP transport protocol to send data between the AAA client and server, making reliable transfers with a connection-oriented protocol.
- Encrypts the entire protocol payload between the switch and the AAA server to ensure higher data confidentiality. The RADIUS protocol only encrypts passwords.

User Login with TACACS+

When a user attempts a Password Authentication Protocol (PAP) login to a switch using TACACS+, the following actions occur:

1. When the switch establishes a connection, it contacts the TACACS+ daemon to obtain the username and password.

Note TACACS+ allows an arbitrary conversation between the daemon and the user until the daemon receives enough information to authenticate the user. This action is usually completed by prompting for a username and password combination, but may include prompts for other items, such as the maiden name of the mother of a user.

2. The switch will receive one of the following responses from the TACACS+ daemon:
   - ACCEPT—User authentication succeeds and service begins. If the switch requires user authorization, authorization begins.
   - REJECT—User authentication failed. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.
   - ERROR—An error occurred at some time during authentication dither at the daemon or in the network connection between the daemon and the switch. If the switch receives an ERROR response, the switch tries to use an alternative method for authenticating the user.

   The user also undergoes an additional authorization phase, if authorization has been enabled on the switch. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.
3. If TACACS+ authorization is required, the switch again contacts the TACACS+ daemon and it returns an ACCEPT or REJECT authorization response. An ACCEPT response contains attributes that are used to direct the EXEC or NETWORK session for that user and determines the services that the user can access.

   Services include the following:
   - Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services
   - Connection parameters, including the host or client IP address (IPv4 or IPv6), access list, and user timeouts

Default TACACS+ Server Encryption Type and Preshared Key

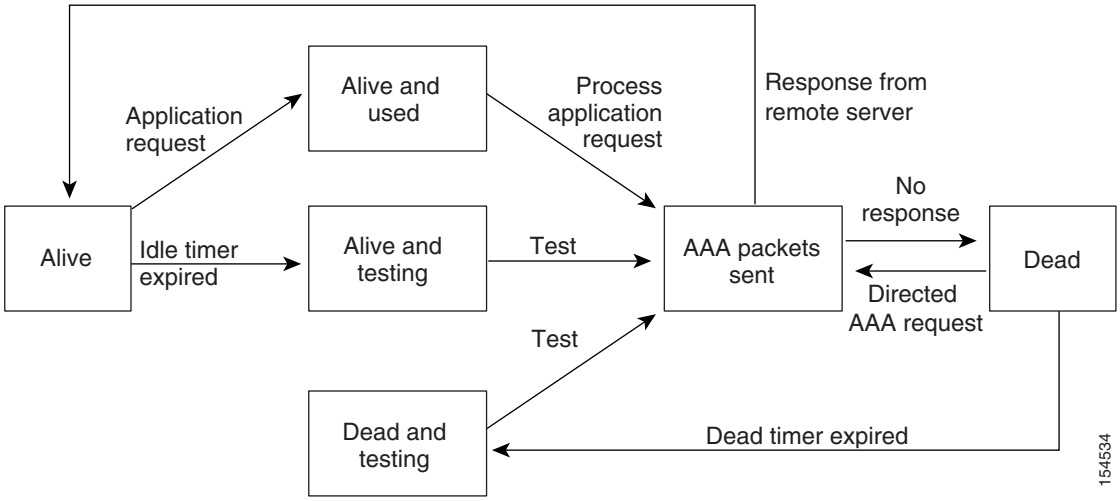
You must configure the TACACS+ preshared key to authenticate the switch to the TACACS+ server. A preshared key is a secret text string shared between the switch and the TACACS+ server host. The length of the key is restricted to 63 characters and can include any printable ASCII characters (white spaces are not allowed). You can configure a global preshared secret key for all TACACS+ server configurations on the switch to use.

You can override the global preshared key assignment by explicitly using the **key** option when configuring an individual TACACS+ server.

TACACS+ Server Monitoring

An unresponsive TACACS+ server can delay the processing of AAA requests. A switch can periodically monitor an TACACS+ server to check whether it is responding (or alive) to save time in processing AAA requests. The switch marks unresponsive TACACS+ servers as dead and does not send AAA requests to any dead TACACS+ servers. A switch periodically monitors dead TACACS+ servers and brings them to the alive state once they are responding. This process verifies that a TACACS+ server is in a working state before real AAA requests are sent its way. Whenever an TACACS+ server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the switch displays an error message that a failure is taking place before it can impact performance. See Figure 19-1.

Figure 19-1 TACACS+ Server States



Note The monitoring interval for alive servers and dead servers are different and can be configured by the user. The TACACS+ server monitoring is performed by sending a test authentication request to the TACACS+ server.

Prerequisites for TACACS+

TACACS+ has the following prerequisites:

- Obtain the IPv4 or IPv6 addresses or hostnames for the TACACS+ servers.

- Obtain the preshared keys from the TACACS+ servers, if any.
- Ensure that the switch is configured as a TACACS+ client of the AAA servers.

Guidelines and Limitations

You can configure a maximum of 64 TACACS+ servers on the switch.

Configuring TACACS+

This section includes the following topics:

TACACS+ Server Configuration Process

To configure TACACS+ servers, perform the following steps:

Step 1 Enable TACACS+.

See the "Enabling TACACS+" section on page 19-5.

Step 2 Establish the TACACS+ server connections to the switch.

See the "Configuring TACACS+ Server Hosts" section on page 19-5.

Step 3 Configure the preshared secret keys for the TACACS+ servers.

See the "Configuring Global Preshared Keys" section on page 19-6 and the "Configuring TACACS+ Server Preshared Keys" section on page 19-7.

Step 4 If needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods.

See the "Configuring TACACS+ Server Groups" section on page 19-7 and the "Configuring AAA" section on page 17-6.

Step 5 If needed, configure any of the following optional parameters:

- Dead-time interval
- Allow TACACS+ server specification at login
- Timeout interval

  See the "Configuring the Global TACACS+ Timeout Interval" section on page 19-9.
- TCP port

  See the "Configuring TCP Ports" section on page 19-10.

Step 6 If needed, configure periodic TACACS+ server monitoring.

See the "Configuring Periodic TACACS+ Server Monitoring" section on page 19-11.

Enabling TACACS+

By default, the TACACS+ feature is disabled on the switch. To explicitly enable the TACACS+ feature to access the configuration and verification commands for authentication, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# feature tacacs+`	Enables TACACS+.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring TACACS+ Server Hosts

To access a remote TACACS+ server, you must configure the IPv4 or IPv6 address or the hostname for the TACACS+ server on the switch. All TACACS+ server hosts are added to the default TACACS+ server group.You can configure up to 64 TACACS+ servers.

If a preshared key is not configured for a configured TACACS+ server, a warning message is issued if a global key is not configured. If a TACACS+ server key is not configured, the global key (if configured) is used for that server (see the "Configuring Global Preshared Keys" section on page 19-6 and the "Configuring TACACS+ Server Preshared Keys" section on page 19-7).

Before you configure TACACS+ server hosts, you should do the following:

- Enable TACACS+ (see the "Enabling TACACS+" section on page 19-5).
- Obtain the IPv4 or IPv6 addresses or the hostnames for the remote TACACS+ servers.

To configure TACACS+ server hosts, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# tacacs-server host {ipv4-address | ipv6-address|host-name}`	Specifies the IPv4 or IPv6 address or hostname for a TACACS+ server.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show tacacs-server`	(Optional) Displays the TACACS+ server configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

You can delete a TACACS+ server host from a server group.

Configuring Global Preshared Keys

You can configure preshared keys at the global level for all servers used by the switch. A preshared key is a shared secret text string between the switch and the TACACS+ server hosts.

Before you configure preshared keys, you should do the following:

- Enable TACACS+ (see the "Enabling TACACS+" section on page 19-5).
- Obtain the preshared key values for the remote TACACS+ servers.

To configure global preshared keys, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# tacacs-server key [0|7] key-value`	Specifies a preshared key for all TACACS+ servers. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. By default, no preshared key is configured.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show tacacs-server`	(Optional) Displays the TACACS+ server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure global preshared keys:

```
switch# configure terminal
switch(config)# tacacs-server key 0 QsEfThUkO
switch(config)# exit
```

```
switch# show tacacs-server
switch# copy running-config startup-config
```

Configuring TACACS+ Server Preshared Keys

You can configure preshared keys for a TACACS+ server. A preshared key is a shared secret text string between the switch and the TACACS+ server host.

To configure the TACACS+ preshared keys, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# tacacs-server host {ipv4-address|ipv6-address|host-name} key [0|7] key-value`	Specifies a preshared key for a specific TACACS+ server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show tacacs-server`	(Optional) Displays the TACACS+ server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure the TACACS+ preshared keys:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.1.1 key 0 PlIjUhYg
switch(config)# exit
switch# show tacacs-server
switch# copy running-config startup-config
```

Configuring TACACS+ Server Groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must belong to the TACACS+ protocol. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time but they only take effect when you apply them to a AAA service. For information on AAA services, see the "Remote AAA Services" section on page 17-2.

To configure TACACS+ server groups, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# aaa group server tacacs+` *group-name*	Creates a TACACS+ server group and enters the TACACS+ server group configuration mode for that group.
Step 3	`switch(config-tacacs+)# server` {*ipv4-address*|*ipv6-address*|*host-name*}	Configures the TACACS+ server as a member of the TACACS+ server group. **Tip** If the specified TACACS+ server is not found, configure it using the **tacacs-server host** command and retry this command.
Step 4	`switch(config-tacacs+)# deadtime` *minutes*	(Optional) Configures the monitoring dead time. The default is 0 minutes. The range is from 0 through 1440. **Note** If the dead-time interval for a TACACS+ server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	`switch(config-tacacs+)# exit`	Exits configuration mode.
Step 6	`switch(config)# show tacacs-server groups`	(Optional) Displays the TACACS+ server group configuration.
Step 7	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure a TACACS+ server group:

```
switch# configure terminal
switch(config)# aaa group server tacacs+ TacServer
switch(config-tacacs+)# server 10.10.2.2
switch(config-tacacs+)# deadtime 30
switch(config-tacacs+)# exit
switch(config)# show tacacs-server groups
switch(config)# copy running-config startup-config
```

Specifying a TACACS+ Server at Login

You can configure the switch to allow the user to specify which TACACS+ server to send the authenticate request by enabling the directed-request option. By default, a switch forwards an authentication request based on the default AAA authentication method. If you enable this option, the user can log in as *username@hostname,* where *hostname* is the name of a configured RADIUS server.

Note User specified logins are only supported for Telnet sessions.

To specify a TACACS+ server at login, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# tacacs-server directed-request`	Allows users to specify a TACACS+ server to send the authentication request when logging in. The default is disabled.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show tacacs-server directed-request`	(Optional) Displays the TACACS+ directed request configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring the Global TACACS+ Timeout Interval

You can set a global timeout interval that the switch waits for responses from all TACACS+ servers before declaring a timeout failure. The timeout interval determines how long the switch waits for responses from TACACS+ servers before declaring a timeout failure.

To specify a TACACS+ global timeout interval, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# tacacs-server timeout` *seconds*	Specifies the timeout interval for TACACS+ servers. The default timeout interval is 5 second and the range is from 1 to 60 seconds.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show tacacs-server`	(Optional) Displays the TACACS+ server configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring the Timeout Interval for a Server

You can set a timeout interval that the switch waits for responses from a TACACS+ server before declaring a timeout failure. The timeout interval determines how long the switch waits for responses from a TACACS+ server before declaring a timeout failure.

To configure the timeout interval for a server, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# switch(config)# tacacs-server host {ipv4-address|ipv6-address|host-name} timeout seconds`	Specifies the timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for a TACACS+ server overrides the global timeout interval value specified for all TACACS+ servers.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show tacacs-server`	(Optional) Displays the TACACS+ server configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring TCP Ports

You can configure another TCP port for the TACACS+ servers if there are conflicts with another application. By default, switch uses port 49 for all TACACS+ requests.

To configure TCP ports, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# tacacs-server host {ipv4-address|ipv6-address|host-name} port tcp-port`	Specifies the UDP port to use for TACACS+ accounting messages.The default TCP port is 49. The range is from 1 to 65535.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show tacacs-server`	(Optional) Displays the TACACS+ server configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure TCP ports:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.1.1 port 2
switch(config)# exit
switch# show tacacs-server
switch# copy running-config startup-config
```

Configuring Periodic TACACS+ Server Monitoring

You can monitor the availability of TACACS+ servers. These parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval in which a TACACS+ server receives no requests before the switch sends out a test packet.You can configure this option to test servers periodically, or you can run a one-time only test.

Note To protect network security, we recommend that you use a username that is not the same as an existing username in the TACACS+ database.

The test idle timer specifies the interval in which a TACACS+ server receives no requests before the switch sends out a test packet.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic TACACS+ server monitoring is not performed.

To configure periodic TACACS+ server monitoring, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# tacacs-server host {ipv4-address | ipv6-address | host-name} test {idle-time minutes | password password [idle-time minutes] | username name [password password [idle-time minutes]]}`	Specifies parameters for server monitoring. The default username is test and the default password is test. The default value for the idle timer is 0 minutes and the valid range is 0 to 1440 minutes. **Note** For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.
Step 3	`switch(config)# tacacs-server dead-time minutes`	Specifies the number of minutes before the switch checks a TACACS+ server that was previously unresponsive. The default value is 0 minutes and the valid range is 0 to 1440 minutes.
Step 4	`switch(config)# exit`	Exits configuration mode.
Step 5	`switch# show tacacs-server`	(Optional) Displays the TACACS+ server configuration.
Step 6	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure periodic TACACS+ server monitoring:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.1.1 test username user1 password Ur2Gd2BH
idle-time 3
switch(config)# tacacs-server dead-time 5
switch(config)# exit
switch# show tacacs-server
switch# copy running-config startup-config
```

Configuring the Dead-Time Interval

You can configure the dead-time interval for all TACACS+ servers. The dead-time interval specifies the time that the switch waits, after declaring a TACACS+ server is dead, before sending out a test packet to determine if the server is now alive.

Note When the dead-timer interval is 0 minutes, TACACS+ servers are not marked as dead even if they are not responding. You can configure the dead-timer per group (see the "Configuring TACACS+ Server Groups" section on page 19-7).

To configure the dead-time interval for all TACACS+ servers, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# tacacs-server deadtime minutes`	Configures the global dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# show tacacs-server`	(Optional) Displays the TACACS+ server configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Manually Monitoring TACACS+ Servers or Groups

To manually issue a test message to a TACACS+ server or to a server group, perform this task:

	Command	Purpose
Step 1	`switch# test aaa server tacacs+ {ipv4-address|ipv6-address|host-name} [vrf vrf-name] username password`	Sends a test message to a TACACS+ server to confirm availability.
Step 2	`switch# test aaa group group-name username password`	Sends a test message to a TACACS+ server group to confirm availability.

The following example shows how to manually issue a test message:

```
switch# test aaa server tacacs+ 10.10.1.1 user1 Ur2Gd2BH
switch# test aaa group TacGroup user2 As3He3CI
```

Disabling TACACS+

You can disable TACACS+.

Caution When you disable TACACS+, all related configurations are automatically discarded.

To disable TACACS+, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# feature tacacs+`	Enables TACACS+.
Step 3	`switch(config)# exit`	Exits configuration mode.
Step 4	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Displaying TACACS+ Statistics

To display the statistics the switch maintains for TACACS+ activity, perform this task:

Command	Purpose
`switch# show tacacs-server statistics {hostname|ipv4-address|ipv6-address}`	Displays the TACACS+ statistics.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 4000I and 4005I Switch Module for IBM BladeCenter NX-OS Command Reference*.

Verifying TACACS+ Configuration

To display TACACS+ configuration information, perform one of these tasks:

Command	Purpose
show running-config tacacs [**all**]	Displays the TACACS+ configuration in the running configuration.
show startup-config tacacs	Displays the TACACS+ configuration in the startup configuration.
show tacacs-server [*host-name* | *ipv4-address* | *ipv6-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured TACACS+ server parameters.

Example TACACS+ Configuration

The following example shows how to configure TACACS+:

```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# tacacs-server key 7 "ToIkLhPpG"
switch(config)# tacacs-server host 10.10.2.2 key 7 "ShMoMhT1"
switch(config)# aaa group server tacacs+ TacServer
switch(config-tacacs+)# server 10.10.2.2
switch(config-tacacs+)# use-vrf management
```

Default Settings

Table 19-1 lists the default settings for TACACS+ parameters.

Table 19-1 Default TACACS+ Parameters

Parameters	Default
TACACS+	Disabled
Dead timer interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

CHAPTER 20

Configuring SSH and Telnet

This chapter describes how to configure Secure Shell Protocol (SSH) and Telnet on the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

This chapter includes the following sections:

Information About SSH and Telnet

This section includes the following topics:

SSH Server

The SSH server feature enables a SSH client to make a secure, encrypted connection to a switch. SSH uses strong encryption for authentication. The SSH server in the switch will interoperate with publicly and commercially available SSH clients.

The user authentication mechanisms supported for SSH are RADIUS, TACACS+, and the use of locally stored usernames and passwords.

SSH Client

The SSH client feature is an application running over the SSH protocol to provide device authentication and encryption. The SSH client enables a switch to make a secure, encrypted connection to another switch or to any other device running the SSH server. This connection provides an outbound connection that is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

The SSH client in the switch works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communications to the switch. You can use SSH keys for the following SSH options:

- SSH version 2 using Rivest, Shamir, and Adelman (RSA) public-key cryptography
- SSH version 2 using the Digital System Algrorithm (DSA)

Be sure to have an SSH server key-pair with the appropriate version before enabling the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts three types of key-pairs for use by SSH version 2:

- The **dsa** option generates the DSA key-pair for the SSH version 2 protocol.
- The **rsa** option generates the RSA key-pair for the SSH version 2 protocol.

By default, the switch generates an RSA key using 1024 bits.

Caution If you delete all of the SSH keys, you cannot start the SSH services.

Telnet Server

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site, and then passes the keystrokes from one system to the other. Telnet can accept either an IP address or a domain name as the remote system address.

The Telnet server is enabled by default on the switch.

Prerequisites for SSH

SSH has a prerequisite that you have IP on a Layer 3 interface, out-of-band on the mgmt 0 interface, or inband on an Ethernet interface configured.

Guidelines and Limitations

SSH has the following configuration guidelines and limitations:

- The switch supports only SSH version 2 (SSHv2).

Configuring SSH

This section includes the following topics:

Generating SSH Server Keys

You can generate an SSH server key based on your security requirements. The default SSH server key is an RSA key generated using 1024 bits. To generate SSH server keys, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# ssh key {dsa [force] | rsa [bits [force]]}`	Generates the SSH server key. The *bits* argument is the number of bits used to generate the key. The range is 768 to 2048 and the default value is 1024. Use the **force** keyword to replace an existing key.
Step 3	`switch(config)# exit`	Exits global configuration mode.
Step 4	`switch# show ssh key`	(Optional) Displays the SSH server keys.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to generate an SSH server key:

```
switch# configure terminal
switch(config)# ssh key rsa 2048
switch(config)# exit
switch# show ssh key
switch# copy running-config startup-config
```

Specifying the SSH Public Keys for User Accounts

You can configure an SSH public key to log in using the SSH client without being prompted for a password. You can specify the SSH public key in one of three different formats:

- Open SSH format
- IETF SECSH format

- Public Key Certificate in PEM format

Specifying the SSH Public Keys in Open SSH Format

You can specify the SSH public keys in SSH format for user accounts.

To specify the SSH public keys in open SSH format, generate an SSH public key in open SSH format and perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# username` *username* `sshkey` *ssh-key*	Configures the SSH public key in SSH format.
Step 3	`switch(config)# exit`	Exits global configuration mode.
Step 4	`switch# show user-account`	(Optional) Displays the user account configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to specify an SSH public keys in open SSH format:

```
switch# configure terminal
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50rv7gsEPjhOBYmsi6PAVKui1nIf/
DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH
3UD/vKyziEh5S4Tplx8=
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

Specifying the SSH Public Keys in IETF SECSH Format

You can specify the SSH public keys in IETF SECSH format for user accounts.

To specify the SSH public keys in IETF SECSH format, generate an SSH public key in IETF SCHSH format, and perform this task:

	Command	Purpose
Step 1	`switch# copy server-file bootflash:`*filename*	Downloads the file containing the SSH key in IETF SECSH format from a server. The server can be FTP, SCP, SFTP, or TFTP.
Step 2	`switch# configure terminal`	Enters configuration mode.
Step 3	`switch(config)# username` *username* `sshkey file` *filename*	Configures the SSH public key in SSH format.
Step 4	`switch(config)# exit`	Exits global configuration mode.
Step 5	`switch# show user-account`	(Optional) Displays the user account configuration.
Step 6	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to specify the SSH public keys in the IETF SECSH format:

```
switch# copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub
switch# configure terminal
switch(config)# username User1 sshkey file bootflash:secsh_file.pub
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

Specifying the SSH Public Keys in PEM-Formatted Public Key Certificate Form

You can specify the SSH public keys in PEM-formatted Public Key Certificate form for user accounts.

To specify the SSH public keys in PEM-formatted Public Key Certificate form, generate an SSH public key in PEM-Formatted Public Key Certificate form and perform this task:

	Command	Purpose
Step 1	`switch# copy server-file bootflash:`*filename*	Downloads the file containing the SSH key in PEM-formatted Public Key Certificate form from a server. The server can be FTP, SCP, SFTP, or TFTP
Step 2	`switch# configure terminal`	Enters configuration mode.
Step 3	`switch# show user-account`	(Optional) Displays the user account configuration.
Step 4	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to specify the SSH public keys in PEM-formatted public key certificate form:

```
switch# copy tftp://10.10.1.1/cert.pem bootflash:cert.pem
switch# configure terminal
switch# show user-account
switch# copy running-config startup-config
```

Starting SSH Sessions to Remote Devices

To start SSH sessions to connect to remote devices from the switch, perform this task:

Command	Purpose
`switch# ssh {`*hostname* `|` *username*`@`*hostname*`} [vrf` *vrf-name*`]`	Creates an SSH session to a remote device. The *hostname* argument can be an IPv4 address, an IPv6 address, or a device name.

Clearing SSH Hosts

When you download a file from a server using SCP or SFTP, you establish a trusted SSH relationship with that server. To clear the list of trusted SSH servers for your user account, perform this task:

Command	Purpose
`switch# clear ssh hosts`	Clears the SSH host sessions.

Disabling the SSH Server

By default, the SSH server is enabled on the switch.

To disable the SSH server to prevent SSH access to the switch, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# no ssh server enable`	Disables the SSH server. The default is enabled.
Step 3	`switch(config)# exit`	Exits global configuration mode.
Step 4	`switch# show ssh server`	(Optional) Displays the SSH server configuration.
Step 5	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Deleting SSH Server Keys

You can delete SSH server keys after you disable the SSH server.

Note To reenable SSH, you must first generate an SSH server key (see "Generating SSH Server Keys" section on page 20-3).

To delete the SSH server keys, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# no ssh server enable`	Disables the SSH server.
Step 3	`switch(config)# no ssh key [dsa | rsa]`	Deletes the SSH server key. The default is to delete all the SSH keys.
Step 4	`switch(config)# exit`	Exits global configuration mode.
Step 5	`switch# show ssh key`	(Optional) Displays the SSH server configuration.
Step 6	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Clearing SSH Sessions

To clear SSH sessions from the switch, perform this task:

	Command	Purpose
Step 1	`switch# show users`	Displays user session information.
Step 2	`switch(config)# clear line vty-line`	Clears a user SSH session.

Configuring Telnet

This section includes the following topics:

Enabling the Telnet Server

By default, the Telnet server is enabled. To disable the Telnet server on your switch, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# telnet server disable`	Disables the Telnet server. The default is enabled.

To reenable the Telnet server, perform this task:

Command	Purpose
`switch(config)# telnet server enable`	Re-enables the Telnet server.

Starting Telnet Sessions to Remote Devices

Before you start a Telnet session to connect to remote devices, you should do the following:

- Obtain the hostname for the remote device and, if needed, the username on the remote device.

- Enable the Telnet server on the switch.
- Enable the Telnet server on the remote device.

To start Telnet sessions to connect to remote devices from your switch, perform this task:

Command	Purpose
`switch# telnet hostname`	Creates a Telnet session to a remote device. The *hostname* argument can be an IPv4 address, an IPv6 address, or a device name.

The following example shows starting a Telnet session to connect to a remote device:

```
switch# telnet 10.10.1.1
Trying 10.10.1.1...
Connected to 10.10.1.1.
Escape character is '^]'.
switch login:
```

Clearing Telnet Sessions

To clear Telnet sessions from the switch, perform this task:

	Command	Purpose
Step 1	`switch# show users`	Displays user session information.
Step 2	`switch(config)# clear line vty-line`	Clears a user Telnet session.

Verifying the SSH and Telnet Configuration

To display the SSH configuration information, perform one of these tasks:

Command	Purpose
show ssh key [dsa | rsa]	Displays SSH server key-pair information.
show running-config security [**all**]	Displays the SSH and user account configuration in the running configuration. The **all** keyword displays the default values for the SSH and user accounts.
show ssh server	Displays the SSH server configuration.
show user-account	Displays user account information.

SSH Example Configuration

To configure SSH, perform the following steps:

Step 1 Generate an SSH server key:

```
switch(config)# ssh key rsa
generating rsa key(1024 bits).....
.
generated rsa key
```

Step 2 Enable the SSH server:

```
switch# configure terminal
switch(config)# ssh server enable
```

Step 3 Display the SSH server key:

```
switch(config)# show ssh key
rsa Keys generated:Sat Sep 29 00:10:39 2007

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAvWhEBsF55oaPHNDBnpXOTw6+/OdHoLJZKr+MZm99n2U0ChzZG4svRW
mHuJY4PeDWl0e5yE3g3EO3pjDDmt923siNiv5aSga60K36lr39HmXL6VgpRVn1XQFiBwn4na+H1d3Q0hDt+uWEA0tk
a2uOtXlDhliEmn4HVXOjGhFhoNE=

bitcount:1024
fingerprint:
51:6d:de:1c:c3:29:50:88:df:cc:95:f0:15:5d:9a:df
*************************************
could not retrieve dsa key information
*************************************
```

Step 4 Specify the SSH public key in Open SSH format:

```
switch(config)# username User1 sshkey ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1
XswK3OiW4H7YyUyuA50rv7gsEPjhOBYmsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9ig
G30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH3UD/vKyziEh5S4Tplx8=
```

Step 5 Save the configuration:

```
switch(config)# copy running-config startup-config
```

Default Settings

Table 20-1 lists the default settings for SSH parameters.

Table 20-1 Default SSH Parameters

Parameters	Default
SSH server	Enabled
SSH server key	RSA key generated with 1024 bits
RSA key bits for generation	1024
Telnet server	Enabled

CHAPTER 21

Configuring ACLs

This chapter describes how to configure access control lists (ACLs).

This chapter includes the following sections:

Information About ACLs

An ACL is an ordered set of rules that you can use to filter traffic. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the switch determines that an ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether the packet is permitted or denied. If there is no match, the switch applies the applicable default rule. The switch continues processing packets that are permitted and drops packets that are denied. For more information, see the "Implicit Rules" section on page 21-3.

You can use ACLs to protect networks and specific hosts from unnecessary or unwanted traffic. For example, you could use ACLs to disallow HTTP traffic from a high-security network to the Internet. You could also use ACLs to allow HTTP traffic but only to specific sites, using the IP address of the site to identify it in an IP ACL.

This sections includes the following topics:

IP ACL Types and Applications

The Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter supports IPv4 and MAC ACLs for security traffic filtering. The switch allows you to use IP ACLs as port ACLs and VLAN ACLs, as shown in Table 21-1.

Table 21-1 Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Port ACL (PACL)	An ACL is considered a port ACL when you apply it to one of the following: • Ethernet interface • Ethernet port-channel interface When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port.	IPv4 ACLs MAC ACLs
VLAN ACL (VACL)	An ACL is a VACL when you use an access map to associate the ACL with an action, and then apply the map to a VLAN.	IPv4 ACLs MAC ACLs

Application Order

When the switch processes a packet, it determines the forwarding path of the packet. The path determines which ACLs the switch applies to the traffic. The switch applies the Port ACLs first.

Rules

You can create rules in access-list configuration mode by using the **permit** or **deny** command. The switch allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet in order to match the rule.

This section includes the following topics:

Source and Destination

In each rule, you specify the source and the destination of the traffic that matches the rule. You can specify both the source and destination as a specific host, a network or group of hosts, or any host.

Protocols

ACLs allow you to identify traffic by protocol. For your convenience, you can specify some protocols by name. For example, in an IPv4 ACL, you can specify ICMP by name.

You can specify any protocol by number. In IPv4 ACLs, you can specify protocols by the integer that represents the Internet protocol number. For example, you can use 115 to specify Layer 2 Tunneling Protocol (L2TP) traffic.

Implicit Rules

IP ACLs and MAC ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the switch applies them to traffic when no other rules in an ACL match.

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

This implicit rule ensures that the switch denies unmatched IP traffic.

All MAC ACLs include the following implicit rule:

```
deny mac any any
```

This implicit rule ensures that the switch denies unmatched MAC traffic.

Additional Filtering Options

You can identify traffic by using additional options.

IPv4 ACLs support the following additional filtering options:

- Layer 4 protocol
- TCP and UDP ports
- ICMP types and codes
- IGMP types
- Precedence level
- Differentiated Services Code Point (DSCP) value
- TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
- Established TCP connections

MAC ACLs support L3 protocol.

Sequence Numbers

The switch supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the switch. Sequence numbers simplify the following ACL tasks:

- Adding new rules between existing rules—By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.
- Removing a rule—Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

  ```
  switch(config-acl)# no permit tcp 10.0.0.0/8 any
  ```

  However, if the same rule had a sequence number of 101, removing the rule requires only the following command:

  ```
  switch(config-acl)# no 101
  ```

- Moving a rule—With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule using the sequence number that positions it correctly, and then you can remove the original instance of the rule. This action allows you to move the rule without disrupting traffic.

If you enter a rule without a sequence number, the switch adds the rule to the end of the ACL and assigns a sequence number that is 10 greater than the sequence number of the preceding rule to the rule. For example, if the last rule in an ACL has a sequence number of 225 and you add a rule without a sequence number, the switch assigns the sequence number 235 to the new rule.

In addition, the switch allows you to reassign sequence numbers to rules in an ACL. Resequencing is useful when an ACL has rules numbered contiguously, such as 100 and 101, and you need to insert one or more rules between those rules.

Logical Operators and Logical Operation Units

IP ACL rules for TCP and UDP traffic can use logical operators to filter traffic based on port numbers.

The switch stores operator-operand couples in registers called logical operator units (LOUs).

LOU usage for the eq operator is never stored in an LOU. The range operation is inclusive of boundary values.

The following guidelines determine when the switch stores operator-operand couples in LOUs:

- If the operator or operand differs from other operator-operand couples that are used in other rules, the couple is stored in an LOU.

  For example, the operator-operand couples gt 10 and gt 11 would be stored separately in half an LOU each. The couples gt 10 and lt 10 would also be stored separately.

- Whether the operator-operand couple is applied to a source port or a destination port in the rule affects LOU usage. Identical couples are stored separately when one of the identical couples is applied to a source port and the other couple is applied to a destination port.

  For example, if a rule applies the operator-operand couple gt 10 to a source port and another rule applies a gt 10 couple to a destination port, both couples would also be stored in half an LOU, resulting in the use of one whole LOU. Any additional rules using a gt 10 couple would not result in further LOU usage.

Configuring IPv4 ACLs

This section includes the following topics:

Creating an IPv4 ACL

You can create an IPv4 ACL on the switch and add rules to it. To create an IP ACL, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# ip access-list` *name*	Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	`switch(config-acl)# [`*sequence-number*`] {permit|deny}` *protocol source destination*	Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 4000I and 4005I Switch Module for IBM BladeCenter NX-OS Command Reference*.
Step 4	`switch(config-acl)# statistics per-entry`	(Optional) Specifies that the switch maintains global statistics for packets matching the rules in the ACL.
Step 5	`switch(config-acl)# show ip access-lists` *name*	(Optional) Displays the IP ACL configuration.
Step 6	`switch(config-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to create an IPv4 ACL:

```
switch# configure terminal
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.0.2.0/24 any
switch(config-acl)# statistics per-entry
switch(config-acl)# show ip access-lists acl-01
switch(config-acl)# copy running-config startup-config
```

Changing an IP ACL

You can add and remove rules in an existing IPv4 ACL. You cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers. For more information, see the "Changing Sequence Numbers in an IP ACL" section on page 21-7.

To change an IP ACL, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# ip access-list` *name*	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 3	`switch(config-acl)# [`*sequence-number*`] {permit|deny}` *protocol source destination*	(Optional) Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 4000I and 4005I Switch Module for IBM BladeCenter NX-OS Command Reference*.
Step 4	`switch(config-acl)# no {`*sequence-number* `| {permit|deny}` *protocol source destination*`}`	(Optional) Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Cisco Nexus 4000I and 4005I Switch Module for IBM BladeCenter NX-OS Command Reference*.
Step 5	`switch(config-acl)# [no] statistics`	(Optional) Specifies that the switch maintains global statistics for packets matching the rules in the ACL. The **no** option stops the switch from maintaining global statistics for the ACL.
Step 6	`switch(config-acl)# show ip access-lists` *name*	(Optional) Displays the IP ACL configuration.
Step 7	`switch(config-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Removing an IP ACL

You can remove an IP ACL from the switch.

Before you remove an IP ACL from the switch, be sure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the switch considers the removed ACL to be empty.

To remove an IP ACL from the switch, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# no ip access-list` *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 3	`switch(config)# show running-config`	(Optional) Displays ACL configuration. The removed IP ACL should not appear.
Step 4	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing Sequence Numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL. To change sequence numbers, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# resequence ip access-list` *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the *increment* argument can be a whole number between 1 and 4294967295.
Step 3	`switch(config)# show ip access-lists` *name*	(Optional) Displays the IP ACL configuration.
Step 4	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Applying an IP ACL as a Port ACL

You can apply an IPv4 ACL to a physical Ethernet interface or a port channel. ACLs applied to these interface types are considered port ACLs. To apply an IP ACL, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# interface ethernet` *slot/port*	Enters interface configuration mode for the specified interface.
	`switch(config)# interface port-channel` *channel-number*	Enters interface configuration mode for a port channel.

	Command	Purpose
Step 3	`switch(config-if)# [ip | mac] port access-group access-list in`	Applies an IPv4 ACL to the interface or port channel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.
Step 4	`switch(config-if)# show running-config`	(Optional) Displays ACL configuration.
Step 5	`switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to apply an IPv4 ACL to the port channel:

```
switch# configure terminal
switch(config)# interface port-channel 5
switch(config-if)# ip port access-group acl-l2-marketing-group in
switch(config-if)# show running-config
switch(config-if)# copy running-config startup-config
```

The following example shows how to create an IPv4 ACL named acl-01 and apply it to Ethernet interface 1/1, which is a Layer 2 interface:

```
ip access-list acl-01
  permit ip 192.168.2.0/24 any
interface ethernet 1/1
  ip access-group acl-01 in
```

Applying an IP ACL as a VACL

For information about configuring VACLs, see "Configuring VACLs" section on page 21-15.

Verifying IP ACL Configurations

To display IP ACL configuration information, perform one of these tasks:

Command	Purpose
`switch# show running-config`	Displays ACL configuration, including IP ACL configuration and interfaces that IP ACLs are applied to.
`switch# show ip access-lists`	Displays the IP ACL configuration.
`switch# show running-config interface`	Displays the configuration of an interface to which you have applied an ACL.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 4000I and 4005I Switch Module for IBM BladeCenter NX-OS Command Reference*.

Displaying and Clearing IP ACL Statistics

Use the **show ip access-lists** command to display statistics about an IP ACL, including the number of packets that have matched each rule. For detailed information about the fields in the output from this command, see the *Cisco Nexus 4000I and 4005I Switch Module for IBM BladeCenter NX-OS Command Reference*.

Note The mac access-list is applicable to non-IPv4 traffic only.

To display or clear VACL statistics, perform one of these tasks:

Command	Purpose
`switch# show ip access-lists`	Displays IP ACL configuration. If the IP ACL includes the **statistics** command, then the **show ip access-lists** command output includes the number of packets that have matched each rule.
`switch# clear ip access-list counters`	Clears statistics for all IP ACLs or for a specific IP ACL.

For detailed information about these commands, see the *Cisco Nexus 4000I and 4005I Switch Module for IBM BladeCenter NX-OS Command Reference*.

Configuring MAC ACLs

This section includes the following topics:

Creating a MAC ACL

To create a MAC ACL and add rules to it, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch# mac access-list` *name*	Creates the MAC ACL and enters ACL configuration mode.
Step 3	`switch(config-mac-acl)# [sequence number] {permit | deny}` *source destination protocol*	Creates a rule in the MAC ACL. The **permit** and **deny** options support many ways of identifying traffic. For more information, see the *Cisco Nexus 4000I and 4005I Switch Module for IBM BladeCenter NX-OS Command Reference*.
Step 4	`switch(config-mac-acl)# statistics per-entry`	(Optional) Specifies that the switch maintains global statistics for packets matching the rules in the ACL.
Step 5	`switch(config-mac-acl)# show mac access-lists` *name*	(Optional) Displays the MAC ACL configuration.
Step 6	`switch(config-mac-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to create a MAC ACL and add rules to it:

```
switch# configure terminal
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config-mac-acl)# statistics per-entry
switch(config-mac-acl)# show mac access-lists acl-mac-01
switch(config-mac-acl)# copy running-config startup-config
```

Changing a MAC ACL

In an existing MAC ACL, you can add and remove rules. You cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers. For more information, see the "Changing Sequence Numbers in an IP ACL" section on page 21-7.

To change a MAC ACL, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# mac access-list` *name*	Enters ACL configuration mode for the ACL that you specify by name.

	Command	Purpose
Step 3	`switch(config-mac-acl)# [sequence-number] {permit|deny} source destination protocol`	(Optional) Creates a rule in the MAC ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The **permit** and **deny** commands support many ways of identifying traffic.
Step 4	`switch(config-mac-acl)# no {sequence-number | {permit|deny} protocol source destination}`	(Optional) Removes the rule that you specify from the MAC ACL. The **permit** and **deny** commands support many ways of identifying traffic.
Step 5	`switch(config-mac-acl)# [no] statistics per-entry`	(Optional) Specifies that the switch maintains global statistics for packets matching the rules in the ACL. The **no** option stops the switch from maintaining global statistics for the ACL.
Step 6	`switch(config-mac-acl)# show mac access-lists name`	(Optional) Displays the MAC ACL configuration.
Step 7	`switch(config-mac-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to change a MAC ACL:

```
switch# configure terminal
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# 100 permit mac 00c0.4f00.00 0000.00ff.ffff any
switch(config-mac-acl)# statistics per-entry
switch(config-mac-acl)# show mac access-lists acl-mac-01
switch(config-mac-acl)# copy running-config startup-config
```

Removing a MAC ACL

You can remove a MAC ACL from the switch.

Be sure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are current applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the switch considers the removed ACL to be empty.

To remove a MAC ACL, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# no mac access-list` *name*	Removes the MAC ACL that you specify by name from the running configuration.
Step 3	`switch(config)# show mac access-lists`	(Optional) Displays the MAC ACL configuration.
Step 4	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing Sequence Numbers in a MAC ACL

You can change all the sequence numbers assigned to rules in a MAC ACL. Resequencing is useful when you need to insert rules into an ACL and there are not enough available sequence numbers. For more information, see the "Rules" section on page 21-2.

To change all the sequence numbers assigned to rules in a MAC ACL, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# resequence mac access-list` *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the number specified by the starting-sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment number that you specify.
Step 3	`switch(config)# show mac access-lists` *name*	(Optional) Displays the MAC ACL configuration.
Step 4	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Applying a MAC ACL as a Port ACL

You can apply a MAC ACL as a port ACL to any of the following interface types:

- Layer 2 interfaces
- Port-channel interfaces

Be sure that the ACL that you want to apply exists and is configured to filter traffic as necessary for this application. For more information about configuring MAC ACLs, see the "Configuring IPv4 ACLs" section on page 21-4.

To apply a MAC ACL as a port ACL, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# interface ethernet`*slot/port*	Enters interface configuration mode for the specified interface.
	`switch(config)# interface port-channel` *channel-number*}	Enters interface configuration mode for a port-channel interface.
Step 3	`switch(config-if)# [ip | mac] port access-group` *access-list*	Applies a MAC ACL to the interface.
Step 4	`switch(config-if)# show running-config`	(Optional) Displays ACL configuration.
Step 5	`switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Applying a MAC ACL as a VACL

You can apply a MAC ACL as a VACL. For information about how to create a VACL using a MAC ACL, see the "Creating or Changing a VACL" section on page 21-15.

Verifying MAC ACL Configurations

To display MAC ACL configuration information, perform one of these tasks:

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration
show running-config	Displays ACL configuration, including MAC ACLs and the interfaces that ACLs are applied to.
show running-config interface	Displays the configuration of the interface to which you applied the ACL.

Displaying and Clearing MAC ACL Statistics

Use the **show mac access-lists** command to display statistics about a MAC ACL, including the number of packets that have matched each rule.

To display or clear MAC ACL statistics, perform one of these tasks:

Command	Purpose
show mac access-lists	Displays MAC ACL configuration. If the MAC ACL includes the **statistics** command, the **show mac access-lists** command output includes the number of packets that have matched each rule.
clear mac access-list counters	Clears statistics for all MAC ACLs or for a specific MAC ACL.

The following example shows how to create a MAC ACL named acl-mac-01 and apply it to Ethernet interface 2/1, which is a Layer 2 interface:

```
mac access-list acl-mac-01
  permit 00c0.4f00.0000 0000.00ff.ffff any
interface ethernet 2/1
  mac access-group acl-mac-01
```

Information About VLAN ACLs

A VLAN ACL (VACL) is one application of a MAC ACL or IP ACL. You can configure VACLs to apply to all packets that are bridged within a VLAN. VACLs are used strictly for security packet filtering. VACLs are not defined by direction (ingress or egress).

For more information about types and applications of ACLs, see the "Information About ACLs" section on page 21-1.

This section includes the following topics:

VACLs and Access Maps

VACLs use access maps to link an IP ACL or a MAC ACL to an action. The switch takes the configured action on packets permitted by the VACL.

VACLs and Actions

In access map configuration mode, you use the **action** command to specify one of the following actions:

- Forward—Sends the traffic to the destination determined by normal operation of the switch.
- Drop—Drops the traffic.

Statistics

The switch can maintain global statistics for each rule in a VACL. If a VACL is applied to multiple VLANs, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that VACL is applied.

Note The switch does not support interface-level VACL statistics.

For each VLAN access map that you configure, you can specify whether the switch maintains statistics for that VACL. This allows you to turn VACL statistics on or off as needed to monitor traffic filtered by a VACL or to help troubleshoot VLAN access-map configuration.

For information about displaying VACL statistics, see the "Displaying and Clearing IP ACL Statistics" section on page 21-9.

Configuring VACLs

This section includes the following topics:

Creating or Changing a VACL

You can create or change a VACL. Creating a VACL includes creating an access map that associates an IP ACL or MAC ACL with an action to be applied to the matching traffic.

To create or change a VACL, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# vlan access-map` *map-name* `[`*sequence number*`]`	Enters access map configuration mode for the access map specified.
Step 3	`switch(config-access-map)# match ip address` *ip-access-list*	Specifies an IPv4 ACL for the map.
	`switch(config-access-map)# match mac address` *mac-access-list*	Specifies a MAC ACL for the map.

	Command	Purpose
Step 4	`switch(config-access-map)# action {drop|forward|redirect}`	Specifies the action that the switch applies to traffic that matches the ACL.
Step 5	`switch(config-access-map)# [no] statistics`	(Optional) Specifies that the switch maintains global statistics for packets matching the rules in the VACL. The **no** option stops the switch from maintaining global statistics for the VACL.
Step 6	`switch(config-access-map)# show running-config`	(Optional) Displays ACL configuration.
Step 7	`switch(config-access-map)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Removing a VACL

You can remove a VACL, which means that you will delete the VLAN access map.

Be sure that you know whether the VACL is applied to a VLAN. The switch allows you to remove VACLs that are current applied. Removing a VACL does not affect the configuration of VLANs where you have applied the VACL. Instead, the switch considers the removed VACL to be empty.

To remove a VACL, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# no vlan access-map` *map-name*	Removes the VLAN access map configuration for the specified access map.
Step 3	`switch(config)# show running-config`	(Optional) Displays ACL configuration.
Step 4	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Applying a VACL to a VLAN

You can apply a VACL to a VLAN. The VACL drop-down list appears in the Advanced Settings area.

To apply a VACL to a VLAN, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# [no] vlan filter` *map-name* `vlan-list` *list*	Applies the VACL to the VLANs by the list that you specified. The **no** option unapplies the VACL. The **vlan-list** command can specify a list of up to 32 VLANs, but multiple **vlan-list** commands can be configured to cover more than 32 VLANs.
Step 3	`switch(config)# show running-config`	(Optional) Displays ACL configuration.
Step 4	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying VACL Configuration

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

Command	Purpose
`switch# show running-config aclmgr`	Displays ACL configuration, including VACL-related configuration.
`switch# show vlan filter`	Displays information about VACLs that are applied to a VLAN.
`switch# show vlan access-map`	Displays information about VLAN access maps.

Displaying and Clearing VACL Statistics

To display or clear VACL statistics, perform one of these tasks:

Command	Purpose
`switch# show vlan access-list`	Displays VACL configuration. If the VLAN access-map includes the **statistics** command, then the **show vlan access-list** command output includes the number of packets that have matched each rule.
`switch# clear vlan access-list counters`	Clears statistics for all VACLs or for a specific VACL.

The following example shows how to configure a VACL to forward traffic permitted by an IP ACL named acl-01 and how to apply the VACL to VLANs 50 through 82:

```
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.0.0/16 any
switch(config-acl)# statistics per-entry
switch(config-acl)# show ip access-lists acl-01
IP access list acl-01
        statistics per-entry
        10 permit ip 192.168.0.0/16 any
switch(config-acl)# copy running-config startup-config
```

```
[########################################] 100%
switch(config-acl)# exit
switch(config)# vlan access-map acl-ip-map 40
switch(config-access-map)# match ip address acl-01
switch(config-access-map)# action forward
switch(config-access-map)# vlan filter acl-ip-map vlan-list 50-82
```

Default Settings

Table 21-2 lists the default settings for IP ACLs parameters.

Table 21-2 Default IP ACLs Parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default.
ACL rules	Implicit rules apply to all ACLs. See the "Implicit Rules" section on page 21-3.

Table 21-3 lists the default settings for MAC ACLs parameters.

Table 21-3 Default MAC ACLs Parameters

Parameters	Default
MAC ACLs	No MAC ACLs exist by default.
ACL rules	Implicit rules apply to all ACLs. See the "Implicit Rules" section on page 21-3.

Table 21-4 lists the default settings for VACL parameters.

Table 21-4 Default VACL Parameters

Parameters	Default
VACLs	No IP ACLs exist by default.
ACL rules	Implicit rules apply to all ACLs. See the "Implicit Rules" section on page 21-3.

PART 4

System Management

CHAPTER 22

Configuring User Accounts and RBAC

This chapter describes how to configure user accounts and role-based access control (RBAC) on the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

This chapter includes the following sections:

Information About User Accounts and RBAC

You can create and manage users accounts and assign roles that limit access to operations on the switch. RBAC allows you to define the rules for an assign role that restrict the authorization that the user has to access management operations.

This section includes the following topics:

About User Accounts

Tip The following words are reserved and cannot be used to configure users: bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, games, gopher, ftp, nobody, nscd, mailnull, rpc, rpcuser, xfs, gdm, mtsuser, ftpuser, man, and sys.

Note User passwords are not displayed in the configuration files.

Caution The switch does not support all numeric usernames, whether created with TACACS+ or RADIUS, or created locally. Local users with all numeric names cannot be created. If an all numeric username exists on a AAA server and is entered during login, the user is not logged in.

Characteristics of Strong Passwords

A strong password has the following characteristics:

- At least eight characters long
- Does not contain many consecutive characters (such as “abcd”)
- Does not contain many repeating characters (such as “aaabbb”)
- Does not contain dictionary words
- Does not contain proper names
- Must contain at least three of the following classes: lower case letters, upper case letters, digits, and special characters.

The following are examples of strong passwords:

- If2CoM18
- 2004AsdfLkj30
- Cb1955S21

Tip If a password is trivial (such as a short, easy-to-decipher password), the switch will reject your password configuration. Be sure to configure a strong password as shown in the sample configuration. Passwords are case sensitive.
Clear text passwords cannot contain dollar signs ($) or spaces anywhere in the password. Also, they cannot include these special characters at the beginning of the password: quotation marks (" or '), vertical bars (|), or right angle brackets (>).

About User Roles

User roles contain rules that define the operations allowed for the user who is assigned the role. Each user role can contain multiple rules and each user can have multiple roles. For example, if role1 allows access only to configuration operations, and role2 allows access only to debug operations, then users who belong to both role1 and role2 can access configuration and debug operations. You can also limit access to specific VLANs and interfaces.

The switch provides the following default user roles:

- network-admin (superuser)—Complete read and write access to the entire switch.
- network-operator—Complete read access to the switch.

Note If you belong to multiple roles, you can execute a combination of all the commands permitted by these roles. Access to a command takes priority over being denied access to a command. For example, suppose a user has RoleA, which denied access to the configuration commands. However, the users also has RoleB, which has access to the configuration commands. In this case, the users has access to the configuration commands.

About Rules

The rule is the basic element of a role. A rule defines what operations the role allows the user to perform. You can apply rules for the following parameters:

- Command—A command or group of commands defined in a regular expression.
- Feature—Commands that apply to a function provided by the switch.
  - Enter the **show role feature** command to display the feature names available for this parameter.
- Feature group—Default or user-defined group of features.
  - Enter the **show role feature-group** command to display the default feature groups available for this parameter.

These parameters create a hierarchical relationship. The most basic control parameter is the command. The next control parameter is the feature, which represents all commands associated with the feature. The last control parameter is the feature group. The feature group combines related features and allows you to easily manage of the rules.

You can configure up to 256 rules for each role. The user-specified rule number determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

About User Role Policies

You can define user role policies to limit the switch resources that the user can access. You can define user role policies to limit access to interfaces, and VLANs.

User role policies are constrained by the rules defined for the role. For example, if you define an interface policy to permit access to specific interfaces, the user will not have access to the interfaces unless you configure a command rule for the role to permit the interface command. The "Changing User Role Interface Policies" section on page 22-7 contains an example configuration.

If a command rule permits access to specific resources (interfaces, or VLANs), the user is permitted to access these resources, even if they are not listed in the user role policies associated with that user.

Guidelines and Limitations

User account and RBAC have the following configuration guidelines and limitations:

- You can add up to 256 rules to a user role.
- You can assign a maximum of 64 user roles to a user account.

Note A user account must have at least one user role.

Configuring User Accounts

You can create a maximum of 256 user accounts on a switch. User accounts have the following attributes:

- Username
- Password
- Expiry date
- User roles

User accounts can have a maximum of 64 user roles. For more information on user roles, see the "Configuring RBAC" section on page 22-5.

Note Changes to user account attributes do not take effect until the user logs in and creates a new session.

To configure a user account, perform this task:

	Command	Purpose
Step 1	`switch# show role`	(Optional) Displays the user roles available. You can configure other user roles, if necessary (see the "Creating User Roles and Rules" section on page 22-5).
Step 2	`switch# configure terminal`	Enters configuration mode.
Step 3	`switch(config)# username` *user-id* `[password` *password*`] [expire` *date*`] [role` *role-name*`]`	Configure a user account. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters. The default password is undefined. **Note** If you do not specify a password, the user might not be able to log in to the switch. The **expire** *date* option format is YYYY-MM-DD. The default is no expiry date.
Step 4	`switch(config)# exit`	Exits global configuration mode.
Step 5	`switch# show user-account`	(Optional) Displays the role configuration.
Step 6	`switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure a user account:

```
switch# configure terminal
switch(config)# username NewUser password 4Ty18Rnt
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

Configuring RBAC

This section includes the following topics:

Creating User Roles and Rules

Each user role can have up to 256 rules. You can assign a user role to more that one user account.

The rule number you specify determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

To create user roles and specify rules, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# role name role-name`	Specifies a user role and enters role configuration mode. The *role-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 16 characters.
Step 3	`switch(config-role)# rule number {deny | permit} command command-string`	Configures a command rule. The *command-string* argument can contain spaces and regular expressions. For example, "interface ethernet *" includes all Ethernet interfaces. Repeat this command for as many rules as needed.
	`switch(config-role)# rule number {deny | permit} {read | read-write}`	Configures a read only or read and write rule for all operations.
	`switch(config-role)# rule number {deny | permit} {read | read-write} feature feature-name`	Configures a read-only or read-and-write rule for a feature. Use the **show role feature** command to display a list of features. Repeat this command for as many rules as needed.
	`switch(config-role)# rule number {deny | permit} {read | read-write} feature-group group-name`	Configures a read-only or read-and-write rule for a feature group. Use the **show role feature-group** command to display a list of feature groups. Repeat this command for as many rules as needed.
Step 4	`switch(config-role)# description text`	(Optional) Configures the role description. You can include spaces in the description.
Step 5	`switch(config-role)# exit`	Exits role configuration mode.

	Command	Purpose
Step 6	`switch(config)# show role`	(Optional) Displays the user role configuration.
Step 7	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to create user roles and specify rules:

```
switch# config terminal
switch(config-role)# rule 1 deny command clear users
switch(config-role)# rule 2 deny read-write
switch(config-role)# rule 3 permit command config t
switch(config-role)# description This role does not allow users to use clear commands
switch(config-role)# exit
switch(config)# show role

Role: network-admin
  Description: Predefined network admin role has access to all commands
  on the switch
  ------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  ------------------------------------------------------------------
  1       permit  read-write

Role: network-operator
  Description: Predefined network operator role has access to all read
  commands on the switch
  ------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  ------------------------------------------------------------------
  1       permit  read

Role: user1
  Description: This role does not allow users to use clear commands
  vsan policy: permit (default)
  Vlan policy: permit (default)
  Interface policy: permit (default)
  Vrf policy: permit (default)
  ------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  ------------------------------------------------------------------
  3       permit  command                         config t
  2       deny    read-write
  1       deny    command                         clar suers
switch(config)# copy running-config startup-config
[########################################] 100%
switch(config)#
```

Creating Feature Groups

To create feature groups, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# role feature-group name group-name`	Specifies a user role feature group and enters role feature group configuration mode. The *group-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 32 characters.
Step 3	`switch(config-role-featuregrp)# exit`	Exits role feature group configuration mode.
Step 4	`switch(config)# show role feature-group`	(Optional) Displays the role feature group configuration.
Step 5	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Changing User Role Interface Policies

You can change a user role interface policy to limit the interfaces that the user can access. To change a user role interface policy, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# role name role-name`	Specifies a user role and enters role configuration mode.
Step 3	`switch(config-role)# rule number permit command configure terminal ; interface *`	Configures a command rule to allow access to all interfaces.
Step 4	`switch(config-role)# interface policy deny`	Enters role interface policy configuration mode.
Step 5	`switch(config-role-interface)# permit interface interface-list`	Specifies a list of interfaces that the role can access. Repeat this command for as many interfaces as needed. For this command, you can specify Ethernet interfaces.
Step 6	`switch(config-role-interface)# exit`	Exits role interface policy configuration mode.
Step 7	`switch(config-role)# show role`	(Optional) Displays the role configuration.
Step 8	`switch(config-role)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

You can specify a list of interfaces that the role can access. You can specify it for as many interfaces as needed:

```
switch(config-role-interface)# permit interface ethernet 1/1
```

Changing User Role VLAN Policies

You can change a user role VLAN policy to limit the VLANs that the user can access. To change a user role VLAN policy, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# role name` *role-name*	Specifies a user role and enters role configuration mode.
Step 3	`switch(config-role)# rule` *number* `permit command configure terminal ; vlan *`	Configures a command rule to allow access to all VLANs.
Step 4	`switch(config-role)# vlan policy deny`	Enters role VLAN policy configuration mode.
Step 5	`switch(config-role-vlan)# permit vlan` *vlan-list*	Specifies a range of VLANs that the role can access. Repeat this command for as many VLANs as needed.
Step 6	`switch(config-role-vlan)# exit`	Exits role VLAN policy configuration mode.
Step 7	`switch(config-role)# show role`	(Optional) Displays the role configuration.
Step 8	`switch(config-role)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying User Accounts and RBAC Configuration

To display user account and RBAC configuration information, perform one of these tasks:

Command	Purpose
`switch# show role`	Displays the user role configuration
`switch# show role feature`	Displays the feature list.
`switch# show role feature-group`	Displays the feature group configuration.
`switch# show startup-config security`	Displays the user account configuration in the startup configuration.
`switch# show running-config security [all]`	Displays the user account configuration in the running configuration. The **all** keyword displays the default values for the user accounts.
`switch# show user-account`	Displays user account information.

Example User Accounts and RBAC Configuration

The following example shows how to configure a user role:

```
switch(config)# role name UserA
switch(config-role)# rule 3 permit command configure terminal ; vlan *
switch(config-role)# rule 2 permit read feature tacacs
switch(config-role)# rule 1 deny command clear *
switch(config-role)# exit
```

The following example shows how to configure a user role feature group:

```
switch(config)# role feature-group name Security-features
switch(config-role-featuregrp)# feature radius
switch(config-role-featuregrp)# feature tacacs
switch(config-role-featuregrp)# feature aaa
```

Default Settings

Table 22-1 lists the default settings for user accounts and RBAC parameters.

Table 22-1 Default User Accounts and RBAC Parameters

Parameters	Default
User account password	Undefined.
User account expiry date.	None.
Interface policy	All interfaces are accessible.
VLAN policy	All VLANs are accessible.

CHAPTER 23

Configuring Session Manager

This chapter describes how to configure the Session Manager features in Cisco NX-OS.

This chapter includes the following sections:

Information About Session Manager

Session Manager allows you to implement your configuration changes in batch mode. Session Manager works in the following phases:

- Configuration session—Creates a list of commands that you want to implement in session manager mode.
- Validation—Provides a basic semantic check on your configuration. Cisco NX-OS returns an error if the semantic check fails on any part of the configuration.
- Verification—Verifies the configuration as a whole, based on the existing hardware and software configuration and resources. Cisco NX-OS returns an error if the configuration does not pass this verification phase.
- Commit—Cisco NX-OS verifies the complete configuration and implements the changes atomically to the device. If a failure occurs, Cisco NX-OS reverts to the original configuration.
- Abort—Discards the configuration changes before implementation.

You can optionally end a configuration session without committing the changes. You can also save a configuration session.

Configuration Guidelines and Limitations

Session Manager has the following configuration guidelines and limitations:

- Session Manager supports only the ACL feature.
- You can create up to 32 configuration sessions.
- You can configure a maximum of 20,000 commands across all sessions.

Configuring Session Manager

This section includes the following topics:

Creating a Session

You can create up to 32 configuration sessions. To create a configuration session, perform this task:

	Command	Purpose
Step 1	`switch# configure session name`	Creates a configuration session and enters session configuration mode. The name can be any alphanumeric string.
Step 2	`switch(config-s)# show configuration session [name]`	(Optional) Displays the contents of the session.
Step 3	`switch(config-s)# save location`	(Optional) Saves the session to a file. The location can be in bootflash or volatile.

Configuring ACLs in a Session

You can configure ACLs within a configuration session. To configure ACLs within a configuration session, perform this task:

	Command	Purpose
Step 1	`switch# configure session name`	Creates a configuration session and enters session configuration mode. The name can be any alphanumeric string.
Step 2	`switch(config-s)# ip access-list name`	Creates an ACL.
Step 3	`switch(config-s-acl)# permit protocol source destination`	(Optional) Adds a permit statement to the ACL.
Step 4	`switch(config-s-acl)# interface interface-type number`	Enters interface configuration mode.
Step 5	`switch(config-s-if)# ip port access-group name in`	Adds a port access group to the interface.
Step 6	`switch# show configuration session [name]`	(Optional) Displays the contents of the session.

Verifying a Session

To verify a session while in session mode, perform this task:

Command	Purpose
`switch(config-s)# verify [verbose]`	Verifies the commands in the configuration session.

Committing a Session

To commit a session while in session mode, perform this task:

Command	Purpose
`switch(config-s)# commit [verbose]`	Commits the commands in the configuration session.

Saving a Session

To save a session while in session mode, perform this task:

Command	Purpose
`switch(config-s)# save location`	(Optional) Saves the session to a file. The location can be in bootflash or volatile.

Discarding a Session

To discard a session while in session mode, perform this task:

Command	Purpose
`switch(config-s)# abort`	Discards the configuration session without applying the commands.

Session Manager Example Configuration

The following example shows how to create a configuration session for ACLs:

```
switch# configure session test1
Config Session started, Session ID is 1
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config-s)# ip access-list acl1
switch(config-s-acl)# permit tcp any any
switch(config-s-acl)# exit
switch(config-s)# interface ethernet 1/20
switch(config-s-if)# ip port access-group acl1 in
switch(config-s-if)# exit
switch(config-s)# verify
Verification Successful
```

```
switch(config-s)# exit
```

Verifying Session Manager Configuration

To verify Session Manager configuration information, perform on of these tasks:

Command	Purpose
`switch# show configuration session [name]`	Displays the contents of the configuration session.
`switch# show configuration session status [name]`	Displays the status of the configuration session.
`switch# show configuration session summary`	Displays a summary of all the configuration sessions.

The following example shows how to verify a configuration session for ACLs:

```
switch# show configuration session test1
config session name test1
0256  ip access-list acl1
0512  permit tcp any any
0768  interface Ethernet1/20
1024  ip port access-group acl1 in
switch#
```

CHAPTER 24

Configuring Online Diagnostics

This chapter describes how to configure the online diagnostics feature.

This chapter includes the following sections:

Online Health Management System

The Online Health Management System (OHMS) is a hardware fault detection and recovery feature. It ensures the general health of the switch.

This section includes the following topics:

The OHMS monitors system hardware in the following ways:

The OHMS application launches a daemon process in the switch and runs multiple tests. The tests run at preconfigured intervals, cover all major fault points, and isolate any failing component in the MDS switch. The OHMS maintains control over all other OHMS components in the switch.

On detecting a fault, the system health application attempts the following recovery actions:

- Performs additional testing to isolate the faulty component
- If unable to recover, sends Call Home notifications, system messages and exception logs; and shuts down and discontinues testing the failed component (such as an ethernet interface)
- Sends Call Home and system messages and exception logs as soon as it detects a failure.
- Shuts down the failing component (such as an ethernet interface).
- Isolates failed ports from further testing.

- Reports the failure to the appropriate software component.
- Provides CLI support to view, test, and obtain test run statistics or change the system health test configuration on the switch.
- Performs tests to focus on the problem area.

The switch is configured to run the relevant test. You can change the default parameters of the test as required.

System Health Initiation

By default, the system health feature is enabled in the switch.

To disable or enable this feature in the switch, perform this task:

	Command	Purpose
Step 1	`switch# config terminal` `switch(config)#`	Enters configuration mode.
Step 2	`switch(config)# no system health` `System Health is disabled.`	Disables system health from running tests in this switch.
	`switch(config)# system health` `System Health is enabled.`	Enables (default) system health to run tests in this switch.
Step 3	`switch(config)# no system health interface ethernet 1/1` `System health for interface ethernet1/1 is disabled.`	Disables system health from testing the Ethernet interface.

Loopback Test Configuration Frequency

Loopback tests are designed to identify hardware errors in the data path in the module. One loopback frame is sent to each module at a preconfigured frequency—it passes through the Ethernet interface.

The loopback tests can be run at frequencies ranging from 60 seconds (default) to 255 seconds. If you do not configure the loopback frequency value, the default frequency of 60 seconds is used for the switch. Loopback test frequencies can be altered for the switch.

To configure the frequency of loopback tests on a switch, perform this task:

	Command	Purpose
Step 1	`switch# config terminal` `switch(config)#`	Enters configuration mode.
Step 2	`switch(config)# system health loopback frequency 60` `The new frequency is set at 60 Seconds.`	Configures the loopback frequency to 60 seconds. The default loopback frequency is 60 seconds. The valid range is from 60 to 255 seconds.

Hardware Failure Action

The **failure-action** command controls the Cisco NX-OS software from taking any action if a hardware failure is determined while running the tests.

By default, this feature is enabled in the switch—action is taken if a failure is determined and the failed component is isolated from further testing.

Failure action is controlled for the entire switch.

To configure failure action in a switch, perform this task:

	Command	Purpose
Step 1	`switch# config terminal` `switch(config)#`	Enters configuration mode.
Step 2	`switch(config)# system health failure-action` `System health global failure action is now enabled.`	Enables the switch to take failure action (default).
Step 3	`switch(config)# no system health failure-action` `System health global failure action now disabled.`	Reverts the switch configuration to prevent failure action being taken.
Step 4	`switch(config)# system health module 1 failure-action` `System health failure action for module 1 is now enabled.`	Enables switch to take failure action for failures in module 1.
Step 5	`switch(config)# no system health module 1 loopback failure-action` `System health failure action for module 1 loopback test is now disabled.`	Prevents the switch from taking action on failures determined by the loopback test in module 1.

Test Run Requirements

Enabling a test does not guarantee that a test will run.

Tests on a given interface or module only run if you enable system health for all of the following items:

- The entire switch.
- The required module.
- The required interface.

Tip The test will not run if system health is disabled in any combination. If system health is disabled to run tests, the test status shows up as disabled.

Tip If the switch or Ethernet interface is enabled to run tests, but is not running the tests due to system health being disabled, then tests show up as enabled (not running).

Tests for a Specified Module

The system health feature in the NX-OS software performs tests in the following areas:

- Bootflash connectivity and accessibility on the switch.
- Data path integrity for each interface on the switch.
- Management port connectivity.
- User-driven test for internal connectivity verification (Ethernet ports).

To perform the required test on a specific module, perform this task:

	Command	Purpose
Step 1	`switch# config terminal` `switch(config)#`	Enters configuration mode.
	Note The following steps can be performed in any order.	
	Note The various options for each test are described in the next step. Each command can be configured in any order. The various options are presented in the same step for documentation purposes.	
Step 2	`switch(config)# system health module 1 bootflash`	Enables the bootflash test on the switch.
	`switch(config)# system health module 1 bootflash frequency 200`	Sets the new frequency of the bootflash test on the switch.
Step 3	`switch(config)# system health module 1 loopback`	Enables the loopback test on the switch.
Step 4	`switch(config)# system health module 1 management`	Enables the management test on the switch.

Clearing Previous Error Reports

You can clear the error history for Ethernet interfaces or the switch. By clearing the history, you are directing the software to retest all failed components that were previously excluded from tests.

If you previously enabled the failure-action option for a period of time (for example, one week) to prevent OHMS from taking any action when a failure is encountered and after that week you are now ready to start receiving these errors again, then you must clear the system health error status for each test.

Use the EXEC-level **system health clear-errors** command for the interface or switch to erase any previous error conditions logged by the system health application. The **bootflash**, the **loopback**, and the **mgmt** test options can be individually specified for a given module.

The following example clears the error history for the specified Ethernet interface:

```
switch# system health clear-errors interface ethernet 1/1
```

The following example clears the error history for the specified module:

```
switch# system health clear-errors module 1
```

The following example clears the management test error history for the switch:

```
switch# system health clear-errors module 1 mgmt
```

Interpreting the Current Status

The status of each switch or test depends on the current configured state of the OHMS test (see Table 24-1).

Table 24-1 OHMS Configured Status for Tests and Modules

Status	Description
Enabled	You have currently enabled the test, and the test is not running.
Disabled	You have currently disabled the test.
Running	You have enabled the test and the test is currently running.

Table 24-1 OHMS Configured Status for Tests and Modules (continued)

Status	Description
Failing	This state is displayed if a failure is imminent for the test running—possibility of test recovery exists in this state.
Failed	The test has failed—and the state cannot be recovered.
Stopped	The test has been internally stopped by the Cisco NX-OS software.
Internal failure	The test encountered an internal failure in this module. For example, the system health application is not able to open a socket as part of the test procedure. **Note** The internal failure status does not apply to the loopback test.
On demand	The system health internal-loopback tests are currently running. This command can be issued on demand.

The status of each test in the switch is visible when you display any of the **show system health** commands. See the "Displaying System Health" section on page 24-5.

Displaying System Health

Use the **show system health** command to display system-related status information (see Example 24-1 to Example 24-6).

Example 24-1 Displays the Current Health of All Modules in the Switch

```
switch# show system health

Current health information for module 1.

Test                    Frequency       Status          Action
------------------------------------------------------------------
Bootflash                10 Sec         Running         Enabled
Management Port          60 Sec         Running         Enabled
Loopback                 60 Sec         Running         Enabled
------------------------------------------------------------------
```

Example 24-2 Displays the Current Health of a Specified Module

```
switch# show system health module 1

Current health information for module 1.

Test                    Frequency       Status          Action
------------------------------------------------------------------
Bootflash                10 Sec         Running         Enabled
Management Port          60 Sec         Running         Enabled
Loopback                 60 Sec         Running         Enabled
------------------------------------------------------------------
```

Example 24-3 Displays Health Statistics for All Modules

```
switch# show system health statistics

Test statistics for module 1
```

```
--------------------------------------------------------------------------------
Test Name          State       Frequency     Run       Pass       Fail CFail Errs
--------------------------------------------------------------------------------
Bootflash          Running        10s        705        705          0     0    0
Management Port    Running        60s        117        117          0     0    0
Loopback           Running        60s       1504       1493         11     0    0
Loopback Port    Status
      1          Failed
      2          Failed
      3          Failed
      4          Failed
      5          Failed
      6          Failed
      7          Passed
      8          Passed
      9          Passed
     10          Passed
     11          Passed
     12          Passed
     13          Passed
     14          Passed
     15          Passed
     16          Failed
     17          Failed
     18          Failed
     19          Failed
     20          Failed
--------------------------------------------------------------------------------
```

Example 24-4 Displays Statistics for a Specified Module

```
switch# show system health statistics module 1

Test statistics for module 1
--------------------------------------------------------------------------------
Test Name          State       Frequency     Run       Pass       Fail CFail Errs
--------------------------------------------------------------------------------
Bootflash          Running        10s        706        706          0     0    0
Management Port    Running        60s        117        117          0     0    0
Loopback           Running        60s       1504       1493         11     0    0
Loopback Port    Status
      1          Failed
      2          Failed
      3          Failed
      4          Failed
      5          Failed
      6          Failed
      7          Passed
      8          Passed
      9          Passed
     10          Passed
     11          Passed
     12          Passed
     13          Passed
     14          Passed
     15          Passed
     16          Failed
     17          Failed
     18          Failed
     19          Failed
     20          Failed
--------------------------------------------------------------------------------
```

Example 24-5 Displays Loopback Test Statistics for the Entire Switch

```
switch# show system health statistics loopback
----------------------------------------------------------------
Mod Port Status                Run      Pass      Fail   CFail Errs
  1   20 Running                 0         0         0       0    0
----------------------------------------------------------------
```

Example 24-6 Displays Loopback Test Statistics for a Specified Interface

```
switch# show system health statistics loopback interface ethernet 1/1
----------------------------------------------------------------
Mod Port Status                Run      Pass      Fail   CFail Errs
  1    1 Running                 0         0         0       0    0
----------------------------------------------------------------
```

Note Interface-specific counters will remain at zero unless the loopback test reports errors or failures.

Example 24-7 Displays the Loopback Test Time Log for the Switch

```
switch# show system health statistics loopback timelog
----------------------------------------------------------------
Mod         Samples     Min(usecs)     Max(usecs)     Ave(usecs)
  1               0              0              0              0
----------------------------------------------------------------
```

Example 24-8 Displays the Loopback Test Time Log for a Specified Module

```
switch# show system health statistics loopback module 1 timelog
----------------------------------------------------------------
Mod         Samples     Min(usecs)     Max(usecs)     Ave(usecs)
  1               0              0              0              0
----------------------------------------------------------------
```

On-Board Failure Logging

The on-board failure logging (OBFL) feature stores failure and environmental information in nonvolatile memory on the module. The information will help in post-mortem analysis of failed cards.

This section includes the following topics:

About OBFL

OBFL data is stored in the existing eUSB on the module. OBFL uses the persistent logging (PLOG) facility available in the module firmware to store data in the eUSB. It also provides the mechanism to retrieve the stored data.

The data stored by the OBFL facility includes the following:

- Time of initial power-on
- Firmware, BIOS, FPGA, and ASIC versions
- Serial number of the card
- Stack trace for crashes
- Software error messages
- Hardware exception logs
- Environmental history
- OBFL specific history information

Configuring OBFL for the Switch

To configure OBFL for all the modules on the switch, perform this task:

	Command	Purpose
Step 1	`switch# config terminal` `switch(config)#`	Enters configuration mode.
Step 2	`switch(config)# hw-module logging onboard`	Enables all OBFL features.
	`switch(config)# hw-module logging onboard environmental-history`	Enables the OBFL environmental history.
	`switch(config)# hw-module logging onboard obfl-log`	Enables the boot uptime, device version, and OBFL history.
	`switch(config)# no hw-module logging onboard`	Disables all OBFL features.

Use the **show logging onboard status** command to display the configuration status of OBFL:

```
switch# show logging onboard status
----------------------------
OBFL Status
----------------------------
    Switch OBFL Log:                                    Enabled

    Module:  1 OBFL Log:                                Enabled
    environmental-history                               Enabled
    exception-log                                       Enabled
    obfl-log (boot-uptime/device-version/obfl-history)  Enabled
    temp error                                          Enabled
    stack-trace                                         Enabled
```

Displaying OBFL Logs

To display OBFL information stored in the switch, use the following commands:

Command	Purpose
show logging onboard boot-uptime	Displays the boot and uptime information.
show logging onboard device-version	Displays device version information.
show logging onboard endtime	Displays OBFL logs to an end time.
show logging onboard environmental-history	Displays environmental history.
show logging onboard exception-log	Displays exception log information.
show logging onboard miscellaneous-error	Displays miscellaneous error information.
show logging onboard obfl-history	Displays history information.
show logging onboard stack-trace	Displays kernel stack trace information.
show logging onboard starttime	Displays OBFL logs from a specified start time.
show logging onboard system-health	Displays system health information.

Default Settings

Table 24-2 lists the default system health and log settings.

Table 24-2 Default System Health and Log Settings

Parameters	Default
Kernel core generation	One module.
System health	Enabled.
Loopback frequency	60 seconds.
Failure action	Enabled.

CHAPTER 25

Configuring Call Home

This chapter describes how to configure the Call Home feature. This chapter includes the following sections:

Information About Call Home

Call Home provides e-mail-based notification of critical system events. The Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter provide a range of message formats for optimal compatibility with pager services, standard e-mail, or XML-based automated parsing applications. You can use this feature to page a network support engineer, e-mail a Network Operations Center, or use Cisco Call Home services to automatically generate a case with the Cisco TAC.

This section includes the following topics:

Call Home Overview

You can use Call Home to notify an external entity when an important event occurs on your device. Call Home delivers alerts to multiple recipients that you configure in *destination profiles* (see "Destination Profiles" section on page 25-2).

Call Home includes a fixed set of predefined alerts on your switch. These alerts are grouped into alert groups and CLI commands that are assigned to execute when an alert in an alert group occurs. The switch includes the command output in the transmitted Call Home message. See the "Call Home Alert Groups" section on page 25-2 for a list of alerts and the predefined set of CLI commands sent when the alert triggers.

The Call Home feature offers the following advantages:

- Automatic execution and attachment of relevant CLI command output.
- Multiple message format options such as the following:
  - Short Text—Suitable for pagers or printed reports.
  - Full Text—Fully formatted message information suitable for human reading.
  - XML—Machine readable format that uses Extensible Markup Language (XML) and Adaptive Messaging Language (AML) XML schema definition (XSD). The XML format enables communication with the Cisco TAC.
- Multiple concurrent message destinations. You can configure up to 50 e-mail destination addresses for each destination profile.

Destination Profiles

A destination profile includes the following information:

- One or more alert groups—The group of alerts that trigger a specific Call Home message if the alert occurs.
- One or more e-mail destinations—The list of recipients for the Call Home messages generated by alert groups assigned to this destination profile.
- Message format—The format for the Call Home message (short text, full text, or XML).
- Message severity level—The Call Home severity level that the alert must meet before the switch generates a Call Home message to all e-mail addresses in the destination profile. For more information about Call Home severity levels, see the "Call Home Message Levels" section on page 25-4. The switch does not generate an alert if the Call Home severity level of the alert is lower than the message severity level set for the destination profile.

You can also configure a destination profile to allow periodic inventory update messages by using the inventory alert group that will send out periodic messages daily, weekly, or monthly.

The switch supports the following predefined destination profiles:

- CiscoTAC-1—Supports the Cisco-TAC alert group in XML message format.
- full-text-destination—Supports the full text message format.
- short-text-destination—Supports the short text message format.

See the "Message Formats" section on page 25-14 for more information about the message formats.

Call Home Alert Groups

An alert group is a predefined subset of Call Home alerts that are supported in the switch. Alert groups allow you to select the set of Call Home alerts that you want to send to a predefined or custom destination profile. The switch sends Call Home alerts to e-mail destinations in a destination profile only if that Call

Home alert belongs to one of the alert groups associated with that destination profile and if the alert has a Call Home message severity at or above the message severity set in the destination profile (see the "Call Home Message Levels" section on page 25-4).

Table 25-1 lists supported alert groups and the default CLI command output included in Call Home messages generated for the alert group.

Table 25-1 Alert Groups and Executed Commands

Alert Group	Description	Executed Commands
Cisco-TAC	All critical alerts from the other alert groups destined for Call Home.	Execute commands based on the alert group that originates the alert.
Diagnostic	Events generated by diagnostics.	**show module** **show version** **show diagnostic result module all** **show tech-support platform callhome**
Linecard-hardware/ Supervisor-hardware	Events related to standard or intelligent switching modules.	**show module** **show version** **show diagnostic result module all** **show tech-support platform callhome**
Configuration	Periodic events related to configuration.	**show version** **show module** **show running-config all** **show startup-config**
System	Events generated by failure of a software system that is critical to unit operation.	**show system redundancy status** **show tech-support**
Environmental	Events related to environment-sensing elements such as temperature alarms.	**show module** **show version** **show environment** **show logging logfile**
Syslog-group-port	Events related to syslogs generated (only license)	**show module** **show version** **show license usage** **show inventory** **show sprom all** **show system uptime** **show interface transceiver**
Inventory	Inventory status that is provided whenever a unit is cold booted, or through a CLI trigger. This alert is considered a noncritical event, and the information is used for status and entitlement.	**show module** **show version** **show license usage** **show inventory** **show sprom all** **show system uptime** **show interface transceiver** **show environment**

Call Home maps the syslog severity level to the corresponding Call Home severity level for syslog port group messages (see the "Call Home Message Levels" section on page 25-4).

You can customize predefined alert groups to execute additional CLI **show** commands when specific events occur and send that **show** output with the Call Home message.

You can add **show** commands only to full text and XML destination profiles. Short text destination profiles do not support additional **show** commands because short text Call Home messages do not include **show** commands output.

Call Home Message Levels

Call Home allows you to filter messages based on their level of urgency. You can associate each destination profile (predefined and user defined) with a Call Home message level threshold. The switch does not generate any Call Home messages with a value lower than this threshold for the destination profile. The Call Home message level ranges from 0 (lowest level of urgency) to 9 (highest level of urgency), and the default is 0 (The switch sends all messages).

Call Home messages that are sent for syslog alert groups have the syslog severity level mapped to the Call Home message level.

Note Call Home does not change the syslog message level in the message text.

Table 25-2 lists each Call Home message level keyword and the corresponding syslog level for the syslog port alert group.

Table 25-2 Severity and syslog Level Mapping

Call Home Level	Keyword	syslog Level	Description
9	**Catastrophic**	N/A	Network-wide catastrophic failure.
8	**Disaster**	N/A	Significant network impact.
7	**Fatal**	Emergency (0)	System is unusable.
6	**Critical**	Alert (1)	Critical conditions that indicate that immediate attention is needed.
5	**Major**	Critical (2)	Major conditions.
4	**Minor**	Error (3)	Minor conditions.
3	**Warning**	Warning (4)	Warning conditions.
2	**Notification**	Notice (5)	Basic notification and informational messages. Possibly independently insignificant.
1	**Normal**	Information (6)	Normal event signifying return to normal state.
0	**Debugging**	Debug (7)	Debugging messages.

Obtaining Smart Call Home

If you have a service contract directly with Cisco Systems, you can register your devices for the Smart Call Home service. Smart Call Home provides fast resolution of system problems by analyzing Call Home messages sent from your devices and providing background information and recommendations. For issues that can be identified as known, particularly GOLD diagnostics failures, Automatic Service Requests will be generated with the Cisco TAC.

Smart Call Home offers the following features:

- Continuous device health monitoring and real-time diagnostic alerts.
- Analysis of Call Home messages from your device and, where appropriate, Automatic Service Request generation, routed to the appropriate TAC team, including detailed diagnostic information to speed problem resolution.
- Secure message transport directly from your device or through a downloadable Transport Gateway (TG) aggregation point. You can use a TG aggregation point in cases that require support for multiple devices or in cases where security requirements mandate that your devices may not be connected directly to the Internet.
- Web-based access to Call Home messages and recommendations, inventory and configuration information for all Call Home devices. Provides access to associated field notices, security advisories and end-of-life information.

You need the following items to register:

- The SMARTnet contract number for your switch.
- Your e-mail address
- Your Cisco.com ID

For more information about Smart Call Home, see the Smart Call Home page at this location:

http://www.cisco.com/go/smartcall/

Prerequisites for Call Home

Call Home has the following prerequisites:

- You must configure an e-mail server.
- You must configure the contact name (SNMP server contact), phone, and street address information before you enable Call Home. This step is required to determine the origin of messages received.
- Your switch must have IP connectivity to an e-mail server.
- If you use Call Home, you need an active service contract for the device that you are configuring.

Configuration Guidelines and Limitations

Call Home has the following configuration guidelines and limitations:

- If there is no IP connectivity or if the interface in the VRF to the profile destination is down, the switch cannot send the Call Home message.
- Operates with any SMTP server.

Configuring Call Home

This section includes the following topics:

Guidelines for Configuring Call Home

To configure Call Home, perform the following steps:

Step 1 Assign contact information.

Step 2 Configure destination profiles.

Step 3 Associate one or more alert groups to each profile.

Step 4 (Optional) Add additional **show** commands to the alert groups.

Step 5 Configure transport options.

Step 6 Enable Call Home.

Step 7 (Optional) Test Call Home messages.

Configuring Contact Information

You must configure the e-mail, phone, and street address information for Call Home. You can optionally configure the contract ID, customer ID, site ID, and switch priority information.

To configure contact information, perform this task:

	Command	Purpose
Step 1	`switch# configuration terminal`	Enters configuration mode.
Step 2	`switch(config)# snmp-server contact sys-contact`	Configures the SNMP sysContact.
Step 3	`switch(config)# callhome`	Enters callhome configuration mode.
Step 4	`switch(config-callhome)# email-contact email-address`	Configures the e-mail address for the primary person responsible for the device. Up to 255 alphanumeric characters are accepted in e-mail address format. **Note** You can use any valid e-mail address. You cannot use spaces.

	Command	Purpose
Step 5	`switch(config-callhome)# phone-contact` *international-phone-number*	Configures the phone number in international phone number format for the primary person responsible for the device. Up to 17 alphanumeric characters are accepted in international format. **Note** You cannot use spaces. Be sure to use the **+** prefix before the number.
Step 6	`switch(config-callhome)# streetaddress` *address*	Configures the street address as an alphanumeric string with white paces for the primary person responsible for the device. Up to 255 alphanumeric characters are accepted, including spaces.
Step 7	`switch(config-callhome)# contract-id` *contract-number*	(Optional) Configures the contract number for this device from the service agreement. The contract number can be up to 255 alphanumeric characters in free format.
Step 8	`switch(config-callhome)# customer-id` *customer-number*	(Optional) Configures the customer number for this device from the service agreement. The customer number can be up to 255 alphanumeric characters in free format.
Step 9	`switch(config-callhome)# site-id` *site-number*	(Optional) Configures the site number for this device. The site number can be up to 255 alphanumeric characters in free format.
Step 10	`switch(config-callhome)# switch-priority` *number*	(Optional) Configures the switch priority for this device. The range is from 0 to 7, with 0 being the highest priority and 7 the lowest. The default is 7.
Step 11	`switch(config-callhome)# show callhome`	(Optional) Displays a summary of the Call Home configuration.
Step 12	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

The following example shows how to configure the contact information for Call Home:

```
switch# configuration terminal
switch(config)# snmp-server contact personname@example.com
switch(config)# callhome
switch(config-callhome)# email-contact admin@example.com
switch(config-callhome)# phone-contact +1-800-555-0100
switch(config-callhome)# street-address 123 Anystreet st. Anytown,AnyWhere
```

Creating a Destination Profile

To create a user-defined destination profile and configure the message format for that new destination profile, perform this task:

	Command	Purpose
Step 1	`switch# configuration terminal`	Enters configuration mode.
Step 2	`switch(config)# callhome`	Enters callhome configuration mode.
Step 3	`switch(config-callhome)# destination-profile` *name* `format {XML | full-txt | short-txt}`	Creates a new destination profile and sets the message format for the profile. The name can be any alphanumeric string up to 31 characters.
Step 4	`switch(config-callhome)# show callhome destination-profile [profile` *name*`]`	(Optional) Displays information about one or more destination profiles.
Step 5	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

The following example shows how to create a destination profile for Call Home:

```
switch# configuration terminal
switch(config)# callhome
switch(config-callhome)# destination-profile Noc101 format full-text
```

Modifying a Destination Profile

You can modify the following attributes for a predefined or user-defined destination profile:

- Destination address—The actual address, pertinent to the transport mechanism, to which the alert should be sent.
- Message formatting—The message format used for sending the alert (full text, short text, or XML).
- Message level—The Call Home message severity level for this destination profile.
- Message size—The allowed length of a Call Home message sent to the e-mail addresses in this destination profile.

See the "Associating an Alert Group with a Destination Profile" section on page 25-9 for information on configuring an alert group for a destination profile.

Note You cannot modify or delete the CiscoTAC-1 destination profile.

To modify the attributes for a destination profile, perform this task:

	Command	Purpose
Step 1	`switch# configuration terminal`	Enters configuration mode.
Step 2	`switch(config)# callhome`	Enters callhome configuration mode.

	Command	Purpose
Step 3	`switch(config-callhome)# destination-profile { name | full-txt-destination | short-txt-destination} email-addr address`	Configures an e-mail address for a user-defined or predefined destination profile. **Tip** You can configure up to 50 e-mail addresses in a destination profile.
Step 4	`destination-profile {name | full-txt-destination | short-txt-destination} message-level number`	Configures the Call Home message severity level for this destination profile. The switch sends only alerts that have a matching or higher Call Home severity level to destinations in this profile. The range is from 0 to 9, where 9 is the highest severity level.
Step 5	`switch(config-callhome)# destination-profile {name | full-txt-destination | short-txt-destination} message-size number`	Configures the maximum message size for this destination profile The range is from 0 to 4000000. The default is 4000000.
Step 6	`switch(config-callhome)# show callhome destination-profile [profile name]`	(Optional) Displays information about one or more destination profiles.
Step 7	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

The following example shows how to modify a destination profile for Call Home:

```
switch# configuration terminal
switch(config)# callhome
switch(config-callhome)# destination-profile full-text-destination email-addr
person@example.com
switch(config-callhome)# destination-profile full-text-destination message-level 5
switch(config-callhome)# destination-profile full-text-destination message-size 10000
```

Associating an Alert Group with a Destination Profile

To associate one or more alert groups with a destination profile, perform this task:

	Command	Purpose
Step 1	`switch# configuration terminal`	Enters configuration mode.
Step 2	`switch(config)# callhome`	Enters callhome configuration mode.
Step 3	`switch(config-callhome)# destination-profile name alert-group {All | Cisco-TAC | Configuration | Diagnostic | Environmental | Inventory | License | Linecard-Hardware | Syslog-group-port | System | Test}`	Associates an alert group with this destination profile. Use the **All** keyword to associate all alert groups with the destination profile.
Step 4	`switch(config-callhome)# show callhome destination-profile [profile name]`	(Optional) Displays information about one or more destination profiles.
Step 5	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

The following example shows how to associate all alert groups with the destination profile Noc101:

```
switch# configuration terminal
switch(config)# callhome
switch(config-callhome)# destination-profile Noc101 alert-group All
```

Adding show Commands to an Alert Group

Note You cannot add user-defined CLI **show** commands to the CiscoTAC-1 destination profile.

To assign a maximum of five user-defined CLI **show** commands to an alert group, perform this task:

	Command	Purpose
Step 1	`switch# configuration terminal`	Enters configuration mode.
Step 2	`switch(config)# callhome`	Enters callhome configuration mode.
Step 3	`switch(config-callhome)# alert-group {Configuration | Diagnostic | Environmental | Inventory | License | Linecard-Hardware | Syslog-group-port | System | Test} user-def-cmd show-cmd`	Adds the **show** command output to any Call Home messages sent for this alert group. You must enclose the **show** command in double quotes. Only valid **show** commands are accepted.
Step 4	`switch(config-callhome)# show callhome user-def-cmds`	(Optional) Displays information about all user-defined **show** commands added to alert groups.
Step 5	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

The following example shows how to add the **show ip routing** command o the Cisco-TAC alert group:

```
switch# configuration terminal
switch(config)# callhome
switch(config-callhome)# alert-group Configuration user-def-cmd "show ip routing"
```

Configuring E-Mail

You must configure the SMTP server address for the Call Home functionality to work. You can also configure the from and reply-to e-mail addresses.

To configure e-mail, perform this task:

	Command	Purpose
Step 1	`switch# configuration terminal`	Enters configuration mode.
Step 2	`switch(config)# callhome`	Enters callhome configuration mode.

	Command	Purpose
Step 3	`switch(config-callhome)# transport email smtp-server ip-address [port number] [use-vrf vrf-name]`	Configures the SMTP server as either the domain name server (DNS) name, IPv4 address, or IPv6 address). Optionally configures the port number. The port ranges is from 1 to 65535. The default port number is 25. Also optionally configures the VRF to use when communicating with this SMTP server.
Step 4	`switch(config-callhome)# transport email from email-address`	(Optional) Configures the e-mail from field for Call Home messages.
Step 5	`switch(config-callhome)# transport email reply-to email-address`	(Optional) Configures the e-mail reply-to field for Call Home messages.
Step 6	`switch(config-callhome)# show callhome transport-email`	(Optional) Displays information about the e-mail configuration for Call Home.
Step 7	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

The following example shows how to configure the e-mail options for Call Home messages:

```
switch# configuration terminal
switch(config)# callhome
switch(config-callhome)# transport email smtp-server 192.0.2.10 use-vrf Red
switch(config-callhome)# transport email from person@example.com
switch(config-callhome)# transport email reply-to person@example.com
```

Configuring Periodic Inventory Notification

You can configure the switch to periodically send a message with an inventory of all software services currently enabled and running on the device along with hardware inventory information. The switch generates two Call Home notifications, periodic configuration messages and periodic inventory messages.

To configure periodic inventory notification, perform this task:

	Command	Purpose
Step 1	`switch# configuration terminal`	Enters configuration mode.
Step 2	`switch(config)# callhome`	Enters callhome configuration mode.
Step 3	`switch(config-callhome)# periodic-inventory notification [interval days][timeofday time]`	Configures the periodic inventory messages. The interval range is from 1 to 30 days. The default is 7 days. The **timeofday** value is in HH:MM format.
Step 4	`switch(config-callhome)# show callhome`	(Optional) Displays information about Call Home.
Step 5	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

The following example shows how to configure the periodic inventory messages to generate every 20 days:

```
switch# configuration terminal
switch(config)# callhome
switch(config-callhome)# periodic-inventory notification interval 20
```

Disabling Duplicate Message Throttle

You can limit the number of duplicate messages received for the same event. By default, the switch limits the number of duplicate messages received for the same event. If the number of duplicate messages sent exceeds 30 messages within a 2-hour time frame, then the switch discards further messages for that alert type.

To disable duplicate message throttling in callhome configuration mode, perform this task:

Command	Purpose
`switch(config-callhome)# no duplicate-message throttle`	Disables duplicate message throttling for Call Home. Enabled by default.

Enabling or Disabling Call Home

Once you have configured the contact information, you can enable the Call Home function.

To enable Call Home in callhome configuration mode, perform this task:

Command	Purpose
`switch(config-callhome)# enable`	Enables Call Home. Disabled by default.

To disable Call Home in the callhome configuration mode, perform this task:

Command	Purpose
`switch(config-callhome)# no enable`	Disables Call Home. Disabled by default.

Testing Call Home Communications

You can generate a test message to test your Call Home communications.

To generate a test Call Home message, perform this task:

Command	Purpose
`switch(config-callhome)# callhome send {diagnostic | configuration}`	Sends the diagnostic or configuration Call Home message to all configured destinations.
`switch(config-callhome)# callhome test [inventory]`	Sends test or inventory Call Home message to all configured destinations.

Verifying Call Home Configuration

To display Call Home configuration information, perform one of these tasks:

Command	Purpose
`switch# show callhome`	Displays the status for Call Home.
`switch# show callhome destination-profile name`	Displays one or more Call Home destination profiles.
`switch# show callhome status`	Displays the Call Home status.
`switch# show callhome transport-email`	Displays the e-mail configuration for Call Home.
`switch# show callhome user-def-cmds`	Displays CLI commands added to any alert groups.
`switch# show running-config [callhome | callhome-all]` `show startup-config callhome`	Displays the running configuration for Call Home.
`switch# show startup-config callhome`	Displays the startup configuration for Call Home.
`switch# show tech-support callhome`	Displays the technical support output for Call Home.

Call Home Example Configuration

The following example uses CFS to create a destination profile called Noc101, associate the Cisco-TAC alert group to that profile, configure contact and e-mail information, and distribute those changes to all CFS-enabled devices:

```
switch(config)# configure terminal
switch(config)# snmp-server contact person@example.com
switch(config)# callhome
switch(config-callhome)# email-contact admin@example.com
switch(config-callhome)# phone-contact +1-800-555-0100
switch(config-callhome)# streetaddress 123 Anystreet st. Anytown, Anywhere
switch(config-callhome)# destination-profile Noc101 format full-txt
switch(config-callhome)# destination-profile full-txt alert-group Configuration
switch(config-callhome)# destination-profile full-txt email-addr person@example.com
switch(config-callhome)# destination-profile full-txt-destination message-level 5
switch(config-callhome)# alert-group Configuration user-def-cmd show running-config
switch(config-callhome)# transport email smtp-server A.B.C.D use-vrf Red
switch(config-callhome)# enable
```

Default Settings

Table 25-3 lists the default settings for Call Home parameters.

Table 25-3 Default Call Home Parameters

Parameters	Default
Destination message size for a message sent in full text format.	4000000
Destination message size for a message sent in XML format.	4000000
Destination message size for a message sent in short text format.	4000
SMTP server port number if no port is specified.	25
Alert group association with profile.	All for full-text-destination and short-text-destination profiles. The cisco-tac alert group for the CiscoTAC-1 destination profile.
Format type.	XML
Call Home message level.	0 (zero)

Additional References

For additional information related to implementing Call Home, see the following sections:

- Message Formats, page 25-14
- Sample Test Inventory Alert Notification in Full-Text Format, page 25-17
- Sample Test Inventory Alert Notification in XML Format, page 25-19

Message Formats

Call Home supports the following message formats:

- Fields in Short Text Message Format
- Common Fields for All Full Text and XML Messages
- Inserted Fields for an Inventory Event Message
- Inserted Fields for a User-Generated Test Message

Table 25-4 describes the short text formatting option for all message types.

Table 25-4 Fields in Short Text Message Format

Data Item	Description
Device identification	Configured device name
Date/time stamp	Time stamp of the triggering event
Error isolation message	Plain English description of triggering event
Alarm urgency level	Error level such as that applied to system message

Table 25-5 describes the common event message format for full text or XML.

Table 25-5 Common Fields for All Full Text and XML Messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Time stamp	Date and time stamp of event in ISO time notation: *YYYY-MM-DD HH:MM:SS GMT+HH:MM*.	/aml/header/time
Message name	Name of message. Specific event names are listed in the Table 25-4.	/aml/header/name
Message type	Name of message type, such as reactive or proactive.	/aml/header/type
Message group	Name of alert group, such as syslog.	/aml/header/group
Severity level	Severity level of message (see "Call Home Message Levels" section on page 25-4).	/aml/header/level
Device ID	Unique device identifier (UDI) for end device that generated the message. This field should be empty if the message is nonspecific to a device. The format is *type*@*Sid*@*serial*. • *type* is the product model number from backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number· • *serial* is the number identified by the Sid field. An example is WS-C6509@C@12345678	/aml/ header/deviceId
Customer ID	Optional user-configurable field used for contract information or other ID by any support service.	/aml/ header/customerID
Contract ID	Optional user-configurable field used for contract information or other ID by any support service.	/aml/ header /contractId
Site ID	Optional user-configurable field used for Cisco-supplied site ID or other data meaningful to alternate support service.	/aml/ header/siteId
Server ID	If the message is generated from the device, this is the unique device identifier (UDI) of the device. The format is *type*@*Sid*@*serial*. • *type* is the product model number from backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number· • *serial* is the number identified by the Sid field. An example is WS-C6509@C@12345678	/aml/header/serverId
Message description	Short text that describes the error.	/aml/body/msgDesc
Device name	Node that experienced the event (hostname of the device).	/aml/body/sysName

Table 25-5 Common Fields for All Full Text and XML Messages (continued)

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Contact name	Name of person to contact for issues associated with the node that experienced the event.	/aml/body/sysContact
Contact e-mail	E-mail address of person identified as the contact for this unit.	/aml/body/sysContactEmail
Contact phone number	Phone number of the person identified as the contact for this unit.	/aml/body/sysContactPhoneNumber
Street address	Optional field that contains the street address for RMA part shipments associated with this unit.	/aml/body/sysStreetAddress
Model name	Model name of the device (the specific model as part of a product family name).	/aml/body/chassis/name
Serial number	Chassis serial number of the unit.	/aml/body/chassis/serialNo
Chassis part number	Top assembly number of the chassis.	/aml/body/chassis/partNo
Fields specific to a particular alert group message are inserted here.		
The following fields may be repeated if multiple CLI commands are executed for this alert group.		
Command output name	Exact name of the issued CLI command.	/aml/attachments/attachment/name
Attachment type	Specific command output.	/aml/attachments/attachment/type
MIME type	Either plain text or encoding type.	/aml/attachments/attachment/mime
Command output text	Output of command automatically executed (see "Call Home Alert Groups" section on page 25-2).	/aml/attachments/attachment/atdata

Table 25-6 describes the inventory event message format for full text or XML.

Table 25-6 Inserted Fields for an Inventory Event Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Model	Model number of switch	/aml/body/chassis/Model
Hardware version	Hardware version of switch	/aml/body/chassis/Hardware
Serial number	Serial number of switch	/aml/body/chassis/Serial number

Table 25-7 describes the user-generated test message format for full text or XML.

Table 25-7 Inserted Fields for a User-Generated Test Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Process ID	Unique process ID.	/aml/body/process/id
Process state	State of process (for example, running or halted).	/aml/body/process/processState
Process exception	Exception or reason code.	/aml/body/process/exception

Sample Test Inventory Alert Notification in Full-Text Format

This sample output shows the full-text format for a syslog port alert-group notification:

```
Severity Level:2
Series:Chassis
Switch Priority:7
Device Id:DS-C9134-K9-SUP@C@jaf1241aarp
Customer Id:string
Contract Id:123
Site Id:Bangalore
Server Id:DS-C9134-K9-SUP@C@jaf1241aarp
Time of Event:2009-05-18 06:25:21 GMT+00:00 Message Name:full Message Type:inventory
System Name:switch Contact Name:XXXXXXX Contact Email:xxxxxxx@example.com Contact
Phone:+91-80-555-0199 Street Address:#71, Miller's Road Event Description:On-demand full
inventory process_name:platform manager start chassis information:
Affected Chassis:DS-C9134-K9-SUP
Affected Chassis Serial Number:jaf1241aarp Affected Chassis Hardware Version:1.0 Affected
Chassis Software Version:4.1(2)E1(1) Affected Chassis Part No:73-11757-01 end chassis
information:
fru
    Affected FRU:DS-C9134-K9-SUP
    Affected FRU Serial Number:jaf1241aarp
    Affected FRU Part No:
    Affected FRU Slot:1
fru
    Affected FRU:DS-C9134-K9-SUP
    Affected FRU Serial Number:jaf1241aarp
    Affected FRU Part No:
    Affected FRU Slot:n/a
start attachment
Ethernet Pluginend attachment start attachment

<output truncated....>

    name:show system uptime
    type:text
    data:
    System start time:          Mon May 18 05:37:46 2009
    System uptime:              0 days, 0 hours, 35 minutes, 43 seconds
    Kernel uptime:              0 days, 0 hours, 49 minutes, 19 seconds
end attachment
start attachment
    name:show interface transceiver
    type:text
    data:
    Ethernet1/1
        sfp is present
        name is CISCO-AVAGO
        part number is SFBR-7700SDZ
        revision is B4
        serial number is AGD121421VL
        nominal bitrate is 10300 MBits/sec
        Link length supported for 50/125um fiber is 82 m(s)
        Link length supported for 62.5/125um fiber is 26 m(s)
        cisco id is --
        cisco extended id number is 4

    Ethernet1/2
        sfp is not present

    Ethernet1/3
        sfp is not present
```

```
Ethernet1/4
    sfp is not present

Ethernet1/5
    sfp is present
    name is CISCO-AVAGO
    part number is SFBR-7700SDZ
    revision is B4
    serial number is AGD1213219F
    nominal bitrate is 10300 MBits/sec
    Link length supported for 50/125um fiber is 82 m(s)
    Link length supported for 62.5/125um fiber is 26 m(s)
    cisco id is --
    cisco extended id number is 4

Ethernet1/6
    sfp is not present

Ethernet1/7
    sfp is not present

Ethernet1/8
    sfp is not present

Ethernet1/9
    sfp is not present

Ethernet1/10
    sfp is not present

Ethernet1/11
    sfp is not present

Ethernet1/12
    sfp is not present

Ethernet1/13
    sfp is not present

Ethernet1/14
    sfp is not present

Ethernet1/15
    sfp is not present

Ethernet1/16
    sfp is not present

Ethernet1/17
    sfp is not present

Ethernet1/18
    sfp is not present

Ethernet1/19
    sfp is not present

Ethernet1/20
    sfp is not present
```

Sample Test Inventory Alert Notification in XML Format

This sample shows the XML format for a syslog port alert-group notification:

```
<?xml version="1.0" encoding="UTF-8" ?>
<soap-env:Envelope xmlns:soap-env="http://www.w3.org/2003/05/soap-envelope">
<soap-env:Header>
<aml-session:Session xmlns:aml-session="http://www.cisco.com/2004/01/aml-session"
soap-env:mustUnderstand="true"
soap-env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
<aml-session:To>http://tools.cisco.com/neddce/services/DDCEService</aml-session:To>
<aml-session:Path>
<aml-session:Via>http://www.cisco.com/appliance/uri</aml-session:Via>
</aml-session:Path>
<aml-session:From>http://www.cisco.com/appliance/uri</aml-session:From>
<aml-session:MessageId>1008:jaf1241aarp:4A10FF55</aml-session:MessageId>
</aml-session:Session>
</soap-env:Header>
<soap-env:Body>
<aml-block:Block xmlns:aml-block="http://www.cisco.com/2004/01/aml-block">
<aml-block:Header>
<aml-block:Type>http://www.cisco.com/2005/05/callhome/inventory</aml-block:Type>
<aml-block:CreationDate>2009-05-18 06:25:21 GMT+00:00</aml-block:CreationDate>
<aml-block:Builder>
<aml-block:Name> Chassis</aml-block:Name>
<aml-block:Version>4.1(2)E1(1)</aml-block:Version>
</aml-block:Builder>
<aml-block:BlockGroup>
<aml-block:GroupId>1009:jaf1241aarp:4A10FF55</aml-block:GroupId>
<aml-block:Number>0</aml-block:Number>
<aml-block:IsLast>true</aml-block:IsLast>
<aml-block:IsPrimary>true</aml-block:IsPrimary>
<aml-block:WaitForPrimary>false</aml-block:WaitForPrimary>
</aml-block:BlockGroup>
<aml-block:Severity>2</aml-block:Severity>
</aml-block:Header>
<aml-block:Content>
<ch-inv:CallHome xmlns:ch-inv="http://www.cisco.com/2005/05/callhome/inventory" "
version="1.0">
<ch-inv:EventTime>2009-05-18 06:25:21 GMT+00:00</ch-inv:EventTime>
<ch-inv:MessageDescription>On-demand full inventory</ch-inv:MessageDescription>
<ch-inv:Event>
<ch-inv:Type>inventory</ch-inv:Type>
<ch-inv:SubType>full</ch-inv:SubType>
<ch-inv:Brand>Cisco</ch-inv:Brand>
<ch-inv:Series> Chassis</ch-inv:Series> </ch-inv:Event> <ch-inv:CustomerData>
<ch-inv:UserData> <ch-inv:Email>xxxxx@cisco.com</ch-inv:Email>
</ch-inv:UserData>
<ch-inv:ContractData>
<ch-inv:CustomerId>xxxxxx</ch-inv:CustomerId>
<ch-inv:SiteId>Bangalore</ch-inv:SiteId>
<ch-inv:ContractId>123</ch-inv:ContractId>
<ch-inv:DeviceId>DS-C9134-K9-SUP@C@jaf1241aarp</ch-inv:DeviceId>
</ch-inv:ContractData>
<ch-inv:SystemInfo>
<ch-inv:Name>switch</ch-inv:Name>
<ch-inv:Contact>xxxxxxx</ch-inv:Contact>
<ch-inv:ContactEmail>xxxxxxx@example.com</ch-inv:ContactEmail>
<ch-inv:ContactPhoneNumber>+91-80-555-0199</ch-inv:ContactPhoneNumber>
<ch-inv:StreetAddress>#71, Miller&apos;s Road</ch-inv:StreetAddress> </ch-inv:SystemInfo>
</ch-inv:CustomerData> <ch-inv:Device> <rme:Chassis
xmlns:rme="http://www.cisco.com/rme/4.0">
<rme:Model>DS-C9134-K9-SUP</rme:Model>
```

```
<rme:HardwareVersion>1.0</rme:HardwareVersion>
<rme:SerialNumber>jaf1241aarp</rme:SerialNumber>
<rme:Card>
<rme:Model>DS-C9134-K9-SUP</rme:Model>
<rme:SerialNumber>jaf1241aarp</rme:SerialNumber>
<rme:LocationWithinContainer>1</rme:LocationWithinContainer>
<rme:PartNumber></rme:PartNumber>
<rme:SoftwareIdentity>
<rme:VersionString></rme:VersionString>
</rme:SoftwareIdentity>
</rme:Card>
<rme:Card>
<rme:Model>DS-C9134-K9-SUP</rme:Model>
<rme:SerialNumber>jaf1241aarp</rme:SerialNumber>
<rme:LocationWithinContainer>n/a</rme:LocationWithinContainer>
<rme:PartNumber></rme:PartNumber>
<rme:SoftwareIdentity>
<rme:VersionString></rme:VersionString>
</rme:SoftwareIdentity>
</rme:Card>
</rme:Chassis>
</ch-inv:Device>
</ch-inv:CallHome>
</aml-block:Content>
<aml-block:Attachments>
<aml-block:Attachment type="inline">


<output truncated....>

<![CDATA[Ethernet1/1
    sfp is present
    name is CISCO-AVAGO
    part number is SFBR-7700SDZ
    revision is B4
    serial number is AGD121421VL
    nominal bitrate is 10300 MBits/sec
    Link length supported for 50/125um fiber is 82 m(s)
    Link length supported for 62.5/125um fiber is 26 m(s)
    cisco id is --
    cisco extended id number is 4

Ethernet1/2
    sfp is not present

Ethernet1/3
    sfp is not present

Ethernet1/4
    sfp is not present

Ethernet1/5
    sfp is present
    name is CISCO-AVAGO
    part number is SFBR-7700SDZ
    revision is B4
    serial number is AGD1213219F
    nominal bitrate is 10300 MBits/sec
    Link length supported for 50/125um fiber is 82 m(s)
    Link length supported for 62.5/125um fiber is 26 m(s)
    cisco id is --
    cisco extended id number is 4

Ethernet1/6
```

```
    sfp is not present

Ethernet1/7
    sfp is not present

Ethernet1/8
    sfp is not present

Ethernet1/9
    sfp is not present

Ethernet1/10
    sfp is not present

Ethernet1/11
    sfp is not present

Ethernet1/12
    sfp is not present

Ethernet1/13
    sfp is not present

Ethernet1/14
    sfp is not present

Ethernet1/15
    sfp is not present

Ethernet1/16
    sfp is not present

Ethernet1/17
    sfp is not present

Ethernet1/18
    sfp is not present

Ethernet1/19
    sfp is not present

Ethernet1/20
    sfp is not present

]]>
</aml-block:Data>
</aml-block:Attachment>
</aml-block:Attachments>
</aml-block:Block>
</soap-env:Body>
</soap-env:Envelope>
```

CHAPTER 26

Configuring System Message Logging

This chapter describes how to configure system message logging on the switch.

This chapter includes the following sections:

Information About System Message Logging

You can use system message logging to control the destination and to filter the severity level of messages that system processes generate. You can configure logging to terminal sessions, a log file, and syslog servers on remote systems.

By default, the switch outputs messages to terminal sessions. For information about configuring logging to terminal sessions, see the "Configuring System Message Logging to Terminal Sessions" section on page 26-2.

By default, the switch logs system messages to a log file. For information about configuring logging to a file, see the "Configuring System Message Logging to a File" section on page 26-3.

Table 26-1 describes the severity levels used in system messages. When you configure the severity level, the system outputs messages at that level and lower.

Table 26-1 System Message Severity Levels

Level	Description
0 – emergency	System unusable
1 – alert	Immediate action needed
2 – critical	Critical condition
3 – error	Error condition
4 – warning	Warning condition
5 – notification	Normal but significant condition

Table 26-1 System Message Severity Levels (continued)

Level	Description
6 – informational	Informational message only
7 – debugging	Appears during debugging only

The switch logs the most recent 100 messages of severity 0, 1, or 2 to the NVRAM log. You cannot configure logging to the NVRAM.

You can configure which system messages should be logged based on the facility that generated the message and its severity level. For information about configuring the severity level by module and facility, see the "Configuring Module and Facility Messages Logged" section on page 26-4.

syslog Servers

Syslog servers run on remote systems that are configured to log system messages based on the syslog protocol. You can configure up to three syslog servers. For information about configuring syslog servers, see the "Configuring syslog Servers" section on page 26-5.

Note When the switch first initializes, messages are sent to syslog servers only after the network is initialized.

Configuring System Message Logging

This section includes the following topics:

Configuring System Message Logging to Terminal Sessions

You can configure the switch to log messages by their severity level to console, Telnet, and SSH sessions.

By default, logging is enabled for terminal sessions. To configure the switch to log messages, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# logging console [severity-level]`	Enables the switch to log messages to the console session based on a specified severity level or higher. Severity levels, which can range from 0 to 7, are listed in Table 26-1. If the severity level is not specified, the default of 2 is used.
	`switch(config)# no logging console [severity-level]`	Disables the ability of the switch to log messages to the console.
Step 3	`switch(config)# show logging console`	(Optional) Displays the console logging configuration.
Step 4	`switch(config)# logging monitor [severity-level]`	Enables the switch to log messages to the monitor based on a specified severity level or higher. The configuration applies to Telnet and SSH sessions. Severity levels, which can range from 0 to 7, are listed in Table 26-1. If the severity level is not specified, the default of 5 is used.
	`switch(config)# no logging monitor [severity-level]`	Disables logging messages to telnet and SSH sessions.
Step 5	`switch(config)# show logging monitor`	(Optional) Displays the monitor logging configuration. The regular severity level is ignored.
Step 6	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure a switch to log messages:

```
switch# configure terminal
switch(config)# logging console 3
switch(config)# no logging console
switch(config)# show logging console
switch(config)# logging monitor 3
switch(config)# no logging monitor
switch(config)# show logging monitor
switch(config)# copy running-config startup-config
```

Configuring System Message Logging to a File

You can configure the switch to log system messages to a file. By default, system messages are logged to the file log:messages.

For information about displaying and clearing log files, see the "Displaying and Clearing Log Files" section on page 26-7.

To configure the switch to log system messages to a file, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# logging logfile` *logfile-name severity-level* [**size** *bytes*]	Configures the name of the log file used to store system messages and the minimum severity level to log. You can optionally specify a maximum file size. The default severity level is 5 and the file size is 10485760. Severity levels are listed in Table 26-1. The file size is from 4096 to 10485760 bytes.
	`switch(config)# no logging logfile` [*logfile-name severity-level* [**size** *bytes*]]	Disables logging to the log file.
Step 3	`switch(config)# show logging info`	(Optional) Displays the logging configuration.
Step 4	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure a switch to log system messages to a file:

```
switch# configure terminal
switch(config)# logging logfile my_log size 6
switch(config)# no logging logfile
switch(config)# show logging info
switch(config)# copy running-config startup-config
```

Configuring Module and Facility Messages Logged

To configure the severity level and time-stamp units of messages logged by modules and facilities, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# logging module` [*severity-level*]	Enables module log messages that have the specified severity level or higher. Severity levels, which range from 0 to 7, are listed in Table 26-1. If the severity level is not specified, the default of 5 is used.
	`switch(config)# no logging module` [*severity-level*]	Disables module log messages.
Step 3	`switch(config)# show logging module`	(Optional) Displays the module logging configuration.

	Command	Purpose
Step 4	`switch(config)# logging level facility severity-level`	Enables logging messages from the specified facility that have the specified severity level or higher. Severity levels, which range from 0 to 7, are listed in Table 26-1. To apply the same severity level to all facilities, use the **all** facility. For defaults, see the **show logging level** command.
	`switch(config)# no logging level [facility severity-level]`	Resets the logging severity level for the specified facility to its default level. If you do not specify a facility and severity level, the switch resets all facilities to their default levels.
Step 5	`switch(config)# show logging level [facility]`	(Optional) Displays the logging level configuration and the system default level by facility. If you do not specify a facility, the switch displays levels for all facilities.
Step 6	`switch(config)# logging timestamp {microseconds | milliseconds | seconds}`	Sets the logging time-stamp units. By default, the units are seconds.
	`switch(config)# no logging timestamp {microseconds | milliseconds | seconds}`	Resets the logging time-stamp units to the default of seconds.
Step 7	`switch(config)# show logging timestamp`	(Optional) Displays the logging time-stamp units configured.
Step 8	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure the severity level and time-stamp units of messages:

```
switch# configure terminal
switch(config)# logging module 3
switch(config)# show logging module
switch(config)# logging level aaa 2
switch(config)# logging timestamp milliseconds
switch(config)# show logging timestamp
switch(config)# copy running-config startup-config
```

Configuring syslog Servers

You can configure up to three syslog servers that reference remote systems where you want to log system messages.

You can configure a syslog server on a UNIX or Linux system by adding the following line to the /etc/syslog.conf file:

```
facility.level <five tab characters> action
```

Table 26-2 describes the syslog fields that you can configure.

Table 26-2 syslog Fields in syslog.conf

Field	Description
Facility	Creator of the message, which can be auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, syslog, user, local0 through local7, or an asterisk (*) for all. These facility designators allow you to control the destination of messages based on their origin. The default outgoing facility value is local7. **Note** Check your configuration before using a local facility.
Level	Minimum severity level at which messages are logged, which can be debug, info, notice, warning, err, crit, alert, emerg, or an asterisk (*) for all. You can use none to disable a facility. The default severity level is notification(5).
Action	Destination for messages, which can be a filename, a hostname preceded by the at sign (@), or a comma-separated list of users or an asterisk (*) for all logged-in users.

To configure a syslog server on a UNIX or Linux system, perform the following steps:

Step 1 Log debug messages with the local7 facility in the file /var/log/myfile.log by adding the following line to the /etc/syslog.conf file:

```
debug.local7              /var/log/myfile.log
```

Step 2 Create the log file by entering these commands at the shell prompt:

```
$ touch /var/log/myfile.log
$ chmod 666 /var/log/myfile.log
```

Step 3 Make sure the system message logging daemon reads the new changes by checking myfile.log after entering this command:

```
$ kill -HUP ~cat /etc/syslog.pid~
```

To configure syslog servers, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# logging server` *host* [*severity-level* [*facility*]]	Configures a syslog server at the specified hostname or IPv4 or IPv6 address. You can limit logging of messages with a minimum severity level and for a specific facility. Severity levels, which range from 0 to 7, are listed in Table 26-1. The default outgoing facility is local7.
	`switch(config)# no logging server` *host*	Removes the logging server for the specified host.
Step 3	Repeat Step 2 for up to three syslog servers.	
Step 4	`switch(config)# show logging server`	(Optional) Displays the syslog server configuration.
Step 5	`switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

The following example shows how to configure a syslog server:

```
switch# configure terminal
switch(config)# logging server 172.28.254.254 5 local3
switch(config)# show logging server
switch(config)# copy running-config startup-config
```

Displaying and Clearing Log Files

To display or clear messages in the log file and the NVRAM, perform this task:

	Command	Purpose
Step 1	`switch# show logging last number-lines`	Displays the last number of lines in the logging file. You can specify from 1 to 9999 for the last number of lines.
Step 2	`switch# show logging logfile [start-time yyyy mmm dd hh:mm:ss] [end-time yyyy mmm dd hh:mm:ss]`	Displays the messages in the log file that have a time stamp within the span entered. If you do not enter an end time, the current time is used. You enter three characters for the month time field, and digits for the year and day time fields.
Step 3	`switch# show logging nvram [last number-lines]`	Displays the messages in the NVRAM. To limit the number of lines displayed, you can enter the last number of lines to display. You can specify from 1 to 100 for the last number of lines.
Step 4	`switch# clear logging logfile`	Clears the contents of the log file.
Step 5	`switch# clear logging nvram`	Clears the logged messages in NVRAM.

The following example shows how to display or clear messages in a log file:

```
switch# show logging last 40
switch# show logging logfile start-time 2007 nov 1 15:10:0
switch# show logging nvram last 10
switch# clear logging logfile
switch# clear logging nvram
```

Verifying System Message Logging Configuration

To display system message logging configuration information, perform one of these tasks:

Command	Purpose
`switch# show logging console`	Displays the console logging configuration.
`switch# show logging info`	Displays the logging configuration.
`switch# show logging internal info`	Displays the syslog distribution information.
`switch# show logging last number-lines`	Displays the last number of lines of the log file.
`switch# show logging level [facility]`	Displays the facility logging severity level configuration.

Command	Purpose
switch# **show logging logfile** [**start-time** *yyyy mmm dd hh:mm:ss*] [**end-time** *yyyy mmm dd hh:mm:ss*]	Displays the messages in the log file.
switch# **show logging module**	Displays the module logging configuration.
switch# **show logging monitor**	Displays the monitor logging configuration.
switch# **show logging nvram** [**last** *number-lines*]	Displays the messages in the NVRAM log.
switch# **show logging pending**	Displays the syslog server pending distribution configuration.
switch# **show logging pending-diff**	Displays the syslog server pending distribution configuration differences.
switch# **show logging server**	Displays the syslog server configuration.
switch# **show logging session**	Displays the logging session status.
switch# **show logging status**	Displays the logging status.
switch# **show logging timestamp**	Displays the logging time-stamp units configuration.

System Message Logging Example Configuration

The following example shows how to configure system message logging:

```
switch(config)# logging console 1
switch(config)# logging monitor 3
switch(config)# logging logfile my_log 6
switch(config)# logging module 3
switch(config)# logging level aaa 2
switch(config)# logging timestamp milliseconds
switch(config)# logging server 172.29.231.8
switch(config)# copy running-config startup-config
```

Default Settings

Table 26-3 lists the default settings for system message logging parameters.

Table 26-3 Default System Message Logging Parameters

Parameters	Default
Console logging	Enabled at severity level 2
Monitor logging	Enabled at severity level 2
Log file logging	Enabled to log:messages at severity level 5
Module logging	Enabled at severity level 5
Facility logging	Enabled
Time-stamp units	Seconds
Syslog server logging	Disabled
Syslog server configuration distribution	Disabled

CHAPTER 27

Configuring SNMP

This chapter describes how to configure the SNMP feature in the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

This chapter includes the following sections:

Information About SNMP

The Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

This section includes the following topics:

SNMP Functional Overview

The SNMP framework consists of three parts:

- An SNMP manager—The system used to control and monitor the activities of network devices using SNMP.
- An SNMP agent—The software component within the managed device that maintains the data for the device and reports these data, as needed, to managing systems. The switch supports the agent and MIB. To enable the SNMP agent, you must define the relationship between the manager and the agent.

- A Management Information Base (MIB)—The collection of managed objects on the SNMP agent

SNMP is defined in RFCs 3411 to 3418.

Note Cisco NX-OS does not support SNMP sets for Ethernet MIBs.

The switch supports SNMPv1, SNMPv2c and SNMPv3. Both SNMPv1 and SNMPv2c use a community-based form of security.

SNMP Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

Cisco NX-OS generates SNMP notifications as either traps or informs. Traps are less reliable than informs because the SNMP manager does not send any acknowledgment when it receives a trap. The switch cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the switch never receives a response, it can send the inform request again.

You can configure Cisco NX-OS to send notifications to multiple host receivers. See the "Configuring SNMP Notification Receivers" section on page 27-6 for more information about host receivers.

SNMPv3

SNMPv3 provides secure access to devices by a combination of authenticating and encrypting frames over the network. The security features provided in SNMPv3 are the following:

- Message integrity—Ensures that a packet has not been tampered with in-transit.
- Authentication—Determines the message is from a valid source.
- Encryption—Scrambles the packet contents to prevent it from being seen by unauthorized sources.

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy that is set up for a user and the role in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet.

This section contains the following topics:

Security Models and Levels for SNMPv1, v2, v3

The security level determines if an SNMP message needs to be protected from disclosure and if the message needs to be authenticated. The various security levels that exist within a security model are as follows:

- noAuthNoPriv—Security level that does not provide authentication or encryption.
- authNoPriv—Security level that provides authentication but does not provide encryption.
- authPriv—Security level that provides both authentication and encryption.

Three security models are available: SNMPv1, SNMPv2c, and SNMPv3. The security model combined with the security level determine the security mechanism applied when the SNMP message is processed.

User-Based Security Model

Table 27-1 identifies what the combinations of security models and levels mean.

Table 27-1 SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	HMAC-MD5 or HMAC-SHA	No	Provides authentication based on the Hash-Based Message Authentication Code (HMAC) Message Digest 5 (MD5) algorithm or the HMAC Secure Hash Algorithm (SHA).
v3	authPriv	HMAC-MD5 or HMAC-SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides Data Encryption Standard (DES) 56-bit encryption in addition to authentication based on the Cipher Block Chaining (CBC) DES (DES-56) standard.

SNMPv3 User-Based Security Model (USM) refers to SNMP message-level security and offers the following services:

- Message integrity—Ensures that messages have not been altered or destroyed in an unauthorized manner and that data sequences have not been altered to an extent greater than can occur non-maliciously.
- Message origin authentication—Ensures that the claimed identity of the user on whose behalf received data was originated is confirmed.
- Message confidentiality—Ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes.

SNMPv3 authorizes management operations only by configured users and encrypts SNMP messages.

Cisco NX-OS uses two authentication protocols for SNMPv3:

- HMAC-MD5-96 authentication protocol
- HMAC-SHA-96 authentication protocol

Cisco NX-OS uses Advanced Encryption Standard (AES) as one of the privacy protocols for SNMPv3 message encryption and conforms with RFC 3826.

The **priv** option offers a choice of DES or 128-bit AES encryption for SNMP security encryption. The **priv** option along with the **aes-128** token indicates that this privacy password is for generating a 128-bit AES key.The AES priv password can have a minimum of eight characters. If the passphrases are specified in clear text, you can specify a maximum of 64 characters. If you use the localized key, you can specify a maximum of 130 characters.

Note For an SNMPv3 operation using the external AAA server, you must use AES for the privacy protocol in user configuration on the external AAA server.

CLI and SNMP User Synchronization

SNMPv3 user management can be centralized at the Authentication, Authorization, and Accounting (AAA) server level. This centralized user management allows the SNMP agent in Cisco NX-OS to leverage the user authentication service of the AAA server. Once user authentication is verified, the SNMP PDUs are processed further. Additionally, the AAA server is also used to store user group names. SNMP uses the group names to apply the access/role policy that is locally available in the switch.

Any configuration changes made to the user group, role, or password results in database synchronization for both SNMP and AAA.

Cisco NX-OS synchronizes user configuration in the following ways:

- The **auth** passphrase specified in the **snmp-server user** command becomes the password for the CLI user.
- The password specified in the **username** command becomes as the **auth** and **priv** passphrases for the SNMP user.
- Deleting a user using either SNMP or the CLI results in the user being deleted for both SNMP and the CLI.
- User-role mapping changes are synchronized in SNMP and the CLI.

Note When you configure passphrase/password in localized key/encrypted format, Cisco NX-OS does not synchronize the password.

Group-Based SNMP Access

Note Because *group* is a standard SNMP term used industry-wide, roles are referred to as groups in this SNMP section.

SNMP access rights are organized by groups. Each group in SNMP is similar to a role through the CLI. Each group is defined with three accesses: read access, write access, and notification access. Each access can be enabled or disabled within each group.

You can begin communicating with the agent once your username is created, your roles are set up by your administrator, and you are added to the roles.

Configuration Guidelines and Limitations

Cisco NX-OS supports read-only access to Ethernet MIBs.

Configuring SNMP

This section includes the following topics:

Configuring SNMP Users

To configure a user for SNMP, perform this task:

	Command	Purpose
Step 1	`switch# configuration terminal`	Enters configuration mode.
Step 2	`switch(config)# snmp-server user` *name* `[auth {md5 | sha} passphrase [auto] [priv [aes-128] passphrase] [engineID id][localizedkey]]`	Configures an SNMP user with authentication and privacy parameters.
Step 3	`switch(config)# show snmp user`	(Optional) Displays information about one or more SNMP users.
Step 4	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

Enforcing SNMP Message Encryption

You can configure SNMP to require authentication or encryption for incoming requests. By default the SNMP agent accepts SNMPv3 messages without authentication and encryption. When you enforce privacy, Cisco NX-OS responds with an authorization Error for any SNMPv3 PDU request using securityLevel parameter of either noAuthNoPriv or authNoPriv.

To enforce SNMP message encryption for a user in the global configuration mode, perform this task:

Command	Purpose
`switch(config)# snmp-server user name enforcePriv`	Enforces SNMP message encryption for this user.

To enforce SNMP message encryption for all users in the global configuration mode, perform this task:

Command	Purpose
`switch(config)# snmp-server globalEnforcePriv`	Enforces SNMP message encryption for all users.

Assigning SNMPv3 Users to Multiple Roles

After you configure an SNMP user, you can assign multiple roles for the user.

Note Only users belonging to a network-admin role can assign roles to other users.

To assign a role to an SNMP user in a global configuration mode, perform this task:

Command	Purpose
`switch(config)# snmp-server user name group`	Associates this SNMP user with the configured user role.

Creating SNMP Communities

You can create SNMP communities for SNMPv1 or SNMPv2c.

To create an SNMP community string in a global configuration mode, perform this task:

Command	Purpose
`switch(config)# snmp-server community name group {ro | rw}`	Creates an SNMP community string.

Configuring SNMP Notification Receivers

You can configure Cisco NX-OS to generate SNMP notifications to multiple host receivers.

To configure a host receiver for SNMPv1 traps in a global configuration mode, perform this task:

Command	Purpose
`switch(config)# snmp-server host` *ip-address* **traps** {**version 1**] *community* [**udp_port** *number*]	Configures a host receiver for SNMPv1 traps. The community can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.

To configure a host receiver for SNMPv2c traps or informs in a global configuration mode, perform this task:

Command	Purpose
`switch(config)# snmp-server host` *ip-address* {**traps** | **informs**} **version 2c** *community* [**udp_port** *number*]	Configures a host receiver for SNMPv2c traps or informs. The community can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.

To configure a host receiver for SNMPv3 traps or informs in a global configuration mode, perform this task:

Command	Purpose
`switch(config)# snmp-server host` *ip-address* {**traps** | **informs**} **version 3** {**auth** | **noauth** | **priv** } *username* [**udp_port** *number*]	Configures a host receiver for SNMPv2c traps or informs. The username can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.

The following example shows how to configure a host receiver for an SNMPv3 inform:

```
switch(config)# snmp-server host 192.0.2.1 informs version 3 auth NMS
```

Note The SNMP manager must know the user credentials (authKey/PrivKey) based on the SNMP engineID of the switch to authenticate and decrypt the SNMPv3 messages.

Configuring the Notification Target User

You must configure a notification target user on the device to send SNMPv3 inform notifications to a notification host receiver.

The switch uses the credentials of the notification target user to encrypt the SNMPv3 inform notification messages to the configured notification host receiver.

Note For authenticating and decrypting the received INFORM PDU, The notification host receiver should have the same user credentials as configured in the switch to authenticate and decrypt the informs.

Use the following command in global configuration mode to configure the notification target user:

Command	Purpose
`switch(config)# snmp-server user` *name* [**auth** {md5 | sha} *passphrase* [**auto**] [**priv** [**aes-128**] *passphrase*] [**engineID** *id*]	Configures the notification target user with the specified engine ID for notification host receiver. The engineID format is a 12-digit colon-separated hexadecimal number.

The following example shows how to configure a notification target user:

```
switch(config)# snmp-server user NMS auth sha abcd1234 priv abcdefgh enginID
00:00:00:63:00:01:00:a1:ac:15:10:03
```

Enabling SNMP Notifications

You can enable or disable notifications. If you do not specify a notification name, Cisco NX-OS enables all notifications.

Note The **snmp-server enable traps** CLI command enables both traps and informs, depending on the configured notification host receivers.

Table 27-2 lists the CLI commands that enable the notifications for Cisco NX-OS MIBs.

Table 27-2 Enabling SNMP Notifications

MIB	Related Commands
All notifications	**snmp-server enable traps**
CISCO-AAA-SERVER-MIB	**snmp-server enable traps aaa**
ENTITY-MIB, CISCO-ENTITY-FRU-CONTROL-MIB, CISCO-ENTITY-SENSOR-MIB	**snmp-server enable traps entity** **snmp-server enable traps entity fru**
CISCO-LICENSE-MGR-MIB	**snmp-server enable traps license**
IF-MIB	**snmp-server enable traps link**
SNMPv2-MIB	**snmp-server enable traps snmp** **snmp-server enable traps snmp authentication**

Note The license notifications are enabled by default. All other notifications are disabled by default.

To enable the specified notification in the global configuration mode, perform one of these tasks:

Command	Purpose
`switch(config)# snmp-server enable traps`	Enables all SNMP notifications.
`switch(config)# snmp-server enable traps aaa [server-state-change]`	Enables the AAA SNMP notifications.
`switch(config)# snmp-server enable traps entity [fru]`	Enables the ENTITY-MIB SNMP notifications.
`switch(config)# snmp-server enable traps license`	Enables the license SNMP notification.
`switch(config)# snmp-server enable traps port-security`	Enables the port security SNMP notifications.
`switch(config)# snmp-server enable traps snmp [authentication]`	Enables the SNMP agent notifications.

Configuring linkUp/linkDown Notifications

You can configure which linkUp/linkDown notifications to enable on a device. You can enable the following types of linkUp/linkDown notifications:

- Cisco—Cisco NX-OS sends only the Cisco-defined notifications (cieLinkUp, cieLinkDow in CISCO-IF-EXTENSION-MIB.my), if ifLinkUpDownTrapEnable (defined in IF-MIB) is enabled for that interface.
- IETF—Cisco NX-OS sends only the IETF-defined notifications (linkUp, linkDown in IF-MIB) with only the defined varbinds, if ifLinkUpDownTrapEnable (defined in IF-MIB) is enabled for that interface.
- IEFT extended—Cisco NX-OS sends only the IETF-defined notifications (linkUp, linkDown defined in IF-MIB), if ifLinkUpDownTrapEnable (defined in IF-MIB) is enabled for that interface. Cisco NX-OS adds additional varbinds specific to Cisco Systems in addition to the varbinds defined in the IF-MIB. This is the default setting.
- IEFT Cisco—Cisco NX-OS sends the notifications (linkUp, linkDown) defined in IF-MIB and notifications (cieLinkUp, cieLinkDown) defined in CISCO-IF-EXTENSION-MIB.my, if ifLinkUpDownTrapEnable (defined in IF-MIB) is enabled for that interface. Cisco NX-OS sends only the varbinds defined in the linkUp and linkDown notifications.
- IEFT extended Cisco—Cisco NX-OS sends the notifications (linkUp, linkDown) defined in IF-MIB and notifications (cieLinkUp, cieLinkDown) defined in CISCO-IF-EXTENSION-MIB.my, if ifLinkUpDownTrapEnable (defined in IF-MIB) is enabled for that interface. Cisco NX-OS adds additional varbinds specific to Cisco Systems in addition to the varbinds defined in the IF-MIB for the linkUp and linkDown notifications.

To configure the type of linkUp/linkDown notifications in a global configuration mode, perform this task:

Command	Purpose
`switch(config)# snmp-server enable traps link [cisco] [ietf | ietf-extended]`	Enables the link SNMP notifications.

Disabling Up/ Down Notifications on an Interface

You can disable linkUp and linkDown notifications on an individual interface. You can use this limit notifications on flapping interface (an interface that transitions between up and down repeatedly).

To disable linkUp/linkDown notifications for the interface in interface configuration mode, perform this task:

Command	Purpose
`switch(config-if)# no snmp trap link-status`	Disables SNMP link-state traps for the interface. Enabled by default.

Enabling One-Time Authentication for SNMP over TCP

You can enable a one-time authentication for SNMP over a TCP session.

To enable one-time authentication for SNMP over TCP in global configuration mode, perform this task:

Command	Purpose
`switch(config)# snmp-server tcp-session [auth]`	Enables a one-time authentication for SNMP over a TCP session. Default is disabled.

Assigning SNMP Switch Contact and Location Information

You can assign the switch contact information, which is limited to 32 characters (without spaces), and the switch location. To assign the information, perform this task:

	Command	Purpose
Step 1	`switch# configuration terminal`	Enters configuration mode.
Step 2	`switch(config)# snmp-server contact` *name*	Configures sysContact, the SNMP contact name.
Step 3	`switch(config)# snmp-server location` *name*	Configures sysLocation, the SNMP location.
Step 4	`switch(config-callhome)# show snmp`	(Optional) Displays information about one or more destination profiles.
Step 5	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

Verifying SNMP Configuration

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

Command	Purpose
`switch# show snmp`	Displays the SNMP status.
`switch# show snmp community`	Displays the SNMP community strings.
`switch# show snmp engineID`	Displays the SNMP engineID.
`switch# show snmp group`	Displays SNMP roles.
`switch# show snmp sessions`	Displays SNMP sessions.
`switch# show snmp trap`	Displays the SNMP notifications enabled or disabled.
`switch# show snmp user`	Displays SNMPv3 users.

SNMP Example Configuration

The following example configures the switch to send the Cisco linkUp/linkDown notifications to one notification host receiver and defines two SNMP users, Admin and NMS:

```
switch # configuration terminal
switch(config)# snmp-server contact Admin@example.com
switch(config)# snmp-server user Admin auth sha A1D2e6te priv W1laT3R9
switch(config)# snmp-server user NMS auth sha A1D2e6te priv W1laT3R9 enginID
00:00:00:63:00:01:00:a1:ac:15:10:03
switch(config)# snmp-server host 192.0.2.1 informs version 3 auth NMS
switch(config)# snmp-server host 192.0.2.1
switch(config)# snmp-server enable traps link
```

Default Settings

Table 27-3 lists the default settings for SNMP parameters.

Table 27-3 Default SNMP Parameters

Parameters	Default
license notifications	enabled
linkUp/Down notification type	ietf-extended

CHAPTER 28

Configuring RMON

This chapter describes how to configure the RMON feature.

This chapter includes the following sections:

Information About RMON

RMON is an Internet Engineering Task Force (IETF) standard monitoring specification that allows various network agents and console systems to exchange network monitoring data. The Cisco NX-OS supports RMON alarms, events and logs to monitor the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

An RMON alarm monitors a specific management information base (MIB) object for a specified interval, triggers an alarm at a specified threshold value (threshold), and resets the alarm at another threshold value. You can use alarms with RMON events to generate a log entry or an SNMP notification when the RMON alarm triggers.

RMON is disabled by default and no events or alarms are configured in the switch. You can configure your RMON alarms and events by using the CLI or an SNMP-compatible network management station

This section includes the following topics:

RMON Alarms

You can set an alarm on any MIB object that resolves into an SNMP INTEGER type. The specified object must be an existing SNMP MIB object in standard dot notation (for example, 1.3.6.1.2.1.2.2.1.17 represents ifOutOctets.17).

When you create an alarm, you specify the following parameters:

- MIB object to monitor
- Sampling interval—The interval that the switch uses to collect a sample value of the MIB object.
- The sample type—Absolute samples take the current snapshot of the MIB object value. Delta samples take two consecutive samples and calculate the difference between them.
- Rising threshold—The value at which the switch triggers a rising alarm or resets a falling alarm.
- Falling threshold—The value at which the switch triggers a falling alarm or resets a rising alarm.
- Events—The action that the switch takes when an alarm (rising or falling) triggers.

Note Use the hcalarms option to set an alarm on a 64-bit integer MIB object.

For example, you can set a delta type rising alarm on an error counter MIB object. If the error counter delta exceeds this value, you can trigger an event that sends an SNMP notification and logs the rising alarm event. This rising alarm will not occur again until the delta sample for the error counter drops below the falling threshold.

Note The falling threshold must be less than the rising threshold.

RMON Events

You can associate a particular event to each RMON alarm. RMON supports the following event types:

- SNMP notification—Sends an SNMP risingAlarm or fallingAlarm notification when the associated alarm triggers.
- Log—Adds an entry in the RMON log table when the associated alarm triggers.
- Both—Sends an SNMP notification and adds an entry in the RMON log table when the associated alarm triggers.

You can specify a different even for a falling alarm and a rising alarm.

Configuration Guidelines and Limitations

RMON has the following configuration guidelines and limitations:

- You must configure an SNMP user an notification receiver to use the SNMP notification event type.
- You can only configure an RMON alarm on a MIB object that resolves to an integer.

Configuring RMON

This section includes the following topics:

Configuring RMON Alarms

You can configure RMON alarms on any integer-based SNMP MIB object.

You can optionally specify the following parameters:

- The event-number to trigger if the rising or falling threshold exceeds the specified limit.
- The owner of the alarm.

Ensure you have configured an SNMP user and enabled SNMP notifications (see the "Configuring SNMP" section on page 27-5).

To configure RMON alarms, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# rmon alarm` *index mib-object sample-interval* {**absolute** | **delta**} **rising-threshold** *value* [*event-index*] f**alling-threshold** *value* [*event-index*] [**owner** *name*]	Creates an RMON alarm. The value range is from -2147483647 to 2147483647. The owner name can be any alphanumeric string.
	`switch(config)# rmon hcalarm` *index mib-object sample-interval* {**absolute** | **delta**} **rising-threshold-high** *value* **rising-threshold-low** *value* [*event-teindex*] **falling-threshold-high** *value* **falling-threshold-low** *value* *[event-index*] [**owner** *name*] [**storagetype** type]	Creates an RMON high-capacity alarm. The value range is from -2147483647 to 2147483647. The owner name can be any alphanumeric string. The storage type range is from 1 to 5.
Step 3	`switch(config)# show rmon` {**alarms** | **hcalarms**}	(Optional) Displays information about RMON alarms or high-capacity alarms.
Step 4	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

The following example shows how to configure RMON alarms:

```
switch# configure terminal
switch(config)# rmon alarm 1 1.3.6.1.2.1.2.2.1.17.83886080 5 delta rising-threshold 5 1
falling-threshold 0 owner test
switch(config)# copy running-config startup-config
switch# show rmon alarms
Alarm 1 is active, owned by test
 Monitors 1.3.6.1.2.1.2.2.1.17.83886080 every 5 second(s)
 Taking delta samples, last value was 0
 Rising threshold is 5, assigned to event 1
 Falling threshold is 0, assigned to event 0
 On startup enable rising or falling alarm
```

Configuring RMON Events

You can configure RMON events to associate with RMON alarms. You can reuse the same event with multiple RMON alarms.

Ensure you have configured an SNMP user and enabled SNMP notifications (see the "Configuring SNMP" section on page 27-5).

To configure RMON events, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# rmon event` *index* `[description` *string*`] [log] [trap] [owner` *name*`]`	Configures an RMON event. The description string and owner name can be any alphanumeric string.
Step 3	`switch(config)# show rmon {alarms | hcalarms}`	(Optional) Displays information about RMON alarms or high-capacity alarms.
Step 4	`switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

Verifying RMON Configuration

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

Command	Purpose
`show rmon alarms`	Displays information about RMON alarms.
`show rmon events`	Displays information about RMON events.
`show rmon hcalarms`	Displays information about RMON hcalarms.
`show rmon logs`	Displays information about RMON logs.

RMON Example Configuration

The following example creates a delta rising alarm on ifOutOctets and associates a notification event with this alarm:

```
configure terminal
 rmon alarm 1 1.3.6.1.2.1.2.2.1.17.83886080 5 delta rising-threshold 5 1 falling-threshold
0 owner test
 rmon event 1 trap public
```

Default Settings

Table 28-1 lists the default settings for RMON parameters.

Table 28-1 Default RMON Parameters

Parameters	Default
Alarms	None configured.
Events	None configured.

PART 5

FIP Snooping

CHAPTER 29

Configuring FCoE Initialization Protocol Snooping

This chapter describes how to configure the FIP snooping bridge feature and includes the following sections:

Information About FCoE

This section provides information about Fibre Channel over Ethernet (FCoE) and includes the following topics:

Note The BASIC_STORAGE_SERVICES_PKG includes the FIP snooping feature license. The licensing model for the Cisco NX-OS software is feature based. Feature-based licenses make features available to the entire physical switch.

FCoE Overview

This section describes FCoE.

The Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter operates as a loss-less Ethernet bridge transparently forwarding FCoE packets. To satisfy the requirements of a loss-less Ethernet bridge, the following key features are supported:

- No-drop VL
- Priority flow control
- Mini jumbo frames
- Data Center Bridge Exchange Protocol
- FIP Snooping Protocol

- Security ACL
- ETS

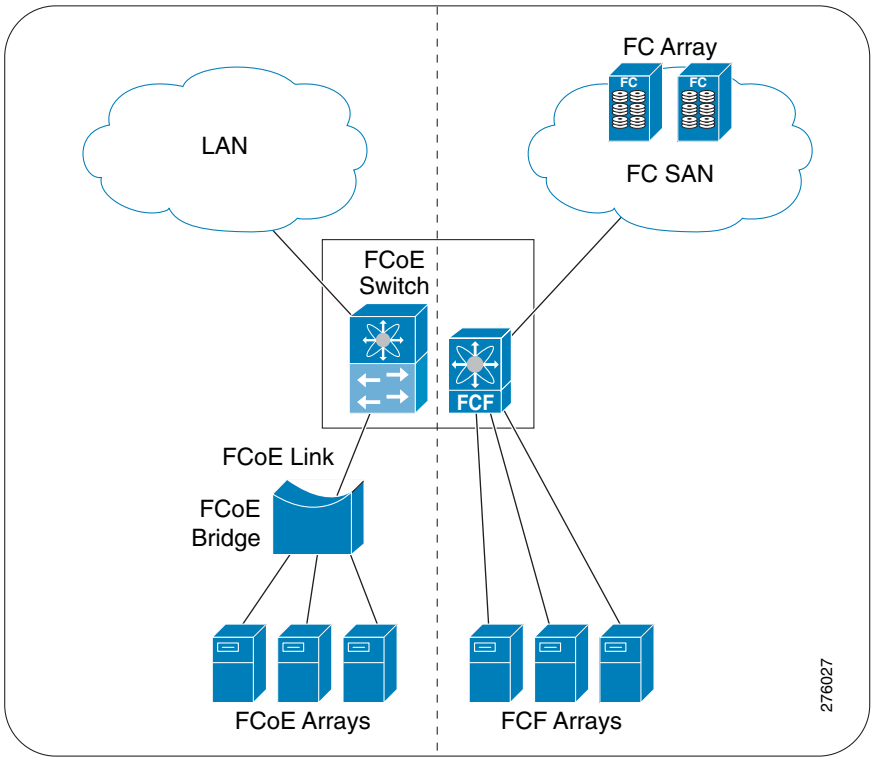
FCoE is an FC-BB-5 in T11 standard developed for FCoE. It primarily provides for IO consolidation on the FCoE- capable host or server that is called an ENode. The FCoE standards discusses two EtherTypes to implement the FCoE functionality. They are FIP (FCoE initialization protocol) and FCoE. The FIP is used to exchange control frames and to discover ENodes and Fibre Channel Forwarders (FCFs) in the network by means of an advertisement-solicitation handshake. The FCoE EtherType is used for data transfer. The standard also describes two methods of addressing ENodes. They are FPMA and SPMA. FPMA (Fabric provided MAC Address) is where the FCF decides the mapped MAC address for the ENode that must be used by the ENode specifically for FCoE. The mapped MAC address is a concatenation of the FC-MAP (originally called the FC-OUI) and the assigned FC-ID for the ENode. Hence FIP frames use the host MAC address while FCoE frames use the FPMA. SPMA (Server provided MAC address) is a case where the MAC address approved for FCoE use can be the same as the MAC address of the server. Therefore, FIP and FCoE frames can be sent from the host with the same MAC address—the host MAC.

Understanding FIP Snooping

This section describes FIP Snooping and its benefits.

In Fibre Channel networks, Fibre Channel switches are generally considered trusted devices. Other Fibre Channel devices must log into the switch before they can communicate with the rest of the fabric. Given that Fibre Channel links are point-to-point, the Fibre Channel switch has complete control over the traffic that a device injects into the fabric or that is received from the fabric. As a result, the switch can ensure that devices are using their assigned addresses and prevent various types of anomalous behaviors that could be erroneous or malicious.

Figure 29-1 Fibre Channel over Ethernet Network Topology



FCoE provides increased flexibility. However, with this flexibility new challenges arise in assuring highly robust fabrics. Specifically, if Ethernet bridges exist between an ENode and the FCF, the point-to-point assurance between ENode and FCF is lost. Thus the FCF does not have the complete authority that a Fibre Channel switch has.

Equivalent robustness between FCoE and Fibre Channel is possible if one can ensure that all FCoE traffic to and from an ENode must pass through an FCF, and that if multiple devices can access an FCF through a single physical FCF port. Doing so, in effect, creates the equivalent of a point-to-point link between the ENode and FCF.

One possible method of accomplishing this is to ensure every ENode is physically connected to an FCF with no intervening Ethernet bridges. Unfortunately, in many deployments this would prove impractical. For example, in large scale blade or 1U server environments, deploying an FCF in each blade system or top-of-rack switch creates the same scaling limitations in FCoE that are well known today in comparably configured Fibre Channel fabrics.

Fibre channel Initialization Protocol (FIP) is an L2 protocol for end point discovery and fabric association. FIP has its own EtherType and uses its own frame formats. There are two phases to FIP, and they are discovery and login. Once the discovery of end nodes and login is complete, FCoE traffic can start flowing between the endpoints. By snooping on FIP packets during the discovery and login phases, intermediate bridges can implement dynamic data integrity mechanisms using ACLs that permit valid FCoE traffic between the ENode and FCF. Implementing such security mechanisms ensures that only valid FCoE traffic is allowed. This is FIP snooping. A bridge implementing the above functionality is what we refer to as the FIP Snooping Bridge. The process implementing this feature is called FIP Snooping Manager (FIPSM). FIPSM is capable of supporting both FPMA and SPMA.

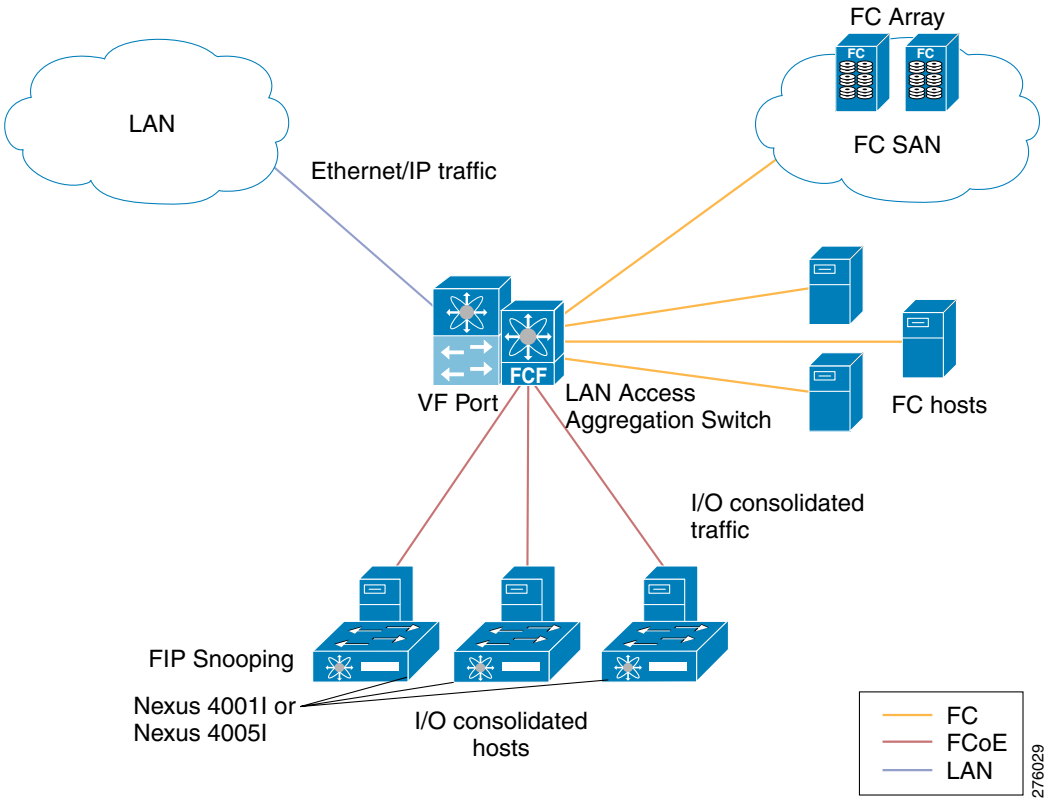
FCoE Connectivity

This section describes options for FCoE connectivity (see Figure 29-2) and includes the following topics:

Non-Redundant FCoE Connectivity

The switch acts as a lossless Ethernet bridge transparently forwarding FCoE packets from the blade servers to a switch. The switch is a FIP snooping bridge.

Figure 29-2 Non-redundant FCoE Connectivity

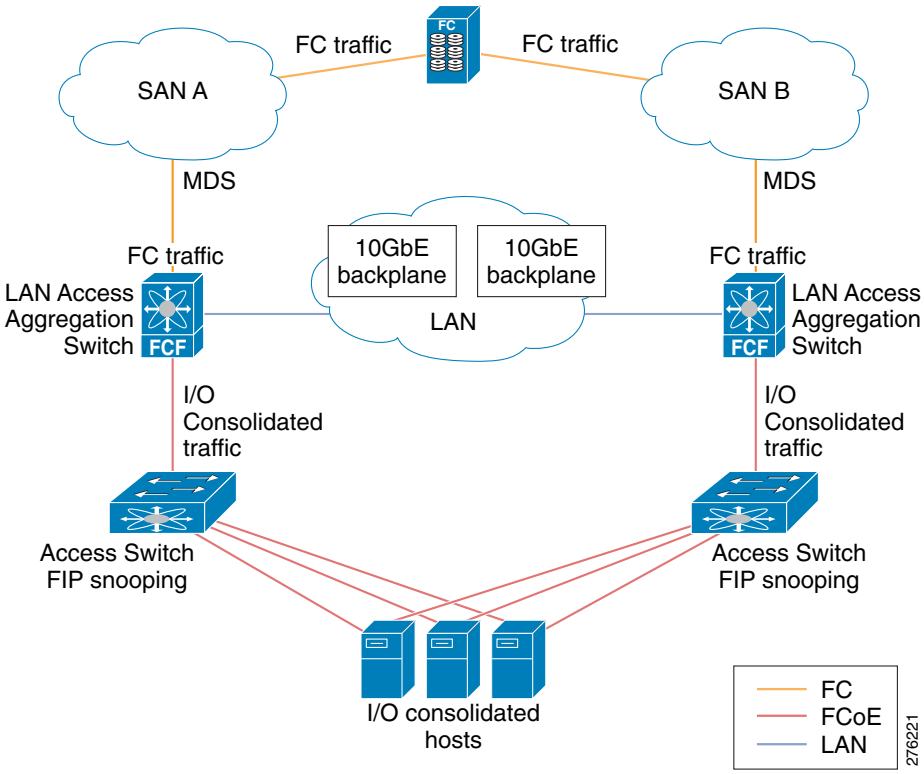


Redundant FCoE Connectivity

The switch acts a lossless Ethernet bridge transparently forwarding FCoE packets from the blade servers to a switch. The switch is a FIP snooping bridge. Each blade server connects to two switches. Each FCF switch connects to a separate switch. Each FCF switch and the LAN Access or Aggregation Switch provides access to a different SAN. See Figure 29-3.

The FCoE Initialization Protocol defined by the T11 standards body enables the host to pick a particular FCF for the fabric login. By using the FIP protocol, the host determines all the available FCFs and then select one from among them.

Figure 29-3 Redundant FCoE Connectivity



Configuring FIP Snooping

When a switch boots up with an empty configuration, it asks the user for a specific configuration. It is also possible to auto-generate or deduce certain configuration, but the user is expected to configure this feature explicitly.

This section includes the following topics:

Enabling DCBXP and LLDP

The Data Center Ethernet Parameter Exchange (DCBXP) is enabled by default. DCBXP is a protocol used to negotiate the FCoE parameters so that the FCoE cloud has end to end auto-configuration for FCoE infrastructure and features. DCBXP uses the standard Link Level Discovery Protocol (LLDP) IEEE standard 802.1ab-2005 to create a bi-directional negotiation path between peer nodes to push FCoE configuration so that FCoE cloud is consistent end to end.

FIPSM interacts with peer using DCBXP to negotiate the following key parameters:

- Priority Flow Control – to exchange per-VL PAUSE configurations
- Priority Scheduling – to exchange bandwidth scheduling and configuration related to priority groups
- FCoE – to exchange FCoE parameters and to determine which VLs should be used by FCoE traffic

There is no specific CLI to enable or disable the DCBXP feature.

LLDP is implemented as part of DCBXP. It is enabled by default. The user can disable it using the **no lldp** command.

Note Disabling the transmit and receive functions of LLDP has a direct impact on the functioning of DCBXP.

To enable or disable LLDP, perform these tasks:

Command	Purpose
`switch (config)# interface` *type slot/port* `switch (config-if)# lldp [transmit | receive]`	Enters interface configuration mode for the specified interface. Enables LLDP.
`switch (config)#` `switch (config-if)# no lldp [transmit | receive]`	Enters interface configuration mode for the specified interface. Disables LLDP.

The following example shows how to enable LLDP transmission on interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/20
switch(config-if)# lldp transmit
```

To display LLDP configuration information on the interface, perform this task:

Command	Purpose
`switch # show lldp [interface type` *slot/port* `| neighbors | timers | traffic]`	Displays the LLDP configuration information.

The following example shows how to display the LLDP configuration information for an Ethernet interface:

```
switch# show lldp interface ethernet 1/20
```

Configuring QoS

QoS must be configured for FCoE before FIP snooping is enabled. MTU, PFC, and ETS are required for FIP snooping. During initial configuration of the switch, QoS is configured by default if you configure FCoE at the time. If you want to change the default QoS configuration, you should configure QoS.

Enabling FIP Snooping Feature

The FIP snooping feature is disabled by default. Only after enabling it, are the FIP related CLIs under VLAN and interface mode visible. The *FIP-snoop* process also starts after the feature is enabled. Until then, the FIP-related packets are treated as normal multicast Ethernet packets with FIP/FcoE EtherType. The CLI is successful only after a cross-check with the license manager. Once the feature is enabled, the FIP-snoop packets and FCoE packets are dropped, unless explicitly enabled on a per-VLAN basis. If FIP snooping is enabled, all the FIP frames are snooped and security ACLs are added. FCoE traffic is blocked on all ports until the device re-initializes with FIP. A warning message for FCoE traffic disruption is issued when enabled. If the feature is disabled, snooping is removed and all programmed ACLs and internal data are cleaned up.

To enable or disable the FIP snooping feature, perform these tasks:

Command	Purpose
`switch (config)# feature fip-snooping`	Enables FIP snooping.
`switch (config)# no feature fip-snooping`	Disables FIP snooping.

The following example shows how to enable the FIP snooping feature:

```
switch# configure terminal
switch(config)# feature fip-snooping
```

Configuring VLAN

VLAN must be configured before it can be used. Once VLAN is enabled, the FIP packets will be snooped only on the configured VLANs. FIP snooping is disabled on VLANs by default.

To enable or disable FIP snooping on a VLAN, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# vlan vlan-id`	Configures specific VLAN port. The range is 1-4095.
Step 3	`switch(config-vlan)# fip-snooping enable`	Enables FIP snooping on a VLAN.
Step 4	`switch(config-vlan)# no fip-snooping enable`	Removes FIP snooping from the VLAN.

The following example shows how to enable FIP snooping for VLAN ID 1-7:

```
switch# configure terminal
switch(config)# vlan 1-7
switch(config-vlan)# fip-snooping enable
```

Configuring VLAN and FC-MAP

The FC-MAP is configured on a per VLAN basis. This FC-MAP is verified with the FC-MAP received from the FCF and if it does not match, the frames are rejected. Only frames that match the configured FC-MAP are allowed to go through and to establish a session between an ENode and FCF. FC-MAP is zero by default.

To configure the VLAN and the FC-MAP, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# vlan vlan-id`	Creates a specific VLAN. The range is 1-4095.
Step 3	`switch(config-vlan)# fip-snooping enable`	Enables the FIP snooping feature.
Step 4	`switch(config-vlan)# fip-snooping fc-map <0x0-0xffffff>`	Configures FC-MAP. **Note** If the FC-MAP is not known, configure it to a definite FC-MAP value of 0x0efc00.

The following example shows how to configure a VLAN and FC-MAP:

```
switch# configure terminal
switch(config)# vlan 101
switch(config-vlan)# fip-snooping enable
switch(config-vlan)# fip-snooping fc-map 0x0efc00
```

Configuring Port Identification

If the FIP snooping feature is enabled and in order to relay the FIP packets from the host to the FCF, the switch needs to know to what interfaces the FCFs are connected. Therefore, the user must specify what is connected to an interface. The FIP Manager keeps track of all interfaces that have FCFs connected, to relay the FIP packets from the hosts. If there is no specific connection information provided, the FIP discovery packets received trigger an identification of the peers connected to the interface. The port is assumed to be in host mode if no user configuration is present.

Note Verify that all the FCoE supporting links to the host or to the FCF are of type trunk and all the FCoE are VLANs.

To configure port identification, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# feature fip-snooping`	Enables fip-snooping
Step 3	`switch(config)# interface type slot/port`	Enters interface configuration mode for the specified interface.
Step 4	`switch(config-if)# fip-snooping port-mode fcf`	Specifies what is connected to the interface.

The following example shows how to configure the FCF for the Ethernet interface slot 1 port 20:

```
switch# configure terminal
switch(config)# feature fip-snooping
switch(config)# interface ethernet 1/20
switch(config-if)# fip-snooping port-mode fcf
```

To configure VLAN characteristics when the interface is in trunking mode, perform this task:

	Command	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)# switchport mode trunk`	Configures the switchport mode trunking parameters.
Step 4	`switch(config-if)# switchport trunk allowed vlan 101`	Sets the allowed VLANs when the interface is in trunking mode.

The following example shows how to set allowed VLANs when the interface is in trunking mode:

```
switch # configure terminal
switch(config)# interface ethernet 1/20
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk allowed vlan 101
```

Verifying FIP Snooping Configuration

To display FIP snooping configuration, perform one of these tasks:

Command	Purpose
`switch# show fip-snooping sessions`	Displays all FIP snooping sessions.
`switch# show fip-snooping fcf`	Displays to what interfaces the FCFs are connected.
`switch# show fip-snooping enode`	Displays the ENode connections.

The following example shows all the FIP snooping sessions:

```
switch# show fip-snooping sessions

  Legend:
--------------------------------------------------------------------------------
 FCF MAC               ENode MAC       VLAN      FCoE MAC          FC ID

--------------------------------------------------------------------------------
00:0d:ec:b2:2c:80    00:0c:29:65:82:bc    1     0e:fc:00:ad:00:00  0x380fdb
```

The following example shows to what interfaces the FCFs are connected:

Note This command must be run for only FCF connected port/s.

```
switch# show fip-snooping fcf

  Legend:
```

```
--------------------------------------------------------------------------------
Interface VLAN    No of    FPMA/  FCMAP       FCF-MAC          NameID  Fabric Name
                  Enodes   SPMA
--------------------------------------------------------------------------------
Eth1/9    1       1        FPMA  0x000000  00:0d:ec:b2:2c:80 00000000  00000000
```

The following example shows the ENode connections:

```
switch# show fip-snooping enode

  Legend:
--------------------------------------------------------------------------------
Interface   VLAN     Name ID        FIP MAC           FCID

--------------------------------------------------------------------------------
Eth1/7      1       00000000   00:0c:29:65:82:bc    0x000000
```

PART 6

Quality of Service

CHAPTER 30

Configuring Quality of Service

This chapter describes the configurable quality of service (QoS) features supported on the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

QoS allows you to classify the network traffic, police and prioritize the traffic flow, and provide congestion avoidance.

This chapter includes the following sections:

Information About QoS Features

The configurable Cisco NX-OS quality of service (QoS) features on the switch allow you to classify the network traffic, police and prioritize the traffic flow, and provide congestion avoidance. The QoS features are configured using Cisco Modular QoS CLI (MQC).

MQC provides a standard set of commands for configuring QoS. You can use MQC to define additional traffic classes and to configure QoS policies for the whole system and for individual interfaces. Configuring a QoS policy with MQC consists of the following steps:

1. Define traffic classes.
2. Associate policies and actions with each traffic class.
3. Attach policies to logical or physical interfaces as well as at the global system level.

MQC provides two command types to define traffic classes and policies:

- Class-map—Defines a class map that represents a class of traffic based on packet-matching criteria. Class maps are referenced in policy maps.

  The class map classifies incoming packets based on matching criteria, such as the IEEE 802.1p CoS value. Unicast and multicast packets are classified.

- Policy-map—Defines a policy map that represents a set of policies to be applied on a class-by-class basis to class maps.

  The policy map defines a set of actions to take on the associated traffic class, such as limiting the bandwidth or dropping packets.

You define the following class-map and policy-map object types when you create them:

- Network-qos—Defines MQC objects that you can use for system level related actions.
- QoS—Defines MQC objects that you can use for classification.
- Queuing—Defines MQC objects that you can use for queuing and scheduling.

You can attach policies to interfaces or EtherChannels as well as at the global system level by using the **service-policy** command. You can view all or individual values for MQC objects by using the **show class-map** and **show policy-map** commands.

An MQC target is an entity (such as an Ethernet interface) that represents a flow of packets. A service policy associates a policy map with an MQC target, and specifies whether to apply the policy on incoming or outgoing packets. This enables the configuration of QoS policies such as marking, bandwidth allocation, buffer allocation, and so on.

The system qos is a type of MQC target. A service policy can associate a policy map with the system qos target. A system qos policy applies to all interfaces on the switch unless a specific interface has an overriding service-policy configuration. The system qos policies are used to define system classes, the classes of traffic across the entire system and their attributes. To ensure QoS consistency (and for ease of configuration), the switch distributes the system class parameter values to all its attached network adapters using the Data Center Bridging Exchange (DCBX) protocol.

If service policies are configured at the interface level, the interface-level policy always takes precedence over system class configuration or defaults.

The qos type is the default for the **class-map** and **policy-map** commands.

Policy Types

This section describes the policy types the switch supports and includes the following topics:

You create class maps in the policy types.

Note Any packet that is not tagged with an 802.1p CoS value is classified into the default drop system class. If the untagged packet is sent over a trunk, it is tagged with the default untagged CoS value, which is zero. After the system applies the untagged CoS value, QoS functions the same as for a packet that entered the system tagged with the CoS value.

Type network-qos

A network-qos policy is used to instantiate system classes and associate parameters with those classes that are of system-wide scope. This section includes the following topics:

Classification

CoS—A class-map of type network-qos identifies a CoS.

Policy

A network-qos policy can only be attached to the system QoS target. The following characteristics can be set under this policy:

1. MTU—The MTU must be enforced for the traffic that is mapped to a system class. Each system class has a default MTU, and the system class MTU is configurable.
2. Pause no-drop—No drop specifies lossless service for the system class. Drop specifies that tail drop is used when a queue for this system class is full. This identifies CoS values to assert priority flow control (PFC) when traffic for a no-drop system class not mapped based purely on CoS experiences congestion.
3. Congestion Control WRED —WRED attempts to anticipate and avoid congestion rather than control congestion once it occurs. You can configure congestion avoidance with WRED in egress policy maps.

Type queuing

A type queuing policy is used to define scheduling characteristics of queues associated with system classes. This section includes the following topics:

- Policy, page 30-4

Classification

CoS—A class-map of type network-qos identifies a CoS.

Policy

These policies can be attached to the system qos target or to any interface. Output queueing policy is used to configure output queues on the switch associated with system classes. The output queuing policy parameters are signaled to the adapter over the DCBX protocol. The following characteristics can be configured under this policy:

Note If you configure any one of the two characteristics (bandwidth or priority), you cannot configure any of the other two characteristics (from among bandwidth, priority, and shaping) in the same policy map.

1. Bandwidth—Sets the guaranteed scheduling deficit weighted round robin (DWRR) percentage for the system class. You can configure the bandwidth and bandwidth remaining on both ingress and egress queues to allocate a minimum percentage of the interface bandwidth to a queue.
2. Priority—Sets a system class for strict-priority scheduling. Only one system class can be configured for priority in an egress priority queue. If you do not specify the priority, the system-defined egress priority queues behave like normal queues. For the non-priority queues, you can configure how much of the remaining bandwidth to assign to each queue. By default, the device evenly distributes the remaining bandwidth among the nonpriority queues.
3. Shaping—Configures shaping on an egress queue to impose a maximum rate on it. You use the system-defined egress queue class. Use the system-defined egress queue class to apply the policy map.

Type qos

A type qos policy is used to classify traffic based on various ACLs and map it to classes. This section includes the following topics:

- Classification, page 30-4
- Policy, page 30-4

Classification

Access control lists—The incoming traffic is classified based on the ACLs.

Policy

This policy can be attached to any interface. It applies to input traffic only.

Cos—Sets the CoS corresponding to the system class which is defined by ACL.

Link-Level Flow Control

The IEEE 802.3x link-level flow control capability allows a congested receiver to communicate the far end to pause its data transmission for a short period of time. The link-level flow control feature applies to all the traffic on the link.

The transmit and receive directions are separately configurable. By default, link-level flow control is disabled for both directions.

On the switch, Ethernet interfaces do not auto-detect the link-level flow control capability. You must configure the capability explicitly on the Ethernet interfaces.

On each Ethernet interface, the switch can enable either priority flow control or link-level flow control (but not both).

Priority Flow Control

The priority flow control (PFC) capability allows you to apply pause functionality to specific classes of traffic on a link (instead of all the traffic on the link). PFC applies pause functionality based on the IEEE 802.1p CoS value. When the switch enables PFC, it communicates to the adapter which CoS values to apply the pause.

Ethernet interfaces use PFC to provide lossless service to no-drop system classes. PFC implements Pause frames on a per-class basis and uses the IEEE 802.1p CoS value to identify the classes that require lossless service.

In the switch, each system class has an associated IEEE 802.1p CoS value (assigned by default or configured on the system class). If PFC is enabled, the switch sends the no-drop CoS values to the adapter, which then applies PFC to these CoS values.

The default CoS value for the FCoE system class is 3 and this value is configurable.

By default, the switch negotiates to enable the PFC capability. If the negotiation succeeds, PFC is enabled and link-level flow control remains disabled (regardless of its configuration settings). If the PFC negotiation fails, you can either force PFC to be enabled on the interface or you can enable IEEE 802.x link-level flow control.

If PFC is not enabled on an interface, you can enable IEEE 802.3X link-level pause. By default, link-level pause is disabled.

MTU

The switch is a Layer 2 switch, and it does not support packet fragmentation. MTU configuration mismatch between ingress and egress interfaces may result in packets being truncated.

When configuring MTU, follow these guidelines:

- MTU is specified per system class. You cannot configure MTU on the interfaces.
- The **system jumbomtu** command defines the upper bound of any MTU in the system. System jumbo MTU has a default value of 9216 bytes. The minimum MTU is 2240 bytes and the maximum MTU is 9216 bytes.
- The system class MTU sets the MTU for all packets in the class. The system class MTU cannot be configured larger than the global jumbo MTU.

- The FCoE system class (for Fibre Channel and FCoE traffic) has a default MTU of 2240 bytes. This value cannot be modified.
- The default drop system class has a default MTU of 1538 bytes. You can configure this value.
- The switch sends the MTU configuration to network adapters that support DCBXP.
- When an MTU size change is made to a system where traffic is flowing, there will be packet drops.

Trust Boundaries

The trust boundary is enforced by the incoming interface as follows:

- By default, all Ethernet interfaces are trusted interfaces. A packet tagged with an 802.1p CoS value is classified into a system class using the value in the packet.
- Any packet that is not tagged with an 802.1p CoS value is tagged with the default untagged CoS value, which is zero.

After the system applies the untagged CoS value, QoS functions in the same way as it does for a packet that entered the system tagged with the CoS value.

Ingress Classification Policies

You can classify traffic by matching packets based on an existing access control list (ACL).

Egress Queuing Policies

You can associate an egress policy map with an Ethernet interface, to guarantee the bandwidth for the specified traffic class or to configure the egress queues.

The bandwidth allocation limit applies to all traffic on the interface.

Each Ethernet interface supports up to eight queues (one for each CoS). The default CoS to queue mapping of each of these queues is listed in Table 30-1. You can also configure a strict priority queue. This queue is serviced before all other queues.

Note The following table lists the default mapping, but CoS to queue mapping can be changed by the user. Also, no-drop traffic can only be mapped to the following three queues: out-q-default, pq1, and out-q2.

Type queuing class maps that are defined by the system are listed in Table 30-1.

Table 30-1 System-Defined Type Queuing Class Maps

Class Map Queue Name	Description	Default CoS Values
1p7q4t-out-pq1	Egress priority queue	1
1p7q4t-out-q2	Egress queue 2	2
1p7q4t-out-q3	Egress queue 3	0
1p7q4t-out-q4	Egress queue 4	4
1p7q4t-out-q5	Egress queue 5	5

Table 30-1 System-Defined Type Queuing Class Maps (continued)

Class Map Queue Name	Description	Default CoS Values
1p7q4t-out-q6	Egress queue 6	6
1p7q4t-out-q7	Egress queue 7	7
1p7q4t-out-q-default	Egress default queue	3

Policy maps that are defined by the system are listed in Table 30-2.

Table 30-2 System-defined Queuing Policy Maps

Queuing Policy Map Name	Description
default-out-policy	policy-map type queuing default-out-policy class type queuing 1p7q4t-out-q3 bandwidth percent 12 class type queuing 1p7q4t-out-pq1 bandwidth percent 12 class type queuing 1p7q4t-out-q2 bandwidth percent 12 class type queuing 1p7q4t-out-q-default bandwidth percent 12 class type queuing 1p7q4t-out-q4 bandwidth percent 12 class type queuing 1p7q4t-out-q5 bandwidth percent 12 class type queuing 1p7q4t-out-q6 bandwidth percent 12 class type queuing 1p7q4t-out-q7 bandwidth percent 12

System-Defined Network QoS Objects

The system-defined network QoS objects are listed in this section.

- policy-map type network-qos p-nq-5e
  - class-map c-nq-5e-drop
    mtu 1538
  - class-map c-nq-5e-ndrop
  - pause no-drop
    mtu 2240

- policy-map type network-qos p-nq-6e
  - class-map c-nq-6e-drop
    mtu 1538
  - class-map c-nq-6e-ndrop
  - pause no-drop
    mtu 2240

- policy-map type network-qos p-nq-8e
  - class-map c-nq-8e
    mtu 1538

QoS for Traffic Directed to the CPU

The switch automatically applies QoS policies to traffic that is directed to the CPU to ensure that the CPU is not flooded with packets. Control traffic, such as BPDU frames, is given higher priority to ensure delivery.

Configuration Guidelines and Limitations

To maintain optimal switch performance, follow these guidelines when configuring system classes and policies:

- One or more CoS are mapped to a queue number (see Table 30-1). A maximum of three no- drop queues can be configured. No-drop traffic can only be mapped to the three queues: out-q-default, pq1, and out-q2. Queue 0-2 can be no-drop only.
- If priority flow control is enabled on an Ethernet interface, pause will never be applied to traffic with a drop system class. PFC does not apply pause to drop classes and the link-level pause feature is never enabled on an interface with PFC.
- If FCoE is enabled in the first setup, CoS 3 is mapped to queue 0, and it belongs to a class that is no-drop.
- When configuring QoS, the no-drop VL, priority grouping, and ETS must be identical for the switch and the Fibre Channel Forwarder (FCF).

When configuring EtherChannel interfaces, note the following guidelines:

- The service policy configured on an EtherChannel applies to all member interfaces.
- The priority flow control configured on an EtherChannel applies to all member interfaces.

Configuring PFC and LLC

This section describes how to configure the Priority Flow Control (PFC) and Link-Level Flow Control (LLC) on Ethernet interfaces supported by the switch and includes the following topics:

- Configuring Priority Flow Control, page 30-9

- Configuring IEEE 802.3x Link-Level Flow Control, page 30-9

The Ethernet interface can operate in two different modes: FCoE mode or standard Ethernet mode.

If the interface is operating in FCoE mode, the Ethernet link is connected at the server port using a converged network adapter (CNA).

If the interface is operating in standard Ethernet mode, the Ethernet link is connected at the server port with a standard Ethernet network adapter (NIC). The network adapter must support DCBX protocol for PFC or ingress policing to be supported on the interface.

Note You must configure a system class with the pause no-drop parameter for PFC to operate on Ethernet traffic (PFC will be applied to traffic that matches the CoS value configured for this class).

Configuring Priority Flow Control

By default, Ethernet interfaces negotiate PFC capability with the network adapter using the DCBX protocol. When PFC is enabled, PFC is applied to traffic that matches the CoS value configured for the no-drop class. You can override the negotiation result by force-enabling the PFC capability.

To force-enable PFC on an interface, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# interface type slot/port`	Specifies the interface to be changed.
Step 3	`switch(config-if)# priority-flow-control mode {auto | on}`	Sets PFC mode for the selected interface. Specify to force-enable PFC. Specify auto to negotiate PFC capability.
Step 4	`switch(config-if)# no priority-flow-control mode on`	(Optional) Disables the PFC setting for the selected interface.

The following example shows how to force-enable PFC on an interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# priority-flow-control mode on
```

Configuring IEEE 802.3x Link-Level Flow Control

By default, link-level flow control capability on Ethernet interfaces is disabled. You can enable link-level flow-control capability for the transmit and receive directions.

To enable link-level flow control capability, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# interface type slot/port`	Specifies the interface to be changed.
Step 3	`switch(config-if)# flowcontrol [receive {on | off} | send {on | off}]`	Enables IEEE 802.3x link-level flow control for the selected interface. Set receive and/or send on or off.
Step 4	`switch(config-if)# no flowcontrol [receive {on | off}] [send {on | off}]`	(Optional) Disables 802.3x link-level flow control for the selected interface.

The following example enables link-level flow control frames on an interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# flowcontrol receive on
```

Configuring System Class Maps

This section describes how to configure system class maps and includes the following topics:

You can create or modify a class map with the **class-map** command. The class map is a names object that represents a class of traffic. In the class map, you specify a set of match criteria for classifying the packets. You can then reference class maps in policy maps.

To configure class maps, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# class-map [type {network-qos | qos | queuing}] class-name`	Creates or accesses a named object that represents a specified class of traffic. Class map names can contain alphabetic, hyphen, or underscrore characters, are case sensitive, and can be up to 40 characters. There are three class-map configuration modes: network-qos—Network-wide (global) mode. CLI prompt: `switch (config-cmap-nq)#` qos—Classification mode; this is the default mode. CLI prompt: `switch (config-cmap-qos)#` queuing—Queuing mode. CLI prompt: `(config-cmap-que)#`
Step 3	`switch(config)# no class-map [type {network-qos | queuing | qos | queuing}] class-name`	(Optional) Deletes the specified class map. **Note** You cannot delete the two system defined class maps, class-fcoe, and class-default.

The following example configures a class map with class name *class1* for type qos:

```
switch# configure terminal
switch(config)# class-map type qos class1
```

Configuring ACL Classification

You can classify traffic by matching packets based on an existing access control list (ACL). Traffic is classified by the criteria defined in the ACL. The permit and deny ACL keywords are ignored in the matching; even if a match criteria in the access-list has a deny action, it is still used for matching for this class.

To configure ACL classification, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# class-map type qos` *class-name*	Creates a named object that represents a class of traffic. Class map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	`switch(config-cmap-qos)# match access-group name` *acl-name*	Configures traffic class by matching packets based on acl-name. The permit and deny ACL keywords are ignored in the matching. **Note** You can only define a single ACL in a class-map.
Step 4	`switch(config-cmap-qos)# no match access-group name` *acl-name*	(Optional) Removes the match from the traffic class.

This example shows how to classify traffic by matching packets based on existing ACLs:

```
switch# configure terminal
switch(config)# class-map type qos class_acl
switch(config-cmap-qos)# match access-group name acl1
```

This example shows how to classify traffic by matching packets after creating MAC ACL:

```
switch# config
switch(config)# mac access-list acl1
switch(config-mac-acl)# permit 0000.aaaa.bbbb 0000.0000.ffff any
switch(config-mac-acl)# exit
switch(config)# class-map type qos class1
switch(config-cmap-qos)# match access-group name acl1
switch(config-cmap-qos)# exit
```

Use the **show class-map** command to display the ACL class map configuration:

```
switch# show class-map class1
```

Configuring CoS Classification

You can classify traffic based on the value of the CoS, which represents a system class. You can set the value of the CoS within a policy map using the **set cos** command.

To configure CoS, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# class-map type {network-qos | queuing} match-any {class-map name | class-map value}`	Creates a named object that represents a class of traffic for type network-qos. For type queuing, only pre-defined class names are supported. Class map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	`switch(config-cmap-que)# match cos cos-value`	Configures the traffic class by matching packets based on CoS values. Values can range from 0 to 7.
Step 4	`switch(config-cmap-qos)# no match cos cos-value`	(Optional) Removes the match from the traffic class.

This example shows how to classify traffic based on the value of the CoS:

```
switch# configure terminal
switch(config)# class-map type queuing match-any 1p7q4t-out-q7
switch(config-cmap-que)# match cos 7
```

Use the **show class-map** command to display the QoS class map configuration:

```
switch# show class-map type queuing 1p7q4t-out-q7
```

Configuring Policy Maps

This section describes how to configure policy maps and includes the following topics:

- Configuring Type Network QoS Policies, page 30-14
- Configuring Type Queuing Policies, page 30-15
- Configuring Type QoS Policies, page 30-16

The following predefined policy maps are used as default service policies:

- network-qos: p-nq-8e
- queuing output: default-out-policy

To create a policy map, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# policy-map [type {network-qos | qos | queuing}] policy-name`	Creates a named object representing a set of policies that are to be applied to a set of traffic classes. Policy map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters. • network-qos—Network-wide (global) mode. CLI prompt: `switch(config-pmap-nq)#` • qos—Classification mode; this is the default mode. CLI prompt: `switch(config-pmap-qos)#` • queuing—Queuing mode. CLI prompt: `switch(config-pmap-que)#`
Step 3	`switch(config)# no policy-map [type {network-qos | qos | queuing}] policy-name`	(Optional) Deletes the specified policy map.
Step 4	`switch(config-pmap)# class [type {network-qos | qos | queuing}] class-name`	Associates a class map with the policy map, and enters configuration mode for the specified system class. There are three class-map configuration modes: • network-qos—Network-wide (global) mode. CLI prompt: `switch(config-pmap-c-nq)#` • qos—Classification mode; this is the default mode. CLI prompt: `switch(config-pmap-c-qos)#` • queuing—Queuing mode. CLI prompt: `switch(config-pmap-c-que)#` The associated class map must be the same type as the policy map type.
Step 5	`switch(config-pmap)# no class [type {network-qos | qos | queuing}] class-name`	(Optional) Deletes the class map association.

This example shows how to configure a policy map after defining an ACL and a class map:

```
switch(config)# mac access-list mac-acl2
switch(config-mac-acl)# permit 0000.aaaa.bbbb 0000.0000.ffff any
switch(config-mac-acl)# exit
switch(config)# class-map type qos my-class
switch(config-cmap-qos)# match access-group name mac-acl2
switch(config-cmap-qos)# exit
switch(config)# policy-map type qos pm-qos-1
switch(config-pmap-qos)# class type qos my-class
switch(config-pmap-c-qos)# set cos 2
switch(config-pmap-c-qos)#
```

Configuring Type Network QoS Policies

Type network qos policies can only be configured on the system qos attachment point. They are applied to the entire system for a particular class.

To configure a type network qos policy, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# policy-map [type {network-qos | qos | queuing}] policy-name`	Creates a named object representing a set of policies that are to be applied to a set of traffic classes. Policy map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	`switch(config-pmap-nq)# class type network-qos class-name`	Associates a class map with the policy map, and enters configuration mode for the specified system class. **Note** The associated class map must be the same type as the policy map type.
Step 4	`switch(config-pmap-nq-c)# {mtu mtu-value | congestion-control random-detect | pause no-drop}`	(Optional) Specifies the MTU value in bytes. **Note** The mtu-value you configure must be less than value set by the **system jumbomtu** command. (Optional) Enables the WRED protocol.
Step 5	`switch(config-pmap-nq-c)# no mtu`	(Optional) Resets the MTU value in this class.
Step 6	`switch(config-pmap-nq-c)# no congestion-control random-detect`	(Optional) Removes the congestion-control random detect feature.
Step 7	`switch(config-pmap-nq-c)# pause no-drop`	Configures a no-drop class. If you do not specify this command, the default policy is drop. The operation for drop policy is simple tail drop, where arriving packets are dropped if the queue increases to its allocated size.
Step 8	`switch(config-pmap-nq-c)# no pause no-drop`	(Optional) Removes the no-drop option from this class.

The following example shows how to define a type network-qos policy map including a class with MTU and pause no-drop and a class with random detect:

```
switch# configure terminal
switch(config)# class-map type network-qos class-que1
switch(config-cmap-nq)# match cos 5
switch(config-cmap-nq)# exit
switch(config)# policy-map type network-qos policy-que1
switch(config-pmap-nq)# class type network-qos class-que1
switch(config-pmap-nq-c)# mtu 5000
switch(config-pmap-nq-c)# pause no-drop
switch(config-pmap-nq-c)# exit
switch(config-pmap-nq)# exit
```

```
switch(config)# class-map type network-qos class-que2
switch(config-cmap-nq)# match cos 4
switch(config-cmap-nq)# exit
switch(config)# policy-map type network-qos policy-que1
switch(config-pmap-nq)# class type network-qos class-que2
switch(config-pmap-nq-c)# congestion-control random-detect
switch(config-pmap-nq-c)#
```

Configuring Type Queuing Policies

Type queuing policies are used for scheduling and buffering the traffic of a specific system class. A type queuing policy is identified by its cos and can be attached to the system or to individual interfaces for output traffic.

To configure a type queuing policy, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# policy-map type queuing` *policy-name*	Creates a named object representing a set of policies that are to be applied to a set of traffic classses. Policy map names can contain alphabetic, hyphen, or or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	`switch(config-pmap-que)# class type queuing` *class-name*	Associates a class map with the policy map, and enters configuration mode for the specified system class.
Step 4	`switch(config-pmap-c-que)# bandwidth` {*<1-10000000000>* {**bps** | **kbps** | **mbps** | **gbps**}| **percent** *value* | **remaining percent** *bandwidth*} | **shape** {*<1-10000000000>* {**bps** | **kbps** | **mbps** | **gbps**}| **average** *rate* | **percent** *rate*}	Specifies the guaranteed percentage of interface bandwidth or shape allocated to this class. By default, no bandwidth or shape is specified for a class. **Note** Before you can successfully allocate bandwidth to the class you must first reduce the default bandwidth configuration on class-default and class-fcoe.
Step 5	`switch(config-pmap-c-que)# no bandwidth` {*<1-10000000000>* {**bps** | **kbps** | **mbps** | **gbps**}| **percent** *value* | **remaining percent** *bandwidth*} | **shape** {*<1-10000000000>* {**bps** | **kbps** | **mbps** | **gbps**}| **average** *rate* | **percent** *rate*}	(Optional) Removes the bandwidth or shape specification from this class.
Step 6	`switch(config-pmap-c-que)# priority`	Specifies that traffic in this class is mapped to a strict priority queue. The priority specification is applicable only with the bandwidth remaining percent option. **Note** Only one class in each policy map can have strict priority set on it.
Step 7	`switch(config-pmap-c-que)# no priority`	(Optional) Removes the strict priority queuing from the traffic in this class.

The following example shows how to define a type queuing policy map that includes a class type with bandwidth as priority and a class type with shape:

```
switch# configure terminal
switch(config)# policy-map type queuing policy-que2
switch(config-pmap-que)# class type queuing 1p7q4t-out-q5
switch(config-pmap-c-que)# bandwidth remaining percent 25
switch(config-pmap-c-que)# priority
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# class type queuing 1p7q4t-out-q4
switch(config-pmap-c-que)# shape percent 35
switch(config-pmap-c-que)#
```

Configuring Type QoS Policies

Type qos policies are used for classifying the traffic of a specific system class identified by a unique qos-group value. A type qos policy can be attached to the system or to individual interfaces for input traffic only.

If the type qos policy is active onany interface or port-channel, you cannot apply a non-default

Note Type qos can only be applied on an interface or port-channel when system network-qos policy is p-nq-8e, which is the default. If the type qos policy is active on any interface or port-channel, we cannot apply non-default network-qos policy to it.

To configure type qos policies that have a pre-defined class type, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# policy-map type qos` *policy-name*	Creates a named object representing a set of policies that are to be applied to a set of traffic classes. Policy map names can contain alphabetic, hyphen, or underscore characters, are case sensitive and can be up to 40 characters.
Step 3	`switch(config-pmap-qos)# class type qos` *class-name*	Associates a class map with the policy map, and enters configuration mode for the specified system class. **Note** The associated class map must be the same type as the policy map type.
Step 4	`switch(config-pmap-c-qos)# set cos` *cos-value*	Configures one of more cos values to match for classification of traffic in this class-map. The range of cos-value is from 0 to 7. There is no default value.
Step 5	`switch(config-pmap-c-qos)# no set cos` *cos-value*	(Optional) Removes the cos values from this class.

The following example shows how to define a type qos policy map after defining a class type:

```
switch# configure
switch(config)# class-map type qos class1
switch(config-cmap-qos)# policy-map type qos policy1
switch(config-pmap-qos)# class type qos class1
switch(config-pmap-c-qos)# set cos 3
switch(config-pmap-c-qos)#
```

Attaching System Service Policy

The **service-policy** command is used to associate the system class policy map as the service policy for the system.

To associate the system policy map as the service policy, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# system qos`	Enters system class configuration mode.
Step 3	`switch(config-sys-qos)# service-policy type {network-qos | qos | queuing} [input | output] policy-name`	Specifies the policy map to use as the service policy for the system. There are three policy map configuration modes: • network-qos—Network-wide (system qos) mode. • qos—Classification mode (system qos input or interface input only). • queuing—Queuing mode (input and output at system qos and interface). **Note** There is no default policy map configuration mode; you must specify the type. The input keyword specifies that this policy map should be applied to traffic received on an interface. The output keyword specifies that this policy map should be applied to traffic transmitted from an interface. You can only apply input to a qos policy; you can apply both to a queuing policy.

The following example sets a no-drop Ethernet policy map as the system class:

```
switch(config)# class-map type network-qos ethCoS4
switch(config-cmap-nq)# match cos 4
switch(config-cmap-nq)# exit
switch(config)# policy-map type network-qos ethNoDrop
switch(config-pmap-nq)# class type network-qos ethCoS4
switch(config-pmap-nq-c)# pause no-drop
switch(config-pmap-nq-c)# exit
switch(config-pmap-nq)# exit
switch(config)# system qos
switch(config-sys-qos)# service-policy type network-qos ethNoDrop
```

Restoring the Default System Service Policies

If you have created and attached new policies to the system qos configuration, you must reapply the default policies to restore the system.

To restore default system service policies, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# system qos`	Enters system class configuration mode.
Step 3	`switch(config-sys-qos)# service-policy type network-qos p-nq-8e`	Resets the network-wide policy map.
Step 4	`switch(config-sys-qos)# service-policy type queuing output default-out-policy`	Resets the output queuing mode policy map.

The following example shows how to reset the system qos configuration:

```
switch# configure terminal
switch(config)# system qos
switch(config-sys-qos)# service-policy type network-qos p-nq-8e
switch(config-sys-qos)# service-policy type queuing output default-out-policy
```

The following example shows the default service policies:

```
switch# show policy-map

  Type queuing policy-maps
  ========================

  policy-map type queuing default-out-policy
    class type queuing 1p7q4t-out-q3
      bandwidth percent 12
    class type queuing 1p7q4t-out-pq1
      bandwidth percent 12
    class type queuing 1p7q4t-out-q2
      bandwidth percent 12
    class type queuing 1p7q4t-out-q-default
      bandwidth percent 12
    class type queuing 1p7q4t-out-q4
      bandwidth percent 12
    class type queuing 1p7q4t-out-q5
      bandwidth percent 12
    class type queuing 1p7q4t-out-q6
      bandwidth percent 12
    class type queuing 1p7q4t-out-q7
      bandwidth percent 12

  Type network-qos policy-maps
  ================================

    policy-map type network-qos p-nq-5e

        class-map  c-nq-5e-drop

        class-map  c-nq-5e-ndrop

        pause no-drop
        mtu 2240

    policy-map type network-qos p-nq-6e

        class-map  c-nq-6e-drop

        class-map  c-nq-6e-ndrop

        pause no-drop
```

```
        mtu 2240

    policy-map type network-qos p-nq-8e

        class-map  c-nq-8e

switch #
```

Enabling Jumbo MTU

To enable jumbo MTU for the entire switch, set the MTU to its maximum size (9216 bytes) in the policy map for the default Ethernet system class (class1).

The following example configures the default Ethernet system class to support the jumbo MTU:

```
switch # configure terminal
switch(config)# class-map type network-qos class1
switch(config-cmap-nq)# match cos 4
switch(config-cmap-nq)# exit
switch(config)# policy-map type network-qos jumbo
switch(config-pmap-nq)# class type network-qos class1
switch(config-pmap-nq-c)# mtu 9216
switch(config-pmap-nq-c)# exit
switch(config-pmap-nq)# exit
switch(config)# system qos
switch(config-sys-qos)# service-policy type network-qos jumbo
```

Note The **system jumbomtu** command defines the maximum MTU size for the switch. However, jumbo MTU is only supported for system classes that have MTU configured.

Configuring QoS on Interface Policy

An input qos policy is a service policy applied to incoming traffic on an Ethernet interface for classification. For type queuing, the output policy is applied to all outgoing traffic that matches the specified class. When you configure an input queuing policy on an interface or EtherChannel, the switch sends the configuration data to the adapter using the DCBX protocol.

To configure the interface service policy, perform this task:

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# interface {ethernet` *slot/port* `| port-channel` *channel-number*`}`	Enters configuration mode for the specified interface. **Note** The service policy on a port channel applies to all member interfaces.

	Command	Purpose
Step 3	`switch(config-if)# service-policy [type {qos | queuing}] [input | output] policy-name`	Specifies the policy map to use as the service policy for the system. There are two policy map configuration modes: • qos—Classification mode; this is the default mode. • queuing—Queuing mode. **Note** The input keyword specifies that this policymap should be applied to traffic received on an interface. The output keyword specifies that this policy map should be applied to traffic transmitted from an interface. You can only apply input to a qos policy; you can apply both to a queuing policy.
Step 4	`switch(config-if)# service-policy [type {qos | queuing}] [input] policy-name`	Applies the policy map to the interface. **Note** There is a restriction that system type qos policy cannot be the same as any of the type qos policy applied to an interface or EtherChannel.

The following example shows how to apply a policy to an Ethernet interface:

```
switch# configure terminal
switch(config)# class-map type qos class-qos-1
switch(config-cmap-qos)# exit
switch(config)# policy-map type qos policy-qos-1
switch(config-pmap-c-qos)# exit
switch(config-pmap-qos)# exit
switch(config)# interface ethernet 1/20
switch(config-if)# service-policy type qos input policy-qos-1
```

QoS Configuration Examples

This section displays some QoS configuration examples and includes the following topics:

Using Access Control List to Ethernet Traffic Configuration Example

The following example shows how to configure traffic in the entire system matching an access control list to have the frame CoS fields rewritten to the value 5:

1. Set up the ingress classification policy:

```
switch# configure terminal
switch(config)# mac access-list acl1
switch(config-mac-acl)# permit 0000.aaaa.bbbb 0000.0000.ffff any
switch(config-mac-acl)# exit
switch(config)# class-map type qos class1
switch(config-cmap-qos)# match access-group name acl1
switch(config-cmap-qos)# exit
switch(config)# policy-map type qos pmap-qos-acl1
switch(config-pmap-qos)# class class1
switch(config-pmap-c-qos)# set cos 5
switch(config-pmap-c-qos)# exit
switch(config-pmap-qos)# exit
```

2. Attach the classification policy to the interface Ethernet 1/1:

```
switch(config)# interface ethernet 1/1
switch(config-if)# service-policy type qos input pmap-qos-acl1
switch(config-if)# exit
```

Using Queuing for Bandwidth Configuration Example

The following example shows how to use queuing to configure bandwidth:

1. Set up the system-wide definition of the cos first:

```
switch(config)# class-map type queuing match-any 1p7q4t-out-q2
switch(config-cmap-que)# match cos 2
switch(config-cmap-que)# exit
```

2. Set up the egress bandwidth policy:

```
switch(config)# policy-map type queuing pmap-que-eth1-2
switch(config-pmap-que)# class type queuing 1p7q4t-out-q-default
switch(config-pmap-c-que)# bandwidth percent 40
switch(config-pmap-c-que)# class type queuing 1p7q4t-out-q6
switch(config-pmap-c-que)# bandwidth remaining percent 60
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# exit
```

3. Attach it to the egress interface:

```
switch(config)# interface ethernet 1/3
switch(config-if)# service-policy type queuing output pmap-que-eth1-2
switch(config-if)# exit
```

Setting MTU with Network QoS Example

1. Allocate the system class for cos 2:

```
switch(config)# class-map type network-qos cmap_nq1
switch(config-cmap-nq)# match cos 2
switch(config-cmap-nq)# exit
```

2. Set up the network-qos policy:

```
switch(config)# policy-map type network-qos pmap_nq1
switch(config-pmap-nq)# class type network-qos cmap_nq1
switch(config-pmap-nq-c)# mtu 5000
switch(config-pmap-nq-c)# exit
switch(config-pmap-nq)# exit
```

3. Attach the network-qos policy to the system:

```
switch(config)# system qos
switch(config-sys-qos)# service-policy type network-qos pmap_nq1
switch(config-sys-qos)# exit
```

Priority Configuration Example

The following example shows how to configure the priority:

```
switch(config)# class-map type queuing match-any 1p7q4t-out-q3
switch(config-cmap-que)# match cos 5-7
switch(config-cmap-que)# class-map type queuing match-any 1p7q4t-out-q2
switch(config-cmap-que)# match cos 3-4
switch(config-cmap-que)# class-map type queuing match-any 1p7q4t-out-q4
switch(config-cmap-que)# match cos 0-2
switch(config-cmap-que)# policy-map type queuing pm_que1
switch(config-pmap-que)# class type queuing 1p7q4t-out-q3
switch(config-pmap-c-que)# bandwidth remaining percent 20
switch(config-pmap-c-que)# class type queuing 1p7q4t-out-q2
switch(config-pmap-c-que)# priority
switch(config-pmap-c-que)# class type queuing 1p7q4t-out-q4
switch(config-pmap-c-que)# bandwidth remaining percent 80
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# exit
switch(config)# system qos
switch(config-sys-qos)# service-policy type queuing output pm_que1
switch(config-sys-qos)#
```

Shaping Configuration Example

The following example shows how to configure the shaping features:

```
switch(config)# class-map type queuing match-any 1p7q4t-out-q7
switch(config-cmap-que)# match cos 5-7
switch(config-cmap-que)# class-map type queuing match-any 1p7q4t-out-q6
switch(config-cmap-que)# match cos 3-4
switch(config-cmap-que)# policy-map type queuing shape1
switch(config-pmap-que)# class type queuing 1p7q4t-out-q7
switch(config-pmap-c-que)# shape percent 50
switch(config-pmap-c-que)# class type queuing 1p7q4t-out-q6
switch(config-pmap-c-que)# shape percent 25
switch(config-pmap-c-que)#
```

Note If the priority keyword is not specified for a **pq1** queue, the queue is just a normal queue, not a priority queue.

Verifying QoS Configuration

To verify QoS configuration information, perform one of these tasks:

Command	Purpose
`switch # show class-map`	Displays the class maps defined on the switch.
`switch # show policy-map`	Displays the policy maps defined on the switch.

Command	Purpose
`switch # show policy-map system`	Displays the policy map settings attached to the system qos.
`switch # show policy-map interface [interface number]`	Displays the policy map settings for an interface or all interfaces.
`switch # show queuing interface [interface number]`	Displays the queue configuration and statistics.

To clear the QoS policy statistics, perform one of these tasks:

Command	Purpose
`switch # clear qos statistics [interface [ethernet slot/port | input type qos | port-channel number | type qos]]`	Clears the policy statistics for type qos packet counteres.
`switch # clear statistics device all`	Clears all counters. **Note** This command clears all queuing counters including other non-QoS packet counters.

The following example shows how to display the class maps defined on the switch:

```
switch # show class-map
Type queuing class-maps
  ========================
    class-map type queuing match-any 1p7q4t-out-pq1
      Description: Classifier for egress priority queue of type 1p7q4t
      match cos 1
    class-map type queuing match-any 1p7q4t-out-q2
      Description: Classifier for egress queue 2 of type 1p7q4t
      match cos 2
    class-map type queuing match-any 1p7q4t-out-q3
      Description: Classifier for egress queue 3 of type 1p7q4t
      match cos 0
    class-map type queuing match-any 1p7q4t-out-q4
      Description: Classifier for egress queue 4 of type 1p7q4t
      match cos 4
    class-map type queuing match-any 1p7q4t-out-q5
      Description: Classifier for egress queue 5 of type 1p7q4t
      match cos 5
    class-map type queuing match-any 1p7q4t-out-q6
      Description: Classifier for egress queue 6 of type 1p7q4t
      match cos 6
    class-map type queuing match-any 1p7q4t-out-q7
      Description: Classifier for egress queue 7 of type 1p7q4t
      match cos 7
    class-map type queuing match-any 1p7q4t-out-q-default
      Description: Classifier for egress default queue of type 1p7q4t
      match cos 3
```

```
  Type network-qos class-maps
  ===============================
    class-map type network-qos c-nq-8e
      match cos 0-7
    class-map type network-qos c-nq-5e-drop
      match cos 3-7
    class-map type network-qos c-nq-6e-drop
      match cos 2-7
    class-map type network-qos c-nq-5e-ndrop
      match cos 0-2
    class-map type network-qos c-nq-6e-ndrop
      match cos 0-1
```

The following example shows how to display the policy maps defined on the switch:

```
switch # show policy-map
  Type queuing policy-maps
  ========================

  policy-map type queuing default-out-policy
    class type queuing 1p7q4t-out-q3
      bandwidth percent 12
    class type queuing 1p7q4t-out-pq1
      bandwidth percent 12
    class type queuing 1p7q4t-out-q2
      bandwidth percent 12
    class type queuing 1p7q4t-out-q-default
      bandwidth percent 12
    class type queuing 1p7q4t-out-q4
      bandwidth percent 12
    class type queuing 1p7q4t-out-q5
      bandwidth percent 12
    class type queuing 1p7q4t-out-q6
      bandwidth percent 12
    class type queuing 1p7q4t-out-q7
      bandwidth percent 12

  Type network-qos policy-maps
  ================================
    policy-map type network-qos p-nq-5e
        class-map  c-nq-5e-drop
        class-map  c-nq-5e-ndrop
        pause no-drop
        mtu 2240
    policy-map type network-qos p-nq-6e
        class-map  c-nq-6e-drop
        class-map  c-nq-6e-ndrop
        pause no-drop
        mtu 2240
    policy-map type network-qos p-nq-8e
        class-map  c-nq-8e
switch #
```

The following example shows how to display the policy maps attached on the system qos:

```
switch # show policy-map system
  Type network-qos policy-maps
```

```
  ================================
    policy-map type network-qos p-nq-8e
        class-map  c-nq-8e
              match cos 0-7
  Service-policy (queuing) output:   default-out-policy
    policy statistics status:   enabled
    Class-map (queuing):   1p7q4t-out-q3 (match-any)
      bandwidth percent 12
    Class-map (queuing):   1p7q4t-out-pq1 (match-any)
      bandwidth percent 12
    Class-map (queuing):   1p7q4t-out-q2 (match-any)
      bandwidth percent 12
    Class-map (queuing):   1p7q4t-out-q-default (match-any)
      bandwidth percent 12
    Class-map (queuing):   1p7q4t-out-q4 (match-any)
      bandwidth percent 12
    Class-map (queuing):   1p7q4t-out-q5 (match-any)
      bandwidth percent 12
    Class-map (queuing):   1p7q4t-out-q6 (match-any)
      bandwidth percent 12
    Class-map (queuing):   1p7q4t-out-q7 (match-any)
      bandwidth percent 12
switch#
```

The following example shows how to display the policy maps attached to an interface:

```
switch(config-if)# show policy-map interface ethernet 1/15
Global statistics status :   enabled
Ethernet1/15
  Service-policy (queuing) output:   default-out-policy
    policy statistics status:   enabled
    Class-map (queuing):   1p7q4t-out-q3 (match-any)
    Class-map (queuing):   1p7q4t-out-pq1 (match-any)
    Class-map (queuing):   1p7q4t-out-q2 (match-any)
    Class-map (queuing):   1p7q4t-out-q-default (match-any)
    Class-map (queuing):   1p7q4t-out-q4 (match-any)
    Class-map (queuing):   1p7q4t-out-q5 (match-any)
    Class-map (queuing):   1p7q4t-out-q6 (match-any)
    Class-map (queuing):   1p7q4t-out-q7 (match-any)
```

The following example shows how to display information related to the counters by entering the **show queuing interface** command:

Note In the following output, the display following *Statistics for CoS* provides information related to the counters. When you use the **clear statistics device all** command, the details about counters is cleared. The **clear statistics device all** command clears all queuing counters including other non-QoS packet counters, while the **clear qos statistics** command clears policy statistics specified for type qos packet counters only.

```
switch # show queuing interface ethernet 1/18
Queuing Configs for interface 0x1a011000(System)
Port QOS is enabled
Transmit queues [type = 1p7q4t]
Queue ID                                COS     Scheduling
-----------------------------------------------------------

1p7q4t-out-q3                            0       WRR
1p7q4t-out-pq1                           1       WRR
1p7q4t-out-q2                            2       WRR
1p7q4t-out-q-default                     3       WRR
1p7q4t-out-q4                            4       WRR
1p7q4t-out-q5                            5       WRR
1p7q4t-out-q6                            6       WRR
1p7q4t-out-q7                            7       WRR

Queue ID                                B/W     Shape
-----------------------------------------------------------

1p7q4t-out-q-default        BW percent Gua:30, Rem:0, RemReal:0
       -
1p7q4t-out-pq1              BW percent Gua:20, Rem:0, RemReal:0
       -
1p7q4t-out-q2               BW percent Gua:10, Rem:0, RemReal:0
       -
1p7q4t-out-q3               BW percent Gua:5, Rem:0, RemReal:0
       -
1p7q4t-out-q4               BW percent Gua:10, Rem:0, RemReal:0
       -
1p7q4t-out-q5               BW percent Gua:15, Rem:0, RemReal:0
       -
1p7q4t-out-q6               BW percent Gua:5, Rem:0, RemReal:0
       -
1p7q4t-out-q7               BW percent Gua:5, Rem:0, RemReal:0
       -

-----------------------------------------------------------
Queuing configs for 1p7q4t-out-q-default**
Group-id: 0
Flags: 0x1
BW percent Gua:30, Rem:0, RemReal:0
Queuing configs for 1p7q4t-out-pq1**
Group-id: 1
Flags: 0x1
BW percent Gua:20, Rem:0, RemReal:0
Queuing configs for 1p7q4t-out-q2**
Group-id: 2
Flags: 0x1
BW percent Gua:10, Rem:0, RemReal:0
Queuing configs for 1p7q4t-out-q3**
Group-id: 3
Flags: 0x1
BW percent Gua:5, Rem:0, RemReal:0
Queuing configs for 1p7q4t-out-q4**
```

```
Group-id: 4
Flags: 0x1
BW percent Gua:10, Rem:0, RemReal:0
Queuing configs for 1p7q4t-out-q5**
Group-id: 5
Flags: 0x1
BW percent Gua:15, Rem:0, RemReal:0
Queuing configs for 1p7q4t-out-q6**
Group-id: 6
Flags: 0x1
BW percent Gua:5, Rem:0, RemReal:0
Queuing configs for 1p7q4t-out-q7**
Group-id: 7
Flags: 0x1
BW percent Gua:5, Rem:0, RemReal:0

Statistics per COS
COS  Packet(RX)  Packet(TX)   Pause(RX)   Pause(TX)
---------------------------------------------------------
0             0          98           0           0
1             0      552000           0           0
2    4708981456           0           0     1909911
3            97   628608570           0  2505454392
4             0           0           0           0
5             0           0           0           0
6             0           0           0           0
7             0           0           0           0
```

Verifying Jumbo MTU

To verify that jumbo MTU is enabled, enter the **show interface ethernet slot/port** command for an Ethernet interface that carries traffic with jumbo MTU.

The following example shows how to display summary jumbo MTU information for Ethernet 1/2 (the relevant part of the output is shown in bold font):

```
switch# show interface ethernet 1/2
Ethernet1/2 is up
...
Rx
1547805598 Input Packets 1547805596 Unicast Packets 0 Multicast Packets
0 Broadcast Packets 1301767362 Jumbo Packets 33690 Storm Suppression Packets
7181776513802 Bytes
Tx
1186564478 Output Packets 7060 Multicast Packets
0 Broadcast Packets 997813205 Jumbo Packets
4813632103603 Bytes
...
The following example shows how to display detailed jumbo MTU information for Ethernet 1/2
(the relevant
part of the output is shown in bold font):
switch# show interface ethernet 1/2 counters detailed
Rx Packets: 1547805598
Rx Unicast Packets: 1547805596
Rx Jumbo Packets: 1301767362
Rx Bytes: 7181776513802
Rx Storm Suppression: 33690
Rx Packets from 0 to 64 bytes: 169219
Rx Packets from 65 to 127 bytes: 10657133
Rx Packets from 128 to 255 bytes: 21644488
Rx Packets from 256 to 511 bytes: 43290596
Rx Packets from 512 to 1023 bytes: 86583071
```

```
Rx Packets from 1024 to 1518 bytes: 83693729
Rx Trunk Packets: 1547805596
Tx Packets: 1186564481
Tx Unicast Packets: 1005445334
Tx Multicast Packets: 7063
Tx Jumbo Packets: 997813205
Tx Bytes: 4813632103819
Tx Packets from 0 to 64 bytes: 137912
Tx Packets from 65 to 127 bytes: 8288443
Tx Packets from 128 to 255 bytes: 16596457
Tx Packets from 256 to 511 bytes: 33177999
Tx Packets from 512 to 1023 bytes: 66363944
Tx Packets from 1024 to 1518 bytes: 64186521
Tx Trunk Packets: 1005451729
```

PART 7

IBM BladeCenter-Specific Features

CHAPTER 31

SoL Features and Concepts and Configuring CIN

This chapter describes features and concepts for Serial over LAN (SoL) and Chassis Internal Network (CIN) VLAN configuration for the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter and includes the following sections:

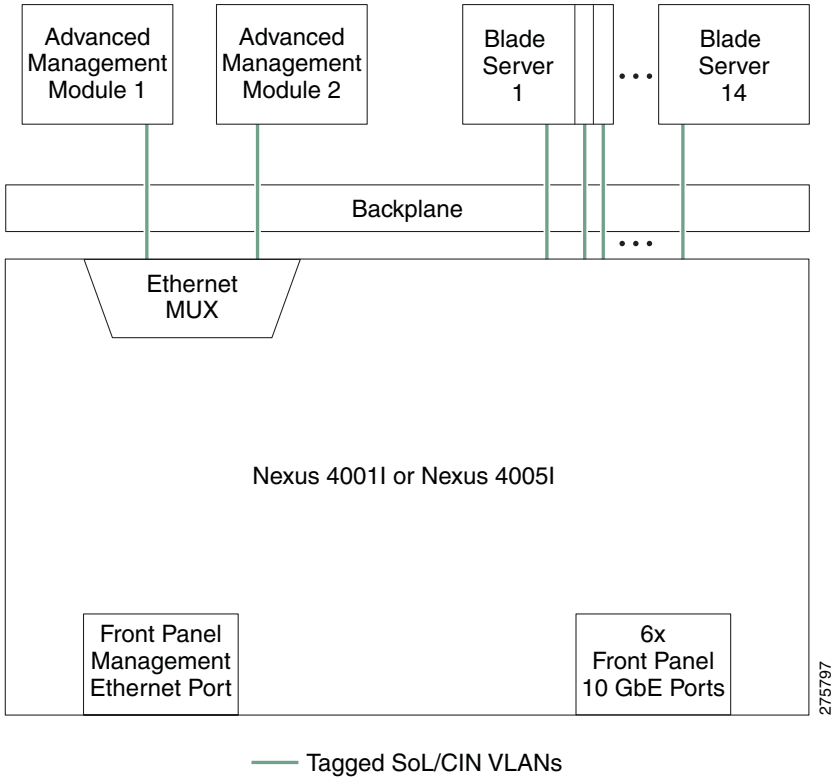
Information About Serial over LAN Management VLAN

This section describes the SoL management VLAN features.

The switch is managed through the Advanced Management Module (AMM) in the chassis as well as from the management port on the front faceplate.

The 14 (12 for IBM Blade Center HT) Blade Servers hosted on a chassis are also managed through the AMM. The switch provides the switching for traffic between the Ethernet interface on the Blade Servers and the AMM. For an overview of the SoL management VLAN feature, see Figure 31-1.

Figure 31-1 SoL Management VLAN Feature Overview



- All unicast, multicast, and broadcast traffic between the IBM-AMM and the Blade Servers are switched at Layer 2.
- The IBM-AMM and the Blade Servers communicate over a VLAN configured by the administrator. The default VLAN is 4095.

Note In this release, the default VLAN is 4095, and it is always enabled.

- Apart from the SoL management VLANs, 14 other CIN VLANs can be configured on specific ports for communication between the AMM and the corresponding Blade Servers.
- VLAN tagged frames are relayed between the Blade Servers and AMM without any changes.
- To avoid disruption of traffic between the AMM and the Blade Servers, this feature cannot be turned off.
- No management traffic is switched to and from the uplink ports and the out-of-band management port that are located on the front panel of the switch.

Configuration Restrictions

This section lists the restrictions to note during configuration.

- The user interface for the switch to support CIN VLANs is through the IBM-AMM. The CIN VLAN can also be configured using the CLI of the out-of-band management port on the front panel of the switch.
- The switch supports 10Mbps of SoL, cKVM, and CIN traffic.
- For SoL and CIN traffic to be switched correctly between the downlink ports (backplane ports) and the backplane management port, the corresponding downlink ports must be set up in the trunkmode (switchport mode trunk) as a prerequisite.

CIN VLAN Configuration

This section shows how to configure CIN VLAN. It includes the following topics:

Associating a VLAN as a CIN VLAN

To configure CIN VLAN, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# vlan {vlan-id | vlan-range}`	Enters VLAN submode.
Step 3	`switch(config-vlan)# cin`	Associates the VLAN as CIN VLAN.

The following example shows how to associate a VLAN as a CIN VLAN:

```
switch# configure terminal
switch(config)#
switch(config)# vlan 3
switch(config-vlan)#
switch(config-vlan)# cin
switch(config-vlan)#
```

Disassociating a CIN VLAN

To disassociate a CIN VLAN, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# vlan {vlan-id | vlan-range}`	Enters VLAN submode.
Step 3	`switch(config-vlan)# no cin`	Disassociates the CIN VLAN.

The following example shows how to disassociate a CIN VLAN:

```
switch# configure terminal
switch(config)#
switch(config)# vlan 3
switch(config-vlan)#
switch(config-vlan)# no cin
switch(config-vlan)#
```

Deleting a VLAN Configured as CIN

To delete a CIN VLAN, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# no vlan {vlan-id | vlan-range}`	Deletes the CIN VLAN configuration.

The following example shows how to delete a configured CIN VLAN:

```
switch# configure terminal
switch(config)#
switch(config)# vlan 3
switch(config-vlan)#
switch(config-vlan)# cin
switch(config-vlan)#
switch(config-vlan)# no vlan 3
switch(config)#
```

Adding CIN VLAN Configuration on a Port

To add CIN VLAN on an existing trunk VLAN port configuration, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# interface ethernet slot/port`	Enter interface configuration mode. The port numbers can be 1-20.
Step 3	`switch(config-if)# switchport trunk allowed vlan add vlan id`	Add the CIN VLAN to the allowed VLAN list for an interface. The VLAN IDs when this port is in trunking mode can be 1-3967, 4048-4093.

The following example shows how to add CIN VLAN configuration on an Ethernet port:

```
switch# configure terminal
```

```
switch(config)# interface ethernet 1/20
switch(config-if)# switchport trunk allowed vlan add 3967
switch(config-if)#
```

Deleting CIN VLAN

To delete CIN VLAN from an existing trunk VLAN port configuration, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# interface ethernet` *slot/port*	Enters interface configuration mode.
Step 3	`switch(config-if)# switchport trunk allowed vlan remove` *vlan id*	Removes the CIN VLAN from the allowed VLAN list for an interface.

The following example shows how to delete CIN VLAN from an existing trunk VLAN port configuration:

```
switch# configure terminal
switch(config)# interface ethernet 1/20
switch(config-if)# switchport trunk allowed vlan remove 3967
switch(config-if)#
```

Verifying CIN VLAN Configuration

This section includes the following topics:

Displaying the CIN VLAN Association

To display a CIN VLAN association, perform this task:

	Command	Purpose
Step 1	`switch# show solm cin`	Displays the CIN VLANs association.

The following is sample output from the **show solm cin** command:

```
switch# show solm cin
20-24,26-33,100
```

Viewing SoL and CIN Traffic Counters

To view SoL and CIN traffic counters, perform this task:

	Command	Purpose
Step 1	`switch# show solm counters [interface ethernet slot/port | mgmt port]`	Displays the SoL and CIN unicast and multicast counters associated with interfaces.

The following is sample output from the **show solm counters** command:

```
switch# show solm counters
------------------------------------------------------------------------
Interface           TX-UCast      TX-MCast      RX-UCast      RX-MCast
------------------------------------------------------------------------
mgmt1               0             0             0             0
Ethernet1/1         0             0             0             0
Ethernet1/2         0             0             0             0
Ethernet1/3         0             0             0             0
Ethernet1/4         0             0             0             0
Ethernet1/5         0             0             0             0
Ethernet1/6         0             0             0             0
Ethernet1/7         0             0             0             0
Ethernet1/8         0             0             0             0
Ethernet1/9         0             0             0             0
Ethernet1/10        0             0             0             0
Ethernet1/11        0             0             0             0
Ethernet1/12        0             0             0             0
Ethernet1/13        0             0             0             0
Ethernet1/14        0             0             0             0
```

CHAPTER 32

Configuring Protected Mode

This chapter describes the protected mode feature supported on the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter and includes the following sections:

About Protected Mode

By default, protected mode is disabled, and the BladeCenter chassis AMM controls the switch. You can enable protected mode to prevent AMM from controlling the switch. By locking out the AMM from control of the switch, server administrators cannot manage the switch from the AMM. When protected mode is enabled, the AMM cannot control or configure the following features and functions of the switch:

- IP addresses
- Administration of external ports
- Managing the switch with traffic received over external ports
- Preventing the switch from reverting to the manufacturing default configuration

Note To prevent physical damage to the switch, the AMM can still reboot or power off the switch when the switch is in protected mode and an over-temperature or over-current condition is detected by the AMM.

These guidelines and restrictions apply to protected mode:

- Protected mode must be enabled on the AMM before you enter this command on the switch. For information about enabling protected mode on the AMM, see the documentation provided with your AMM product.
- After protected mode is operational on the switch, the AMM cannot configure or administer the switch.
- The switch must be rebooted for protected mode to become operational.
- Protected mode remains active even when the switch is moved to another chassis.
- Recovery from lost passwords requires direct access through the external serial port on the switch.

Configuring Protected Mode

To enable protected mode and prevent the AMM from controlling the switch, perform the following task:

Note Protected mode must be enabled on the AMM before you enter this command on the switch. For information about enabling protected mode on the AMM, see the documentation provided with your AMM product.

	Command	Purpose
Step 1	`switch # configure terminal`	Enters configuration mode.
Step 2	`switch(config)# platform chassis-management protected-mode`	Enable the switch to block control from the AMM.
Step 3	`switch (config)# copy running-config startup-config`	Copy running-configuration to startup configuration.
Step 4	`switch(config)# reload`	Reload the switch. **Note** Wait for the software to complete reloading.
Step 5	`switch(config)# exit`	Return to global configuration mode.
Step 6	`switch # end`	Return to privileged EXEC mode.

The following example shows how to configure protected mode on the switch after it has been enabled on the AMM:

```
switch(config)# platform chassis-management protected-mode
The switch has been configured to disallow the Chassis management module from operational
control. The switch must be rebooted for this feature to take effect. Use the "reload"
command from CLI.
switch (config)# copy running-config startup-config
[########################################] 100%
switch(config)# reload
WARNING: There is unsaved configuration!!!
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
switch(config)# exit
switch # end
```

To disable protected mode and return control of the switch to the AMM, enter the **no platform chassis-management protected-mode** and reboot the switch. Then, disable protected mode from AMM.

Verifying Protected Mode

To verify that protected mode is enabled, perform this task:

Command	Purpose
`switch # show chassis summary`	Verify that protected mode is enabled on the next reboot. **Note** After rebooting the switch, enter the **show chassis summary** user EXEC command to verify that protected mode is operational.

The following example shows how to verify that protected mode has been configured on the switch:

Note *MM Prot Mode Support* indicates *yes* in the following example. This shows that protected mode is configured on the AMM.
MM Prot Mode Config indicates *yes* in the following example. This shows that protected mode is configured on the switch.

```
switch# show chassis summary
Switch Slot ID          : 10
Chassis type            : IBM BladeCenter   BC-H
Chassis ID              : KQFXBLB
Active MM in Slot       : 1

MM Stack Mode Support   : No
MM Prot Mode Support    : Yes
MM Prot Mode Status     : Operational
MM Prot Mode Config     : yes
Ext Mgmt/Ext Ports      : Disabled

Switch IP Acquisition   : static
Amm IP Acquisition      : static

VPD Def       IP  Addr  : 10.0.0.1
VPD Def       IP  Mask  : 255.255.255.0
VPD Def      Gateway    : 0.0.0.0

VPD Curr      IP  Addr  : 10.0.0.1
VPD Curr      IP  Mask  : 255.255.255.0
VPD Curr     Gateway    : 0.0.0.0
```

CHAPTER 33

Wake on LAN Feature

This chapter describes the Wake on Lan (WOL) feature.

Wake On LAN (WOL) is a combination of hardware and software technologies to wake up sleeping systems. WOL is enabled or disabled for each individual switch using AMM. You must also enable WOL on the NIC of the server that is connected to the switch.

Note Confirm that WOL is enabled for a particular switch in the IBM BladeCenter and on the NIC of the server, before you perform a WOL function on that switch.

WOL uses specially coded network packets, called magic packets, to systems equipped and enabled to respond to these packets. WOL is based on the principle that when the server blade in the IBM BladeCenter chassis shuts down, the NIC still receives power and operates in the 1 Gb mode, and keeps listening on the network for the magic packet to arrive that wakes up the server. When the server wakes up, the NIC transitions from 1 Gb to 10 Gb mode for normal operation.

When the NIC transitions from 10 Gb to 1 Gb and from 1 Gb to 10 Gb, the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter auto-negotiates with the NIC to operate in the corresponding mode. The switch provides 1 Gb/10 Gb internal support for auto-negotiation of WOL and transparently transports the magic packet to the server blades.

PART 8

Troubleshooting

CHAPTER 34

Configuring SPAN

The Switched Port Analyzer (SPAN) feature (sometimes called port mirroring or port monitoring) selects network traffic for analysis by a network analyzer. The network analyzer can be a Cisco SwitchProbe or other Remote Monitoring (RMON) probes.

This chapter includes the following sections:

SPAN Sources

SPAN sources refer to the interfaces from which traffic can be monitored. The Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter supports Ethernet, port channels, and VLANs as SPAN sources. With VLANs, all supported interfaces in the specified VLAN are included as SPAN sources. You can choose the SPAN traffic in the ingress direction, the egress direction, or both directions for Ethernet interfaces:

- Ingress source (Rx)—Traffic entering the switch through this source port is copied to the SPAN destination port.
- Egress source (Tx)—Traffic exiting the switch through this source port is copied to the SPAN destination port.

Characteristics of Source Ports

A source port, also called a monitored port, is a switched interface that you monitor for network traffic analysis. The switch supports any number of ingress source ports (up to the maximum number of available ports on the switch) and any number of source VLANs.

A source port has these characteristics:

- Can be of any port type: Ethernet, port channel, and VLAN.
- Cannot be monitored in multiple SPAN sessions.
- Cannot be a destination port.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor. For VLAN, port channel sources, the monitored direction can only be ingress and applies to all physical ports in the group. The rx/tx option is not available for VLAN sessions.

- Source ports can be in the same or different VLANs.
- For VLAN SPAN sources, all active ports in the source VLAN are included as source ports.
- The switch supports a maximum of two egress SPAN source ports.

SPAN Destinations

SPAN destinations refer to the interfaces that monitors source ports. The switch supports Ethernet interfaces as SPAN destinations.

Characteristics of Destination Ports

Each local SPAN session must have a destination port (also called a monitoring port) that receives a copy of traffic from the source ports, VLANs. A destination port has these characteristics:

- Can be any physical Ethernet port.
- Cannot be a source port.
- Cannot be the destination port for two different sessions.
- Cannot be a port channel.
- Does not participate in spanning tree while the SPAN session is active.
- Is excluded from the source list and is not monitored if it belongs to a source VLAN of any SPAN session.
- Receives copies of sent and received traffic for all monitored source ports. If a destination port is oversubscribed, it can become congested. This congestion can affect traffic forwarding on one or more of the source ports.

Configuring SPAN

You can configure a SPAN session to duplicate packets from source ports to the specified destination ports on the switch. This section includes the following topics:

Creating and Deleting a SPAN Session

You create a SPAN session by assigning a session number using the monitor command. If the session already exists, any additional configuration is added to that session.

To create a SPAN session, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# monitor session` *session-number*	Enters the monitor configuration mode. New session configuration is added to the existing session configuration.

The following example shows creating a SPAN session:

```
switch# configure terminal
switch(config)# monitor session 2
```

To ensure that you are working with a completely new session, you can delete the desired session number or all SPAN sessions.

To delete SPAN sessions, perform this task:

Command	Purpose
`switch(config)# no monitor session {all | session-number}`	Deletes the configuration of the specified SPAN session or all sessions.

Configuring the Destination Port

The SPAN destination port can only be a physical port on the switch. To configure an Ethernet interface as a SPAN destination port, perform this task:

	Command	Purpose
Step 1	`switch# configure terminal`	Enters configuration mode.
Step 2	`switch(config)# interface ethernet` *slot/port*	Enters interface configuration mode for the specified Ethernet interface selected by the slot and port values.
Step 3	`switch(config-if)# switchport monitor`	Sets the interface to monitor mode. Priority flow control is disabled when the port is configured as a SPAN destination.
Step 4	`switch(config-if)# exit`	Reverts to global configuration mode.
Step 5	`switch(config)# monitor session` *session-number*	Enters the monitor configuration mode.
Step 6	`switch(config-monitor)# destination interface ethernet` *slot/port*	Configures the Ethernet destination port.

The following example shows configuring an Ethernet SPAN destination port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport monitor
switch(config-if)# exit
switch(config)# monitor session 2
switch(config-monitor)# destination interface ethernet 1/3
```

Configuring Source Ports

You can configure the source ports for a SPAN session. The source ports are Ethernet ports.

To configure the source ports for a SPAN session, perform this task:

Command	Purpose
`switch(config-monitor)# source interface` *type slot/port* [**rx** | **tx** | **both**]	Configures sources and the traffic direction in which to duplicate packets. You can enter a range of Ethernet ports. You can specify the traffic direction to duplicate as ingress (rx), egress (tx), or both. By default, the direction is both.

The following example shows configuring an Ethernet SPAN source port:

```
switch# configure terminal
switch(config)# monitor session 2
switch(config-monitor)# source interface ethernet 1/16
```

Configuring Source Port Channels or VLANs

You can configure the source channels for a SPAN session. These ports can be port channels and VLANs. The monitored direction can only be ingress and applies to all physical ports in the group.

To configure the source channels for a SPAN session, perform this task:

Command	Purpose
`switch(config-monitor)# source` {**interface** {**port-channel** | *channel-number* **rx** | **ethernet** *slot/port* | **vlan** *vlan-range*}	Configures port channel or VLAN sources. The monitored direction can only be ingress and applies to all physical ports in the group. For VLAN sources, the monitored direction is implicit.

The following example shows configuring a port channel SPAN source:

```
switch# configure terminal
switch(config)# monitor session 2
switch(config-monitor)# source interface port-channel 1 rx
```

The following example shows configuring a VLAN SPAN source:

```
...
switch(config-monitor)# source vlan 1
```

Configuring the Description of a SPAN Session

To provide a descriptive name of the SPAN session for ease of reference, perform this task:

Command	Purpose
`switch(config-monitor)# description` *description*	Applies a descriptive name to the SPAN session.

The following example shows configuring a description of a SPAN session:

```
switch# configure terminal
switch(config)# monitor session 2
switch(config-monitor)# description monitoring ports ethernet1/2-1/4
```

Suspending or Activating a SPAN Session

The default is to keep the session state shut. To open a session that duplicates packets from sources to destinations, perform this task:

Command	Purpose
`switch(config)# no monitor session {all | session-number} shut`	Opens the specified SPAN session or all sessions.

The following example shows suspending a SPAN session:

```
...
switch(config)# monitor session 3 shut
```

To suspend a SPAN session, perform this task:

Command	Purpose
`switch(config)# monitor session {all | session-number} shut`	Suspends the specified SPAN session or all sessions.

Note The switch supports two active SPAN sessions. When you configure more than two SPAN sessions, the first two sessions are active. During startup, the order of active sessions is reversed; the last two sessions are active. For example, if you configured ten sessions 1 to 10 where 1 and 2 are active, after a reboot, sessions 9 and 10 will be active. To enable deterministic behavior, explicitly suspend the sessions 3 to 10 with the **monitor session** *session-number* **shut** command.

Displaying SPAN Information

To display SPAN information, perform this task:

Command	Purpose
`switch# show monitor [session {all | session-number | range session-range} [brief]]`	Displays the SPAN configuration.

The following example shows how to display SPAN session information:

```
switch# show monitor
SESSION  STATE        REASON                  DESCRIPTION
-------  -----------  ----------------------  --------------------------------
2        up           The session is up
3        down         Session suspended
4        down         No hardware resource
```

The following example shows how to display SPAN session details:

```
switch#
switch(config-monitor)# show monitor session 2
   session 2
---------------
type              : local
state             : down (Session admin shut)
source intf       :
    rx            : Eth1/20
    tx            : Eth1/20
    both          : Eth1/20
source VLANs      :
    rx            :
destination ports :

Legend: f = forwarding enabled, l = learning enabled

    rx            :     rx            :
switch(config-monitor)#
```

CHAPTER 35

Troubleshooting

This chapter describes basic troubleshooting methods used to resolve issues with the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter. This chapter includes the following sections:

Recovering a Lost Password

This section describes how to recover a lost network administrator password using the console port of the switch.

You can recover the network administrator password using one of two methods:

- From the CLI with a username that has network-admin privileges
- By power cycling the switch

This section includes the following topics:

Using the CLI with Network-Admin Privileges

If you are logged in to, or can log into, the switch with a username that has network-admin privileges, perform the following steps:

Step 1 Verify that your username has network-admin privileges:

```
switch# show user-account
user:root
        this user account has no expiry date
        roles:network-operator
user:adminbackup
        this user account has no expiry date
        roles:network-operator
user:admin
        this user account has no expiry date
        roles:network-admin
```

```
user:USERID
        this user account has no expiry date
        roles:network-operator
```

Step 2 Assign a new network administrator password if your username has network-admin privileges:

```
switch# configure terminal
switch(config)# username admin password <new password>
switch(config)# exit
switch#
```

Step 3 Save the configuration:

```
switch# copy running-config startup-config
```

Power Cycling the Switch

If you cannot start a session on the switch that has network-admin privileges, you must recover the network administrator password by power cycling the switch.

Caution This procedure disrupts all traffic on the switch.

Note You cannot recover the administrator password from a Telnet or SSH session. You must have access to the local console connection.

To recover the network administrator password by power cycling the switch, perform the following steps:

Step 1 Establish a terminal session on the console port.

Step 2 Power cycle the switch.

Step 3 Press the **Ctrl-]** key sequence from the console port session when the switch begins the Cisco NX-OS software boot sequence to enter the boot prompt mode:

```
Ctrl-]
switch(boot)#
```

Step 4 Reset the network administrator password:

```
switch(boot)# configure terminal
switch(boot-config)# admin-password <new password>
switch(boot-config)# exit
switch(boot)#
```

Step 5 Display the bootflash: contents to locate the Cisco NX-OS software image file:

```
switch(boot)# dir bootflash:
```

Step 6 Load the Cisco NX-OS system software image.

In the following example, the system image filename is nx-os.bin:

```
switch(boot) # load bootflash:nx-os.bin
```

Step 7 Log in to the switch using the new administrator password:

```
switch login: admin
Password: <new password>
```

Step 8 Reset the new password to ensure that is it is also the SNMP password:

```
switch# configure terminal
switch(config)# username admin password <new password>
switch(config)# exit
switch#
```

Step 9 Save the configuration:

```
switch# copy running-config startup-config
```

Using Ethanalyzer

Ethanalyzer is a Cisco NX-OS protocol analyzer tool based on the Wireshark (formerly Ethereal) open source code. Ethanalyzer is a command-line version of Wireshark that captures and decodes packets. You can use Ethanalyzer to troubleshoot your network and analyze the control-plane traffic.

To configure Ethanalyzer, perform one or more of the following tasks:

Command	Purpose
`switch# ethanalyzer local interface`	Captures packets sent or received and provides detailed protocol information.
`switch# ethanalyzer local interface inband`	Captures packets sent or received and provides detailed protocol information in the inband and outband interfaces.
`switch# ethanalyzer local interface mgmt`	Captures packets sent or received and provides detailed protocol information in the management interfaces.
`switch# ethanalyzer local interface {inband | mgmt | mgmt-backplane} brief`	Captures packets sent or received and provides a summary of protocol information.
`switch# ethanalyzer local interface {inband | mgmt | mgmt-backplane} limit-captured-frames`	Limits the number of frames to capture.
`switch# ethanalyzer local interface {inband | mgmt | mgmt-backplane} limit-frame-size`	Limits the length of the frame to capture.
`switch# ethanalyzer local interface {inband | mgmt | mgmt-backplane} capture-filter`	Filters the types of packets to capture.
`switch# ethanalyzer local interface {inband | mgmt | mgmt-backplane} display-filter`	Filters the types of captured packets to display.
`switch# ethanalyzer local interface {inband | mgmt | mgmt-backplane} decode-internal`	Decodes the internal frame header for Cisco NX-OS. **Note** Do not use this option if you plan to analyze the data using the Wireshark instead of Ethanalyzer.

Command	Purpose
switch# **ethanalyzer local interface** {**inband** | **mgmt** | **mgmt-backplane**} **write**	Saves the captured data to a file.
switch# **ethanalyzer local read**	Opens the captured data file and analyzes it.

Ethanalyzer does not capture data traffic that Cisco NX-OS forwards in the hardware.

Ethanalyzer uses the same capture filter syntax as tcpdump. For more information, see the following URL:

http://www.tcpdump.org/tcpdump_man.html

For information on the syntax of the display filter, see the following URL:

http://wiki.wireshark.org/DisplayFilters

The following example shows captured data (limited to four packets) on the management interface:

```
switch# ethanalyzer local interface mgmt brief limit-captured-frames 4
Capturing on eth2
2009-05-19 11:07:06.633801 00:05:ad:00:33:37 -> ff:ff:ff:ff:ff:ff ARP Who has
172.29.231.1?  Tell 172.29.231.177
2009-05-19 11:07:06.813956 172.29.230.3 -> 224.0.0.2    HSRP Hello (state Standby)
2009-05-19 11:07:06.829894 172.29.230.3 -> 224.0.0.2    HSRP Hello (state Standby)
2009-05-19 11:07:06.980957 172.29.230.2 -> 224.0.0.5    OSPF Hello Packet
4 packets captured
```

The following example shows captured data (limited to 2 packets) on the inband interface:

```
switch# ethanalyzer local interface inband brief limit-captured-frames 2
Capturing on inb0
2009-05-19 11:08:42.911357 00:05:ad:00:34:73 -> 01:80:c2:00:00:00 STP RST. Root =
32769/00:05:ad:00:34:71  Cost = 0  Port = 0x8093
2009-05-19 11:08:42.911390 00:05:ad:00:34:73 -> 01:80:c2:00:00:00 STP RST. Root =
32769/00:05:ad:00:34:71  Cost = 0  Port = 0x8093
2 packets captured
```

The following example shows detailed captured data for one HSRP packet:

```
switch(config)# ethanalyzer local interface mgmt capture-filter "tcp port 23"
limit-captured-frames 1
Capturing on eth2
Frame 1 (74 bytes on wire, 74 bytes captured)
    Arrival Time: May 19, 2009 11:07:52.061847000
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 74 bytes
    Capture Length: 74 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:ip:tcp]
Ethernet II, Src: 00:1a:30:00:bc:00 (00:1a:30:00:bc:00), Dst: 00:05:ad:00:34:5a
(00:05:ad:00:34:5a)
    Destination: 00:05:ad:00:34:5a (00:05:ad:00:34:5a)
        Address: 00:05:ad:00:34:5a (00:05:ad:00:34:5a)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Source: 00:1a:30:00:bc:00 (00:1a:30:00:bc:00)
        Address: 00:1a:30:00:bc:00 (00:1a:30:00:bc:00)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
```

```
    Type: IP (0x0800)
Internet Protocol, Src: 171.69.27.169 (171.69.27.169), Dst: 172.29.231.226
(172.29.231.226)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 60
    Identification: 0x6c57 (27735)
    Flags: 0x04 (Don't Fragment)
        0... = Reserved bit: Not set
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 56
    Protocol: TCP (0x06)
    Header checksum: 0x7b76 [correct]
        [Good: True]
        [Bad : False]
    Source: 171.69.27.169 (171.69.27.169)
    Destination: 172.29.231.226 (172.29.231.226)
Transmission Control Protocol, Src Port: 51225 (51225), Dst Port: telnet (23), Seq: 0,
Len: 0
    Source port: 51225 (51225)
    Destination port: telnet (23)
    Sequence number: 0    (relative sequence number)
    Header length: 40 bytes
    Flags: 0x02 (SYN)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...0 .... = Acknowledgment: Not set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..1. = Syn: Set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0xbe6e [correct]
        [Good Checksum: True]
        [Bad Checksum: False]
    Options: (20 bytes)
        Maximum segment size: 1460 bytes
        SACK permitted
        Timestamps: TSval 3876668892, TSecr 0
        NOP
        Window scale: 4 (multiply by 16)

1 packet captured
```

For more information on Wireshark, see the following URL: http://www.wireshark.org/docs/

show tech-support Command

This section describes the **show tech-support** commands and includes the following topics:

The **show tech-support** command is useful when collecting a large amount of information about the switch for troubleshooting purposes. The output of this command can be provided to Cisco TAC representatives when reporting a problem.

The **show tech-support** command displays the output of several **show** commands at once. The output from this command varies depending on your configuration. Use the **show tech-support** command in EXEC mode to display general information about the switch when reporting a problem.

You can choose to have detailed information for each command. You can specify the output for a particular interface, module, or VSAN. Each command output is separated by line and the command precedes the output.

Note Explicitly set the **terminal length** command to 0 (zero) to disable auto-scrolling and enable manual scrolling. Use the **show terminal** command to view the configured the terminal size. After obtaining the output of this command, remember to reset your terminal length as required.

Tip You can save the output of this command to a file by appending **>** (left arrow) and the filename to the **show tech-support** command. If you save this file, verify you have sufficient space to do so—each of these files may take about 1.8 MB. However, you can zip this file using the **gzip** *filename* command. Copy the zipped file to the required location using the **copy** command and unzip the file using the **gunzip** command.

The default output of the **show tech-support** command includes the output of the following commands:

- **show switchname**
- **show system uptime**
- **show interface mgmt0**
- **show interface mgmt1**
- **show system resources**
- **show version**
- **dir bootflash:**
- **show inventory**
- **show diagnostic result all**
- **show logging log**
- **show module**
- **show environment**
- **show sprom backplane**
- **show clock**
- **show callhome**
- **show snmp**
- **show interface brief**
- **show interface**
- **show running-config**
- **show startup-config**

- show ip route
- show arp
- show monitor session all
- show accounting log
- show process
- show process cpu
- show process log
- show process memory
- show processes log details
- show logging log
- show license host-id
- show license
- show license usage
- show system reset-reason
- show logging nvram
- show install all status
- show install all failure-reason
- show system internal log install
- show system internal log install details
- show cores
- show topology
- show kernel internal aipc
- show tech-support acl
- show vlan
- show vlan access-map
- show mac-address-table
- show spanning-tree summary
- show spanning-tree active
- show interface trunk
- show aclmgr status
- show aclmgr internal dictionaries
- show aclmgr internal log
- show aclmgr internal ppf
- show aclmgr internal state-cache
- show access-lists
- show platform software ethpm internal info all
- show logging onboard obfl-logs

show tech-support brief Command

Use the **show tech-support brief** command to obtain a quick, condensed review of the switch configurations. This command provides a summary of the current running state of the switch (see the following example).

The **show tech-support brief** command is useful when collecting information about the switch for troubleshooting purposes. The output of this command can be provided to technical support representatives when reporting a problem.

Tip You can save the output of this command to a file by appending **>** (left arrow) and the filename to the **show tech-support brief** command.

The following example shows how to display a condensed view of the switch configurations:

```
switch# show tech-support brief
Switch Name           : switch
Switch Type           : DS-C9134-K9-SUP
Kickstart Image       : 4.1(2)E1(1) bootflash:///n4000_kickstart.4.1.2.E1.0.175.gbin
System Image          : 4.1(2)E1(1) bootflash:///n4000_system.4.1.2.E1.0.189.bin
IP Address/Mask       : 209.165.200.225/254
Switch WWN            : parsing


--------------------------------------------------------------------------------
Ethernet      VLAN   Type Mode   Status  Reason                   Speed     Port
Interface                                                                   Ch #
--------------------------------------------------------------------------------
Eth1/1        1      eth  access up      none                       10G(D) --
Eth1/2        1      eth  access up      none                       10G(D) --
Eth1/3        1      eth  trunk  up      none                       10G(D) --
Eth1/4        1      eth  access up      none                       10G(D) --
Eth1/5        1      eth  access up      none                       10G(D) --
Eth1/6        1      eth  access up      none                       10G(D) --
Eth1/7        1      eth  access up      none                       10G(D) --
Eth1/8        1      eth  access up      none                       10G(D) --
Eth1/9        1      eth  access up      none                       10G(D) --
Eth1/10       1      eth  access up      none                       10G(D) --
Eth1/11       1      eth  access up      none                       10G(D) --
Eth1/12       1      eth  access up      none                       10G(D) --
Eth1/13       1      eth  access up      none                       10G(D) --
Eth1/14       1      eth  access up      none                       10G(D) --
Eth1/15       1      eth  access down    SFP not inserted           10G(D) --
Eth1/16       1      eth  access down    SFP not inserted           10G(D) --
Eth1/17       1      eth  access down    SFP not inserted           10G(D) --
Eth1/18       1      eth  access down    SFP not inserted           10G(D) --
Eth1/19       1      eth  access down    SFP not inserted           10G(D) --
Eth1/20       monitr eth  access down    SFP not inserted           10G(D) --

--------------------------------------------------------------------------------
Port   VRF          Status IP Address                              Speed    MTU
--------------------------------------------------------------------------------
mgmt0  --           up     209.165.200.225                          1000     1500
mgmt1  --           up     --                                       100      1500
switch#
```

show tech-support platform Command

Use the **show tech-support platform** command to obtain information about the platform configuration of your switch.

The output of the **show tech-support platform** command includes the output of the following commands:

- **show platform fwm mem-stats detail**
- **show platform fwm info global**
- **show platform fwm info pif all verbose**
- **show platform fwm info lif all verbose**
- **show platform fwm info error stats**
- **show platform fwm info error history**
- **show platform fwm info stm-stats**
- **show platform fwm info pc all verbose**
- **show platform fwm info ppf**
- **show platform fwm info pss all**
- **show platform fwm info pif all**
- **show platform fwm info lif all**
- **show platform fwm info global**
- **show hardware internal cpu-mac mgmt counters**
- **show hardware internal cpu-mac mgmt stats**
- **show hardware internal cpu-mac inband counters**
- **show platform software pfm internal errors**
- **show platform software pfm internal msgs**
- **show platform software pfm internal info**
- **show environment**
- **show sprom all**
- **show module**
- **show hardware internal pci**
- **show system health internal errors**
- **show system health internal messages**
- **show system health internal plog**
- **show chassis summary**

show tech-support platform callhome Command

Use the **show tech-support platform callhome** command to obtain information about the callhome platform configuration of your switch.

The output of the **show tech-support platform callhome** command includes the output of the following commands:

- **show hardware internal cpu-mac inband counters**
- **show hardware internal cpu-mac mgmt counters**
- **show hardware internal cpu-mac mgmt stats**
- **show hardware internal xcvr event-history errors**
- **show hardware internal xcvr event-history msgs**
- **show platform software pfm internal errors**
- **show platform software pfm internal msgs**
- **show platform software pfm internal info**
- **show system health internal errors**
- **show system health internal messages**
- **show system health internal plog**
- **show environment**
- **show sprom all**
- **show module**
- **show hardware internal pci**

CHAPTER 36

Configuration Limits

The features supported by the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter Switch have maximum configuration limits. For some of the features, we have verified configurations that support limits less than the maximum. Table 36-1 lists the Cisco verified limits and maximum limits for switches running Cisco NX-OS for the Cisco Nexus 4001I and 4005I Switch Module for IBM BladeCenter.

Table 36-1 Configuration Limits

Feature	Verified Limit	Maximum Limit
VLANs per switch[1]	512	512
PVLANs per switch	100	100
Ethernet MTU	9,216 bytes	9,216 bytes (ASIC limit)
STP logical interface instances	3,000 instances Only 2,500 can be true STP bridge to STP bridge connections	N/A
MST instances per switch (every instance is RSTP enabled)	64	64 (IEEE standard)
Station Table[2]	16,000 entries	32,000 entries
IP Multicast addresses (IGMP snooping)	1,000 addresses	1,000 addresses
Device Aliases per fabric	8,000	8,000
Event Traps - forward via Email	4 destinations	50 destinations
FLOGIs or FDISCs per NPV port group	62	100
QoS System Classes	5 user-configurable classes	5 user-configurable classes
VLAN ACL (VACL) entries for the whole switch	1,024	1,024
Port ACL (PACL) entries per physical Ethernet interface	512	512
EtherChannels	12 EtherChannels	16
SPAN Sessions	2 active sessions	18 sessions configured (2 active)
Egress SPAN sources	2	2

1. The entire 4094 VLAN ID space is supported.
2. Station table contains 8000 unicast plus 1000 multicast addresses.

INDEX

A

B

C

D

E

F

M

N

O

P

Q

R

S

T

U

V

W

X