Catalyst 6500 Series Switch SIP, SSC, and SPA Software Configuration Guide

July 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-8655-04

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

Catalyst 6500 Series Switch SIP, SSC, and SPA Software Configuration Guide

CONTENTS

CHAPTER 2

PART 2

CHAPTER 3

Preface

This preface describes the objectives and organization of this document and explains how to find additional information on related products and services. This preface contains the following sections:

Objectives

This document describes the configuration and troubleshooting of shared port adapters (SPAs) and SPA interface processors (SIPs) that are supported on a Catalyst 6500 Series switch.

Audience

This publication is for experienced network administrators who configure and maintain VPN systems and the Catalyst 6500 Series switch.

Document Revision History

Table 1 records technical changes to this document. The table shows the Cisco IOS software release number and document revision number for the change, the date of the change, and a brief summary of the change.

Table 1 Document Revision History

Release No.	Revision	Date	Change Summary
12.2(33)SXI2	OL-8655-05	July 9, 2009	• Support was added for the following SPAs on the Cisco 7600 SIP-400: – 1-Port 10 Gigabit Ethernet SPA, Version 2 (SPA-1X10GE-L-V2)
12.2(33)SXI	OL-8655-04	October 31, 2008	The following modifications were made: • Support was restored for the Cisco 7600 SIP-600. • Support was restored for the ATM SPAs. • Support was introduced for the following feature on the Cisco 7600 SIP-200: – Asymmetric Carrier Delay • Support was added for the following SPAs on the Cisco 7600 SIP-400: – 2-Port and 4-Port Clear Channel T3/E3 SPA – 2-Port and 4-Port Channelized T3 SPA – 8-Port Channelized T1/E1 SPA – 5-Port Gigabit Ethernet SPA (V2) • Support was introduced for the following features on the Cisco 7600 SIP-400: – Asymmetric Carrier Delay – Any Transport over MPLS over GRE (AToMoGRE) • Support was added for the following SPAs on the Cisco 7600 SIP-600: – 2-Port OC-48c/STM-16 POS SPA • New features were introduced for the IPsec VPN SPA

Table 1 Document Revision History (continued)

12.2(33)SXH	OL-8655-03	August 20, 2007	The following modifications were made: • Support was removed for the Cisco 7600 SIP-600. • Support was removed for the ATM SPAs. • Support was added for the following SPAs on the Cisco 7600 SIP-200: – 1-Port Channelized OC-3/STM-1 SPA – 4-Port and 8-Port Fast Ethernet SPA • Support for the following features were introduced on the Cisco 7600 SIP-200: – BCP over dMLPPP (Trunk Mode)—Channelized SPAs – MPLS over RBE—ATM SPAs – Multi-VC to VLAN scalability – QoS Support on Bridging Features • Support was added for the following SPA on the Cisco 7600 SIP-400: – 2-Port Channelized T3 SPA • Support for the following features were introduced on the Cisco 7600 SIP-400: – Ethernet over MPLS (EoMPLS) VC Scaling—Increase from 4K to 10K VCs – Ingress/Egress COS Classification with Ingress Policing per VLAN or EoMPLS VC – Hierarchical VPLS (H-VPLS) with MPLS Edge – VPLS Multiple VCs per Spoke – Hierarchical QoS Support for EoMPLS VCs – QoS Support on Bridging Features – Lawful Intercept

Table 1 Document Revision History (continued)

12.2(33)SXH	OL-8655-03	August 20, 2007	• The following features were introduced for the IPsec VPN SPA: – IPsec Anti-replay Window size – IPsec Preferred Peer – Persistent Self-signed Certificates – Easy VPN Remote RSA Signature Storage • The following feature was removed for the IPsec VPN SPA: – IPsec stateful failover using HSRP and SSP • The single configuration chapter for the IPsec VPN SPA has been restructured into several smaller chapters.
12.2(18)SXF10	OL-5070-05 OL-8655-02	July 13, 2007	Support was introduced for the 1-Port OC-48c/STM-16 POS SPA on the Cisco 7600 SIP-400.
12.2(18)SXF2	OL-5070-04 OL-8655-01	April 25, 2006	Modified references to cRTP to include support for the 2-Port and 4-Port Clear Channel T3/E3 SPA.

Table 1 Document Revision History (continued)

12.2(18)SXF2	OL-5070-04 OL-8655-01	February 28, 2006	The following updates were made to the documentation: • Removed the restriction of "Mapping DSCP values to MPLS EXP bits is not supported" from the Cisco 7600 SIP-600 list of restrictions. • Added the following VPLS scalability support information for the Cisco 7600 SIP-600: – Up to 4000 VPLS domains – Up to 60 VPLS peers per domain – Up to 30,000 Pseudo Wires, used in any combination of domains and peers up to the 4000-domain or 60-peer maximums. For example, support of up to 4000 domains with 7 peers or up to 60 peers in 500 domains. • Added H-VPLS with QinQ edge feature support on Cisco 7600 SIP-600—Requires Cisco 7600 SIP-600 in the uplink, and any LAN port or Cisco 7600 SIP-600 on the downlink. • Removed VPLS pseudo-wire redundancy feature support for the Cisco 7600 SIP-600. • Removed the "Cisco 7600 SIP-600 MPLS Marking" section and bullet. • Modified the encapsulations supported in the ATM chapters to "aal5snap" only. • Corrected the note in the "Configuring Compressed Real-Time Protocol" section of Chapter 4, "Configuring the SIPs and SSC" to state: "cRTP is supported only on the Cisco 7600 SIP-200 with the 8-Port Channelized T1/E1 SPA and *2-Port and 4-Port Channelized T3 SPA*."
12.2(18)SXF2	OL-5070-04 OL-8655-01	January 27, 2006	The following update to the hardware-based MLPPP LFI guidelines was made in Chapter 15, "Configuring the 8-Port Channelized T1/E1 SPA," and Chapter 17, "Configuring the 2-Port and 4-Port Channelized T3 SPAs": • When hardware-based LFI is enabled, fragmentation counters are not displayed.

Table 1 Document Revision History (continued)

12.2(18)SXF2	OL-5070-04 OL-8655-01	January 20, 2006	Fourth release. The following modifications were made: • The 1-Port OC-192c/STM-64 POS/RPR VSR Optics SPA was introduced on the Cisco 7600 SIP-600. • Support was introduced for the configuration of IP multicast over a GRE tunnel on the IPsec VPN SPA. • Support for the “Enhancements to RFC 1483 Spanning Tree Interoperability” feature was added for ATM SPAs on the Cisco 7600 SIP-200. • Documentation of a workaround for ATM SPA configuration on the Cisco 7600 SIP-200 has been added in Chapter 7, “Configuring the ATM SPAs” to address a Routed Bridge Encapsulation (RBE) limitation where only one remote MAC address is supported.

Table 1 Document Revision History (continued)

12.2(18)SXF	OL-5070-03	January 12, 2006	The following modifications were made: • Adjusted ATM SPA PVC restriction (correctly noted elsewhere in the documentation) from "A maximum number of 400 PVCs or SVCs. . ." to "A maximum number *of 1000 PVCs* or 400 SVCs configured with MQC policy maps." • Added cross-references throughout the "Overview of the SIPs and SSC" chapter to the Cisco IOS Release SX Supervisor Engine release notes. • Updated the Cisco 7600 SIP-400 restrictions to clarify that the SIP does not work with the Supervisor Engine PFC3A *or in PFC3A mode*. • Updated the Cisco 7600 SIP-600 restrictions to clarify lack of support for the Supervisor Engine 720 PFC3A or PFC3A mode: "The Cisco 7600 SIP-600 is not supported by the Supervisor Engine 32. The Cisco 7600 SIP-600 is supported by the Supervisor Engine 720 PFC3B and Supervisor Engine 720 PFC3BXL. *It is not supported with a Supervisor Engine 720 PFC3A or in PFC3A mode.*" • Added a cross-reference to the "Overview of the SIPs and SSC" chapter in each of the SPA overview chapters to ease location of additional features/restrictions that are SIP- or SSC-specific. • Removed the list of supported modules from the "Overview of the IPsec VPN SPA" chapter. Any unsupported modules will be documented in the restrictions section.

Table 1 Document Revision History (continued)

Release No.	Revision	Date	Change Summary
12.2(18)SXF	OL-5070-03	January 12, 2006	• Further qualified Cisco 7600 SIP-200 Any Transport over MPLS (AToM) support for ATM in the “Overview of the SIPs and SSC” chapter to state: “Any Transport over MPLS (AToM) support, including: – ATM over MPLS (ATMoMPLS)—AAL5 *VC* mode – Ethernet over MPLS (EoMPLS)—*(Single cell relay) VC mode*” • Removed references to “*1-Port 10-Gigabit Ethernet SPA and 10-Port Gigabit Ethernet SPA on a SIP-400*” in the “Enabling Autonegotiation” and “Disabling Autonegotiation” sections of the “Configuring Gigabit Ethernet SPAs” chapter. • Qualified AToM core-facing restriction for the Cisco 7600 SIP-200 as follows: – AToM (ATMoMPLS, FRoMPLS, HDLCoMPLS, and PPPoMPLs) on a SPA requires a Cisco 7600 SIP-200, FlexWAN, Enhanced FlexWAN, or OSM PXF interface as the core-facing interface. – AToM (ATMoMPLS, FRoMPLS) on SIP-200 also are supported with a Cisco 7600 SIP-400 as the core-facing interface. • Documentation of the Fast Software Upgrade (FSU) procedure supported by Route Processor Redundancy (RPR) for supervisor engines was added to Chapter 31, “Upgrading Field-Programmable Devices.”

Table 1 Document Revision History (continued)

Release No.	Revision	Date	Change Summary
12.2(18)SXF	OL-5070-03	September 19, 2005	Third release. The following hardware was introduced: • 1-Port OC-48c/STM-16 ATM SPA • 2-Port Gigabit Ethernet SPA • 5-Port Gigabit Ethernet SPA • 10-Port Gigabit Ethernet SPA • 1-Port 10-Gigabit Ethernet SPA • 1-Port OC-192c/STM-64 POS/RPR SPA • 1-Port OC-192c/STM-64 POS/RPR XFP SPA For specific feature changes, see the Feature History tables in the “Overview” chapters of this book.
12.2(18)SXE2	OL-5070-02	August 17, 2005	• The “Configuring the 8-Port Channelized T1/E1 SPA” and “Configuring the 2-Port and 4-Port Channelized T3 SPAs” were modified to clarify support of MLPPP and MLFR for both E1 and T1 links. • Added cRTP to the supported features list for the serial SPAs in the “Overview of the Serial SPAs” chapter. • Document was modified with the following updates in the “Configuring the SIPs and SSC” chapter: – Removed references to support of software-based MLFR. – In the “Assigning an Interface to an MLPPP bundle,” moved step order of the **ppp multilink** command and qualified it as optional. – Under “MLPPP Configuration Guidelines,” added guidelines for distributed links on the Cisco 7600 SIP-200 and restrictions. – Under “MLPPP Configuration Tasks” and “MLFR Configuration Tasks, added task to emphasize that distributed CEF is required for these features; however, dCEF is automatically enabled on the Catalyst 6500 Series switch.

Table 1 Document Revision History (continued)

12.2(18)SXE2	OL-5070-02	July 25, 2005	Second release. The Cisco 7600 SSC-400 and IPsec VPN SPA are introduced.
12.2(18)SXE	OL-5070-01	March 28, 2005	First release.

Organization

This document contains the following chapters:

Chapter	Title	Description
Chapter 1	Using Cisco IOS Software	Provides an introduction to accessing the command-line interface (CLI) and using the Cisco IOS software and related tools.
Chapter 2	SIP, SSC, and SPA Product Overview	Provides a brief introduction to the SIP and SPA products on the Catalyst 6500 Series switch, and information about SIP, SSC, SPA, and optics compatibility.
Chapter 3	Overview of the SIPs and SSC	Describes release history, and feature and Management Information Base (MIB) support for the SIPs and SSCs on the Catalyst 6500 Series switch.
Chapter 4	Configuring the SIPs and SSC	Describes related configuration and verification information for the SIPs and SSCs on the Catalyst 6500 Series switch.
Chapter 5	Troubleshooting the SIPs and SSC	Describes techniques that you can use to troubleshoot the operation of the SIPs and SSCs on the Catalyst 6500 Series switch.
Chapter 6	Overview of the ATM SPAs	Describes release history, feature and Management Information Base (MIB) support, and an introduction to the ATM SPA architecture on the Catalyst 6500 Series switch.
Chapter 7	Configuring the ATM SPAs	Describes the configuration and verification information for the ATM SPAs on the Catalyst 6500 Series switch.
Chapter 8	Troubleshooting the ATM SPAs	Describes techniques that you can use to troubleshoot the operation of the ATM SPAs on the Catalyst 6500 Series switch.
Chapter 9	Overview of the Fast Ethernet and Gigabit Ethernet SPAs	Describes release history, feature and Management Information Base (MIB) support, and an introduction to the Gigabit Ethernet SPA architecture on the Catalyst 6500 Series switch.
Chapter 10	Configuring the Fast Ethernet and Gigabit Ethernet SPAs	Describes the configuration and verification information for the Gigabit Ethernet SPAs on the Catalyst 6500 Series switch.

Chapter	Title	Description
Chapter 11	Troubleshooting the Fast Ethernet and Gigabit Ethernet SPAs	Describes techniques that you can use to troubleshoot the operation of the Gigabit Ethernet SPAs on the Catalyst 6500 Series switch.
Chapter 12	Overview of the POS SPAs	Describes release history, feature and Management Information Base (MIB) support, and an introduction to the POS SPA architecture on the Catalyst 6500 Series switch.
Chapter 13	Configuring the POS SPAs	Describes the configuration and verification information for the POS SPAs on the Catalyst 6500 Series switch.
Chapter 14	Overview of the Serial SPAs	Describes release history, feature and Management Information Base (MIB) support, and an introduction to the serial SPA architecture on the Catalyst 6500 Series switch.
Chapter 15	Configuring the 8-Port Channelized T1/E1 SPA	Describes the configuration and verification information for the 8-Port Channelized T1/E1 SPAs on the Catalyst 6500 Series switch.
Chapter 16	Configuring the 2-Port and 4-Port Clear Channel T3/E3 SPAs	Describes the configuration and verification information for the 2-Port and 4-Port Clear Channel T3/E3 SPAs on the Catalyst 6500 Series switch.
Chapter 17	Configuring the 2-Port and 4-Port Channelized T3 SPAs	Describes the configuration and verification information for the 2-Port and 4-Port Channelized T3 SPAs on the Catalyst 6500 Series switch.
Chapter 18	Configuring the 1-Port Channelized OC-3/STM-1 SPA	Describes the configuration and verification information for the 1-Port Channelized OC-3/STM-1 SPA on the Catalyst 6500 Series switch.
Chapter 19	Troubleshooting the Serial SPAs	Describes techniques that you can use to troubleshoot the operation of the serial SPAs on the Catalyst 6500 Series switch.
Chapter 20	Overview of the IPsec VPN SPA	Describes release history, feature and Management Information Base (MIB) support, and an introduction to the IPsec VPN SPA architecture on the Catalyst 6500 Series switch.
Chapter 21	Configuring VPNs in Crypto-Connect Mode	Describes the configuration and verification information for IPsec VPNs using Crypto-Connect Mode on the Catalyst 6500 Series switch.
Chapter 22	Configuring VPNs in VRF Mode	Describes the configuration and verification information for IPsec VPNs using VRF Mode on the Catalyst 6500 Series switch.
Chapter 23	Configuring IPsec VPN Fragmentation and MTU	Describes the configuration and verification information for IPsec Fragmentation and MTU on the Catalyst 6500 Series switch.
Chapter 24	Configuring IKE Features Using the IPsec VPN SPA	Describes the configuration and verification information for Internet Key Exchange (IKE) features using the IPsec VPN SPA on the Catalyst 6500 Series switch.

Chapter	Title	Description
Chapter 25	Configuring Enhanced IPsec Features Using the IPsec VPN SPA	Describes the configuration and verification information for enhanced IPsec features using the IPsec VPN SPA on the Catalyst 6500 Series switch.
Chapter 26	Configuring PKI Using the IPsec VPN SPA	Describes the configuration and verification information for Public Key Infrastructure (PKI) features using the IPsec VPN SPA on the Catalyst 6500 Series switch.
Chapter 27	Configuring Advanced VPNs Using the IPsec VPN SPA	Describes the configuration and verification information for advanced IPsec VPNs using the IPsec VPN SPA on the Catalyst 6500 Series switch.
Chapter 28	Configuring Duplicate Hardware and IPsec Failover Using the IPsec VPN SPA	Describes the configuration and verification information for duplicate hardware configurations and IPsec failover using the IPsec VPN SPA on the Catalyst 6500 Series switch.
Chapter 29	Configuring Monitoring and Accounting for the IPsec VPN SPA	Describes the configuration and verification information for the IPsec VPN SPA on the Catalyst 6500 Series switch.
Chapter 30	Troubleshooting the IPsec VPN SPA	Describes techniques that you can use to troubleshoot the operation of the IPsec VPN SPA on the Catalyst 6500 Series switch.
Chapter 31	Upgrading Field-Programmable Devices	Provides information about upgrading the field-programmable devices on the Catalyst 6500 Series switch.

Document Conventions

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.

Tip Means *the following information will help you solve a problem*.

Command descriptions use these conventions:

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

Screen examples use these conventions:

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

Related Documentation

This section refers you to other documentation that also might be useful as you configure your Catalyst 6500 Series switch. The documentation listed in this section is available online.

Catalyst 6500 Series Switch Documentation

As you configure SIPs and SPAs on your Catalyst 6500 Series switch, you should also refer to the following companion publication for important hardware installation information:

- *Catalyst 6500 Series Switch SIP and SPA Hardware Installation Guide*

Some of the following other Catalyst 6500 Series switch publications might be useful to you as you configure your Catalyst 6500 Series switch:

- *Cisco IOS Software Configuration Guide, Release 12.2(33)SXH and Later Releases*

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/book.html

- *Cisco IOS Master Command List, Release 12.2SX*

  http://www.cisco.com/en/US/docs/ios/mcl/122sxmcl/12_2sx_mcl_book.html

- *Cisco IOS Release 12.2SX System Message Guide*

  http://www.cisco.com/en/US/docs/ios/12_2sx/system/messages/122sxsms.html

- *Cisco 7600 Series Internet Router MIB Specifications Guide*

  http://www.cisco.com/en/US/docs/routers/7600/technical_references/7600_mib_guides/MIB_Guide_ver_6/mibgde6.html

Several other publications are also related to the Catalyst 6500 Series switch. For a complete reference of related documentation, refer to the *Cisco Catalyst 6500 Series Switch Support Documentation* located at the following URL:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Other Cisco IOS Software Publications

Your switch and the Cisco IOS software running on it contain extensive features. You can find documentation for Cisco IOS software features at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/tsd_products_support_category_home.html

Other Cisco IOS Release 12.2SX Software Publications

Documentation for Cisco IOS Release 12.2SX, including command reference and system error messages, can be found at the following URL:

http://www.cisco.com/en/US/products/ps6017/tsd_products_support_series_home.html

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.

PART 1

Introduction

CHAPTER 1

Using Cisco IOS Software

This chapter provides information to prepare you to configure a SPA interface processor (SIP) or shared port adapter (SPA) using the Cisco IOS software. It includes the following sections:

- Accessing the CLI Using a Switch Console, page 1-1
- Using Keyboard Shortcuts, page 1-6
- Using the History Buffer to Recall Commands, page 1-6
- Understanding Command Modes, page 1-6
- Getting Help, page 1-8
- Using the no and default Forms of Commands, page 1-11
- Saving Configuration Changes, page 1-12
- Filtering Output from the show and more Commands, page 1-12
- Finding Support Information for Platforms and Cisco Software Images, page 1-13

Accessing the CLI Using a Switch Console

The following sections describe how to access the command-line interface (CLI) using a directly-connected console or by using Telnet or a modem to obtain a remote console:

- Accessing the CLI Using a Directly-Connected Console, page 1-1
- Accessing the CLI from a Remote Console Using Telnet, page 1-3
- Accessing the CLI from a Remote Console Using a Modem, page 1-5

For more detailed information about configuring and accessing a switch through various services, refer to the *Cisco IOS Terminal Services Configuration Guide* and *Cisco IOS Terminal Services Command Reference* publications.

For more information about making the console cable connections, refer to the *Catalyst 6500 Series Switch Module Installation Guide*.

Accessing the CLI Using a Directly-Connected Console

This section describes how to connect to the console port on the switch and use the console interface to access the CLI.

The console port on a Catalyst 6500 Series switch is an EIA/TIA-232 asynchronous, serial connection with hardware flow control and an RJ-45 connector. The console port is located on the front panel of the supervisor engine, as shown in Figure 1-1 and Figure 1-2.

Figure 1-1 Supervisor Engine 720 Console Port Connector

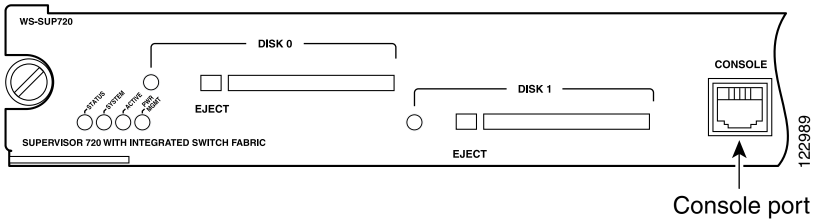
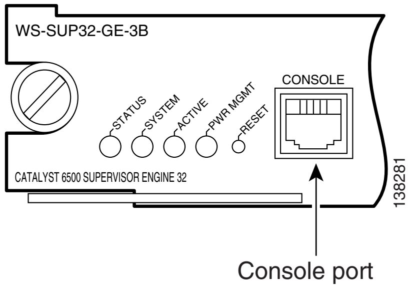


Figure 1-2 Supervisor Engine 32 Console Port Connector



Connecting to the Console Port

Before you can use the console interface on the switch using a terminal or PC, you must perform the following steps:

Step 1 Configure your terminal emulation software with the following settings:

- 9600 bits per second (bps)
- 8 data bits
- No parity
- 2 stop bits

Note These are the default serial communication parameters on the switch. For information about how to change the default settings to meet the requirements of your terminal or host, refer to the *Cisco IOS Terminal Services Configuration Guide.*

Step 2 Connect a terminal or PC to the console port using *one* of the following methods:

a. To connect to the console port using the cable and adapters provided in the accessory kit that shipped with your Catalyst 6500 Series switch:

– Place the console port mode switch in the in position (factory default).

- Connect to the port using the RJ-45-to-RJ-45 cable and RJ-45-to-DB-25 DTE adapter or using the RJ-45-to-DB-9 DTE adapter (labeled "Terminal").

b. To connect to the console port using a Catalyst 5000 family Supervisor Engine III console cable:

- Place the console port mode switch in the out position.
- Connect to the port using the Supervisor Engine III cable and the appropriate adapter for the terminal connection.

Using the Console Interface

To access the CLI using the console interface, complete the following steps:

Step 1 After you attach the terminal hardware to the console port on the switch and you configure your terminal emulation software with the proper settings, the following prompt appears:

```
Press Return for Console prompt
```

Step 2 Press **Return** to enter user EXEC configuration mode. The following prompt appears:

```
Router>
```

Step 3 From user EXEC configuration mode, enter the **enable** command as shown in the following example:

```
Router> enable
```

Step 4 At the password prompt, enter your system's password. (The following example shows entry of the password called "enablepass"):

```
Password: enablepass
```

Step 5 When your enable password is accepted, the privileged EXEC configuration mode prompt appears:

```
Router#
```

Step 6 You now have access to the CLI in privileged EXEC configuration mode and you can enter the necessary commands to complete your desired tasks.

Step 7 To exit the console session, enter the **quit** command as shown in the following example:

```
Router# quit
```

Accessing the CLI from a Remote Console Using Telnet

This section describes how to connect to the console interface on a switch using Telnet to access the CLI.

Preparing to Connect to the Switch Console Using Telnet

Before you can access the switch remotely using Telnet from a TCP/IP network, you need to configure the switch to support virtual terminal lines (vtys) using the **line vty** global configuration command. You also should configure the vty lines to require login and specify a password.

Note To prevent disabling login on the line, be careful that you specify a password with the **password** command when you configure the **login** line configuration command. If you are using authentication, authorization, and accounting (AAA), you should configure the **login authentication** line configuration command. To prevent disabling login on the line for AAA authentication when you configure a list with the **login authentication** command, you must also configure that list using the **aaa authentication login** global configuration command. For more information about AAA services, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference* publications.

In addition, before you can make a Telnet connection to the switch, you must have a valid host name for the switch or have an IP address configured on the switch. For more information about requirements for connecting to the switch using Telnet, information about customizing your Telnet services, and using Telnet key sequences, refer to the *Cisco IOS Terminal Services Configuration Guide*.

Using Telnet to Access a Console Interface

To access a console interface using Telnet, complete the following steps:

Step 1 From your terminal or PC, enter one of the following commands:

- **connect** *host* [*port*] [*keyword*]
- **telnet** *host* [*port*] [*keyword*]

In this syntax, *host* is the switch host name or an IP address, *port* is a decimal port number (23 is the default), and *keyword* is a supported keyword. For more information, refer to the *Cisco IOS Terminal Services Command Reference*.

Note If you are using an access server, then you will need to specify a valid port number such as **telnet 172.20.52.40 2004**, in addition to the host name or IP address.

The following example shows the **telnet** command to connect to the switch named router:

```
unix_host% telnet router
Trying 172.20.52.40...
Connected to 172.20.52.40.
Escape character is '^]'.
unix_host% connect
```

Step 2 At the password prompt, enter your login password. The following example shows entry of the password called “mypass”:

```
User Access Verification

Password: mypass
```

Note If no password has been configured, press **Return**.

Step 3 From user EXEC configuration mode, enter the **enable** command as shown in the following example:

```
Router> enable
```

Step 4 At the password prompt, enter your system's password. (The following example shows entry of the password called "enablepass"):

```
Password: enablepass
```

Step 5 When the enable password is accepted, the privileged EXEC configuration mode prompt appears:

```
Router#
```

Step 6 You now have access to the CLI in privileged EXEC configuration mode and you can enter the necessary commands to complete your desired tasks.

Step 7 To exit the Telnet session, use the **exit** or **logout** command as shown in the following example:

```
Router# logout
```

Accessing the CLI from a Remote Console Using a Modem

To access the switch remotely using a modem through an asynchronous connection, connect the modem to the console port.

The console port on a Catalyst 6500 Series switch is an EIA/TIA-232 asynchronous, serial connection with hardware flow control and an RJ-45 connector. The console port is located on the front panel of the supervisor engine, as shown in Figure 1-3 and Figure 1-4.

Figure 1-3 Supervisor Engine 720 Console Port Connector

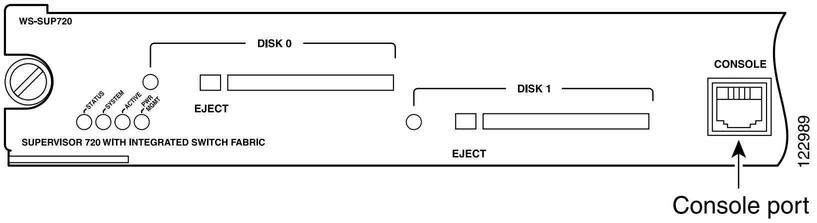
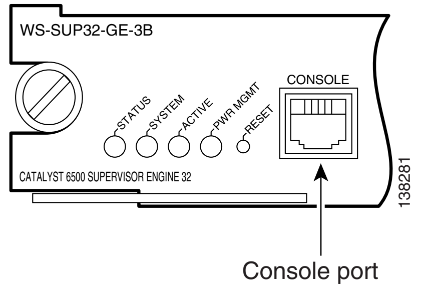


Figure 1-4 Supervisor Engine 32 Console Port Connector



To connect a modem to the console port, place the console port mode switch in the in position. Connect to the port using the RJ-45-to-RJ-45 cable and the RJ-45-to-DB-25 DCE adapter (labeled "Modem").

Using Keyboard Shortcuts

Commands are not case sensitive. You can abbreviate commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters.

Table 1-1 lists the keyboard shortcuts for entering and editing commands.

Table 1-1 Keyboard Shortcuts

Keystrokes	Purpose
Ctrl-B or the **Left Arrow key**[1]	Move the cursor back one character
Ctrl-F or the **Right Arrow key**[1]	Move the cursor forward one character
Ctrl-A	Move the cursor to the beginning of the command line
Ctrl-E	Move the cursor to the end of the command line
Esc B	Move the cursor back one word
Esc F	Move the cursor forward one word

[1] The arrow keys function only on ANSI-compatible terminals such as VT100s.

Using the History Buffer to Recall Commands

The history buffer stores the last 20 commands you entered. History substitution allows you to access these commands without retyping them, by using special abbreviated commands.

Table 1-2 lists the history substitution commands.

Table 1-2 History Substitution Commands

Command	Purpose
Ctrl-P or the **Up Arrow key**[1]	Recall commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or the **Down Arrow key**[1]	Return to more recent commands in the history buffer after recalling commands with **Ctrl-P** or the **Up Arrow key**.
`Router# show history`	While in EXEC mode, list the last several commands you have just entered.

[1] The arrow keys function only on ANSI-compatible terminals such as VT100s.

Understanding Command Modes

You use the CLI to access Cisco IOS software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode that you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode you can issue any EXEC command—user or privileged mode—or you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

Table 1-3 describes how to access and exit various common command modes of the Cisco IOS software. It also shows examples of the prompts displayed for each mode.

Table 1-3 Accessing and Exiting Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** EXEC command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, use the **configure terminal** privileged EXEC command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, use the **exit** or **end** command.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config-if)#`	To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command.
ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the **Break** key during the first 60 seconds while the system is booting.	`>`	To exit ROM monitor mode, use the **continue** command.

For more information on command modes, refer to the “Using the Command-Line Interface” chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2*.

Getting Help

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help specific to a command mode, a command, a keyword, or an argument, use one of the following commands:

Table 1-4 Help Commands and Purpose

Command	Purpose
help	Provides a brief description of the help system in any command mode.
*abbreviated-command-entry***?**	Provides a list of commands that begin with a particular character string. (No space between command and question mark.)
abbreviated-command-entry<**Tab**>	Completes a partial command name.
?	Lists all commands available for a particular command mode.
command **?**	Lists the keywords or arguments that you must enter next on the command line. (Space between command and question mark.)

Finding Command Options Example

This section provides an example of how to display syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering part of a command followed by a space. The Cisco IOS software displays a list and brief description of available keywords and arguments. For example, if you were in global configuration mode and wanted to see all the keywords or arguments for the **arap** command, you would type **arap ?**.

The <cr> symbol in command help output stands for "carriage return." On older keyboards, the carriage return key is the Return key. On most modern keyboards, the carriage return key is the Enter key. The <cr> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <cr> symbol are optional. The <cr> symbol by itself indicates that no more arguments or keywords are available and that you must press **Enter** to complete the command.

Table 1-5 shows examples of how you can use the question mark (**?**) to assist you in entering commands.

Table 1-5 Finding Command Options

Command	Comment
Router> **enable** Password: *<password>* Router#	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to a "#" from the ">"; for example, `Router> to Router#`.
Router# **configure terminal** Enter configuration commands, one per line. End with CNTL/Z. Router(config)#	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to `Router(config)#`.
Router(config)# **interface serial ?** <0-6> Serial interface number Router(config)# **interface serial 4 ?** / Router(config)# **interface serial 4/ ?** <0-3> Serial interface number Router(config)# **interface serial 4/0 ?** <cr> Router(config)# **interface serial 4/0** Router(config-if)#	Enter interface configuration mode by specifying the serial interface that you want to configure using the **interface serial** global configuration command. Enter **?** to display what you must enter next on the command line. In this example, you must enter the serial interface slot number and port number, separated by a forward slash. When the <cr> symbol is displayed, you can press **Enter** to complete the command. You are in interface configuration mode when the prompt changes to `Router(config-if #`.

Table 1-5 Finding Command Options (continued)

Command	Comment

`Router(config-if)# ?`
`Interface configuration commands:`
`  .`
`  .`
`  .`
`  ip                  Interface Internet Protocol config commands`
`  keepalive           Enable keepalive`
`  lan-name            LAN Name command`
`  llc2                LLC2 Interface Subcommands`
`  load-interval       Specify interval for load calculation for an`
`                      interface`
`  locaddr-priority    Assign a priority group`
`  logging             Configure logging for interface`
`  loopback            Configure internal loopback on an interface`
`  mac-address         Manually set interface MAC address`
`  mls                 mls router sub/interface commands`
`  mpoa                MPOA interface configuration commands`
`  mtu                 Set the interface Maximum Transmission Unit (MTU)`
`  netbios             Use a defined NETBIOS access list or enable`
`                      name-caching`
`  no                  Negate a command or set its defaults`
`  nrzi-encoding       Enable use of NRZI encoding`
`  ntp                 Configure NTP`
`  .`
`  .`
`  .`
`Router(config-if)#`
	Enter **?** to display a list of all the interface configuration commands available for the serial interface. This example shows only some of the available interface configuration commands.

`Router(config-if)# ip ?`
`Interface IP configuration subcommands:`
`  access-group        Specify access control for packets`
`  accounting          Enable IP accounting on this interface`
`  address             Set the IP address of an interface`
`  authentication      authentication subcommands`
`  bandwidth-percent   Set EIGRP bandwidth limit`
`  broadcast-address   Set the broadcast address of an interface`
`  cgmp                Enable/disable CGMP`
`  directed-broadcast  Enable forwarding of directed broadcasts`
`  dvmrp               DVMRP interface commands`
`  hello-interval      Configures IP-EIGRP hello interval`
`  helper-address      Specify a destination address for UDP broadcasts`
`  hold-time           Configures IP-EIGRP hold time`
`  .`
`  .`
`  .`
`Router(config-if)# ip`
	Enter the command that you want to configure for the interface. This example uses the **ip** command.
	Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.

Table 1-5 Finding Command Options (continued)

Command	Comment

`Router(config-if)# ip address ?`
`  A.B.C.D             IP address`
`  negotiated          IP Address negotiated over PPP`
`Router(config-if)# ip address`
	Enter the command that you want to configure for the interface. This example uses the **ip address** command.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword.
	A carriage return (<cr>) is not displayed; therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 ?`
`  A.B.C.D             IP subnet mask`
`Router(config-if)# ip address 172.16.0.1`
	Enter the keyword or argument that you want to use. This example uses the 172.16.0.1 IP address.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask.
	A <cr> is not displayed; therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0 ?`
`  secondary           Make this IP address a secondary address`
`  <cr>`
`Router(config-if)# ip address 172.16.0.1 255.255.255.0`
	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask.
	Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**.
	A <cr> is displayed; you can press **Enter** to complete the command, or you can enter another keyword.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0`
`Router(config-if)#`
	In this example, **Enter** is pressed to complete the command.

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a function. Use the command without the **no** keyword to re-enable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command; to re-enable IP routing, use the **ip routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Many CLI commands also have a **default** form. By issuing the command **default** *command-name*, you can configure the command to its default setting. The Cisco IOS software command reference publications describe the function of the **default** form of the command when the **default** form performs a different function than the plain and **no** forms of the command. To see what default commands are available on your system, enter **default ?** in the appropriate command mode.

Saving Configuration Changes

Use the **copy running-config startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs. For example:

```
Router# copy running-config startup-config
Building configuration...
```

It might take a minute or two to save the configuration. After the configuration has been saved, the following output appears:

```
[OK]
Router#
```

On most platforms, this task saves the configuration to NVRAM. On the Class A Flash file system platforms, this task saves the configuration to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Filtering Output from the show and more Commands

You can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

To use this functionality, enter a **show** or **more** command followed by the “pipe” character (|); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case sensitive):

show *command* | {**begin** | **include** | **exclude**} *regular-expression*

The output matches certain lines of information in the configuration file. The following example illustrates how to use output modifiers with the **show interface** command when you want the output to include only lines in which the expression “protocol” appears:

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

For more information on the search and filter functionality, refer to the “Using the Command-Line Interface” chapter in the *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*.

Finding Support Information for Platforms and Cisco Software Images

Cisco IOS software is packaged in feature sets consisting of software images that support specific platforms. The feature sets available for a specific platform depend on which Cisco IOS software images are included in a release. To identify the set of software images available in a specific release or to find out if a feature is available in a given Cisco IOS software image, you can use Cisco Feature Navigator or the software release notes.

Using Cisco Feature Navigator

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Using Software Advisor

To see if a feature is supported by a Cisco IOS release, to locate the software document for that feature, or to check the minimum software requirements of Cisco IOS software with the hardware installed on your switch, Cisco maintains the Software Advisor tool on Cisco.com at http://www.cisco.com/cgi-bin/Support/CompNav/Index.pl.

You must be a registered user on Cisco.com to access this tool.

Using Software Release Notes

Cisco IOS software releases include release notes that provide the following information:

- Platform support information
- Memory recommendations
- New feature information
- Open and resolved severity 1 and 2 caveats for all platforms

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative in providing information about features that first appeared in previous releases. Refer to Cisco Feature Navigator for cumulative feature information.

CHAPTER 2

SIP, SSC, and SPA Product Overview

This chapter provides an introduction to SPA interface processors (SIPs), SPA services cards (SSCs), and shared port adapters (SPAs). It includes the following sections:

- Introduction to SIPs, SSCs, and SPAs, page 2-1
- SIP, SSC, and SPA Compatibility, page 2-3
- Modular Optics Compatibility, page 2-6

For more hardware details for the specific SIPs, SSCs, and SPAs that are supported on the Catalyst 6500 Series switch, refer to the companion publication, *Cisco 7600 Series Router SIP, SSC, and SPA Hardware Installation Guide*.

Introduction to SIPs, SSCs, and SPAs

SIPs, SSCs, and SPAs are a new carrier card and port adapter architecture to increase modularity, flexibility, and density across Cisco Systems switches for network connectivity. This section describes the SIPs, SSCs, and SPAs and provides some guidelines for their use.

SPA Interface Processors

The following list describes some of the general characteristics of a SIP:

- A SIP is a carrier card that inserts into a switch slot like a line card. It provides no network connectivity on its own.
- A SIP contains one or more subslots, which are used to house one or more SPAs. The SPA provides interface ports for network connectivity.
- During normal operation the SIP should reside in the switch fully populated either with functional SPAs in all subslots, or with a blank filler plate (SPA-BLANK=) inserted in all empty subslots.
- SIPs support online insertion and removal (OIR) with SPAs inserted in their subslots. SPAs also support OIR and can be inserted or removed independently from the SIP.

SPA Services Cards

The following list describes some of the general charateristics of an SSC:

- An SSC is a carrier card that inserts into a switch slot like a line card. It provides no network connectivity.
- An SSC provides one or more subslots, which are used to house one or more SPAs. The supported SPAs do not provide interface ports for network connectivity, but provide certain services.
- During normal operation the SSC should reside in the switch fully populated either with functional SPAs in all subslots, or with a blank filler plate (SPA-BLANK=) inserted in all empty subslots.
- SSCs support online insertion and removal (OIR) with SPAs inserted in their subslots. SPAs also support OIR and can be inserted or removed independently from the SSC.

Shared Port Adapters

The following list describes some of the general characteristics of a SPA:

- A SPA is a modular type of port adapter that inserts into a subslot of a compatible SIP carrier card to provide network connectivity and increased interface port density. A SIP can hold one or more SPAs, depending on the SIP type.
- Some SPAs provide services rather than network connectivity, and insert into subslots of compatible SSCs. For example, the IPsec VPN SPA provides services such as IP security (IPsec) encryption/decryption, generic routing encapsulation (GRE), and Internet Key Exchange (IKE) key generation.
- SPAs are available in the following sizes, as shown in Figure 2-1 and Figure 2-2:
  - Single-height SPA—Inserts into one SIP subslot.
  - Double-height SPA—Inserts into two single, vertically aligned SIP subslots.

Figure 2-1 Single-Height and Double-Height SPA Sizes

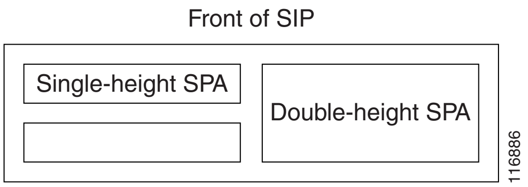
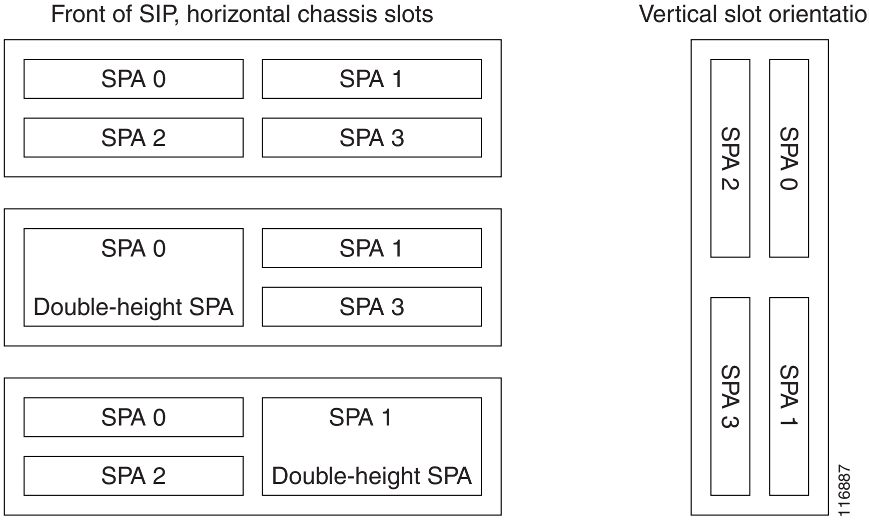


Figure 2-2 *Horizontal and Vertical Chassis Slot Orientation for SPAs*



- Each SPA provides a certain number of connectors, or ports, that are the interfaces to one or more networks. These interfaces can be individually configured using the Cisco IOS command-line interface (CLI).
- Either a blank filler plate or a functional SPA should reside in every subslot of an SIP during normal operation to maintain cooling integrity. Blank filler plates are available in single-height form only.
- SPAs support online insertion and removal (OIR). They can be inserted or removed independently from the SIP. SIPs also support online insertion and removal (OIR) with SPAs inserted in their subslots.

SIP, SSC, and SPA Compatibility

The following tables show SIP and SPA compatibility by SPA technology area on the Catalyst 6500 Series switch.

Note For more information about the introduction of support for different SIPs and SPAs, refer to the "Release History" sections in the overview chapters of this guide.

Note Do not install the IPsec VPN SPA in the same chassis as a Cisco 7600 SIP-600.

Table 2-1 *SIP and SPA Compatibility Table for ATM SPAs*

SPA	Product ID	SIP Type			
		SIP-200	SIP-400	SIP-600	SSC-400
2-Port and 4-Port OC-3c/STM-1 ATM SPA	SPA-2XOC3-ATM SPA-4XOC3-ATM	(Note1)	(Note1)	No	No

Table 2-1 SIP and SPA Compatibility Table for ATM SPAs (continued)

SPA	Product ID	SIP Type			
1-Port OC-12c/STM-4 ATM SPA	SPA-1XOC12-ATM	No	(Note1)	No	No
1-Port OC-48c/STM-16 ATM SPA	SPA-1XOC48-ATM	No	(Note1)	No	No

Table 2-2 SIP and SPA Compatibility Table for Ethernet SPAs

SPA	Product ID	SIP Type			
		SIP-200	SIP-400	SIP-600	SSC-400
1-Port 10-Gigabit Ethernet SPA	SPA-1XTENGE-XENPK SPA-1XTENGE-XFP	No	No	(Note3)	No
	SPA-1X10GE-L-V2	No	(Note6)	(Note6)	No
2-Port Gigabit Ethernet SPA	SPA-2X1GE	No	Yes	No	No
5-Port Gigabit Ethernet SPA	SPA-5X1GE	No	Yes	(Note3)	No
	SPA-5X1GE-V2	No	(Note5)	No	No
10-Port Gigabit Ethernet SPA	SPA-10X1G	No	No	(Note3)	No
4-Port and 8-Port Fast Ethernet SPA	SPA-4X1FE-V2 SPA-8X1FE-V2	(Note2)	No	No	No

Table 2-3 SIP and SPA Compatibility Table for the IPsec VPN SPA

SPA	Product ID	SIP Type			
		SIP-200	SIP-400	SIP-600	SSC-400
IPsec VPN SPA	SPA-IPSEC-2G	No	No	No	Yes

Table 2-4 SIP and SPA Compatibility Table for POS SPAs

SPA	Product ID	SIP Type			
		SIP-200	SIP-400	SIP-600	SSC-400
2-Port and 4-Port OC-3c/STM-1 POS SPA	SPA-2XOC3-POS SPA-4XOC3-POS	Yes	Yes	No	No
1-Port OC-12c/STM-4 POS SPA	SPA-1XOC12-POS	No	Yes	No	No
1-Port OC-48c/STM-16 POS SPA	SPA-1XOC48POS/RPR	No	(Note4)	No	No
2-Port OC-48c/STM-16 POS SPA	SPA-2XOC48POS/RPR	No	No	(Note5)	No
1-Port OC-192c/STM-64 POS/RPR SPA	SPA-OC192POS-LR SPA-OC192POS-VSR SPA-OC192POS-XFP	No	No	(Note3)	No

Table 2-5 SIP and SPA Compatibility Table for Serial SPAs

SPA	Product ID	SIP Type			
		SIP-200	SIP-400	SIP-600	SSC-400
1-Port Channelized OC-3/STM-1 SPA	SPA-1XCHSTM1/OC3	(Note2)	No	No	No
2-Port and 4-Port Channelized T3 SPA	SPA-2XCT3/DS0 SPA-4XCT3/DS0	Yes	(Note5)	No	No
2-Port and 4-Port Clear Channel T3/E3 SPA	SPA-2XT3/E3 SPA-4XT3/E3	Yes	(Note5)	No	No
8-Port Channelized T1/E1 SPA	SPA-8XCHT1/E1	Yes	(Note5)	No	No

Table 2-6 SIP and SPA Compatibility Table for CEoP SPAs

SPA	Product ID	SIP Type			
		SIP-200	SIP-400	SIP-600	SSC-400
24-Port Channelized T1/E1/J1 CEoP SPA	SPA-24CHT1-CE-ATM	No	Yes	No	No

The following notes apply to the SIP, SSC, and SPA compatibility tables:

- Note1—Supported in 12.2SXE and SXF. Support removed in 12.2(33)SXH. Support restored in 12.2(33)SXI.
- Note2—Support added in 12.2(33)SXH.
- Note3—Supported in 12.2SXF. Support removed in 12.2(33)SXH.
- Note4—Support added in 12.2(18)SXF10.
- Note5—Support added in 12.2(33)SXI.
- Note6—Support added in 12.2(33)SXI2.

Modular Optics Compatibility

Some SPAs implement small form-factor pluggable (SFP) optical transceivers to provide network connectivity. An SFP module is a transceiver device that mounts into the front panel to provide network connectivity.

Cisco Systems qualifies the SFP modules that can be used with SPAs.

Note The SPAs will accept only the SFP modules listed as supported in this document. An SFP check is run every time an SFP module is inserted into a SPA and only SFP modules that pass this check will be usable.

Table 2-7 shows the types of optics modules that have been qualified for use with a SPA:

Table 2-7 SPA Optics Compatibility

SPA	Qualified Optics Modules (Cisco Part Numbers)
2-Port and 4-Port OC-3c/STM-1 ATM SPA	• SFP-OC3-MM • SFP-OC3-SR • SFP-OC3-IR1 • SFP-OC3-LR1 • SFP-OC3-LR2
1-Port OC-12c/STM-4 ATM SPA	• SFP-OC12-MM • SFP-OC12-SR • SFP-OC12-IR1 • SFP-OC12-LR1 • SFP-OC12-LR2
1-Port OC-48c/STM-16 ATM SPA	• SFP-OC48-IR1 • SFP-OC48-SR
1-Port 10-Gigabit Ethernet SPA	• XFP-10GLR-OC192SR • XFP-10GER-OC192IR • SFP-GE-T
2-Port Gigabit Ethernet SPA	• SFP-GE-S • SFP-GE-L • SFP-GE-Z • SFP-GE-T
5-Port Gigabit Ethernet SPA	• SFP-GE-S • SFP-GE-L • SFP-GE-Z • SFP-GE-T

Table 2-7 SPA Optics Compatibility (continued)

SPA	Qualified Optics Modules (Cisco Part Numbers)
10-Port Gigabit Ethernet SPA	• SFP-GE-S • SFP-GE-L • SFP-GE-Z • SFP-GE-T
2-Port and 4-Port OC-3c/STM-1 POS SPA	• SFP-OC3-MM • SFP-OC3-SR • SFP-OC3-IR1 • SFP-OC3-LR1 • SFP-OC3-LR2
1-Port OC-12c/STM-4 POS SPA	• SFP-OC12-MM • SFP-OC12-SR • SFP-OC12-IR1 • SFP-OC12-LR1 • SFP-OC12-LR2
1-Port OC-48c/STM-16 POS SPA	• SFP-OC48-SR • SFP-OC48-IR1 • SFP-OC48-LR2
1-Port OC-192c/STM-64 POS/RPR XFP SPA	• XFP-10GLR-OC192SR • XFP-10GER-OC192IR
1-Port Channelized OC-3/STM-1 SPA	• SFP-OC3-SR • SFP-OC3-IR1 • SFP-OC3-LR1 • SFP-OC3-LR2
1-Port Channelized OC-3 ATM CEoP SPA	• SFP-OC3-MM • SFP-OC3-SR • SFP-OC3-IR1 • SFP-OC3-LR1 • SFP-OC3-LR2

PART 2

SPA Interface Processors and SPA Services Cards

CHAPTER 3

Overview of the SIPs and SSC

This chapter provides an overview of the release history, and feature and Management Information Base (MIB) support for the Cisco 7600 SIP-200, Cisco 7600 SIP-400, Cisco 7600 SIP-600, and Cisco 7600 SSC-400.

This chapter includes the following sections:

- Release History, page 3-1
- Supported SIP Features, page 3-3
- Supported SSC Features, page 3-15
- Restrictions, page 3-15
- Supported MIBs, page 3-20
- Displaying the SIP and SSC Hardware Type, page 3-21

Release History

Note For release history information about the introduction of SPA support on the SIPs, refer to the corresponding "Overview" chapters in the SPA technology sections of this document. In addition, features specific to certain SPA technologies are documented in the corresponding SPA sections of this document.

Release	Modification
12.2(33)SXI	Support for the Cisco 7600 SIP-600 was restored. Support for Asymmetric Carrier Delay was introduced on the Cisco 7600 SIP-200 and Cisco 7600 SIP-400.

12.2(33)SXH	Support for the Cisco 7600 SIP-600 was removed. Support for the following features was introduced on the Cisco 7600 SIP-200: • BCP over dMLPPP • MPLS over RBE • Multi-VC to VLAN scalability • QoS Support on Bridging Features • Software-based dMLPPP • Software-based dMLFR Support for the following features was introduced on the Cisco 7600 SIP-400: • Ethernet Over MPLS (EoMPLS) VC Scaling • Ingress/Egress COS Classification with Ingress Policing per VLAN or EoMPLS VC • Hierarchical VPLS (H-VPLS) with MPLS Edge • VPLS Multiple VCs per Spoke • Hierarchical QoS Support for Ethernet Over MPLS (EoMPLS) VCs • QoS Support on Bridging Features • Lawful Intercept
12.2(18)SXF	Support for the following SIP hardware was introduced on the Cisco 7600 series router and Catalyst 6500 series switch: • Cisco 7600 SIP-600 Support for the following features were introduced on the Cisco 7600 SIP-200: • Software-based MLPPP • Software-based MLFR Support for the following features were introduced on the Cisco 7600 SIP-400: • Policing by committed information rate (CIR) percentage • QoS matching on class of service (CoS)—2-Port Gigabit Ethernet SPA only.
12.2(18)SXE2	Support for the SPA services card (SSC) was introduced on the Cisco 7600 series router and Catalyst 6500 series switch: • Cisco 7600 SSC-400
12.2(18)SXE	Support for the following SPA interface processor (SIP) hardware was introduced on the Cisco 7600 series router and Catalyst 6500 series switch: • Cisco 7600 SIP-200 • Cisco 7600 SIP-400

Supported SIP Features

The Cisco 7600 SIP-200, Cisco 7600 SIP-400, and Cisco 7600 SIP-600 are high-performance, feature-rich SPA interface processors that function as carrier cards for shared port adapters (SPAs) on the Catalyst 6500 Series switch. These SIPs are supported on the Cisco 7600 series router and Catalyst 6500 series switch, and are compatible with one or more platform-independent SPAs. For more information on SPA compatibility, see the "SIP, SSC, and SPA Compatibility" section on page 2-3.

The Catalyst 6500 series switch can provide edge aggregation services, and the SIPs provide a cost-effective solution for customers seeking moderate- to high-port density and line rate services:

- The Cisco 7600 SIP-200 provides WAN edge aggregation through lower-speed and low-density SPAs for network environments requiring regional office connectivity to headquarters, or collapsed LAN/WAN deployment.
- The Cisco 7600 SIP-400 provides higher-speed, high-density link aggregation for network environments requiring leased line and metro aggregation.
- The Cisco 7600 SIP-600 provides a high-speed interface for WANs and metro aggregation.

Note Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

Note The Cisco 7600 SIP-600 should not be used in the same chassis with an IPsec VPN SPA.

This section provides a list of some of the primary features supported by the SIP hardware and software. For feature compatibility information by SIP and SPA combination, and information about configuring these features, see Chapter 4, "Configuring the SIPs and SSC."

Cisco 7600 SIP-200 Features

- Field-programmable device (FPD) upgrade support

  The Cisco 7600 SIP-200 supports the standard FPD upgrade methods for the Catalyst 6500 Series switch. For more information about FPD support, see Chapter 31, "Upgrading Field-Programmable Devices."

Cisco 7600 SIP-200 High Availability Features

- Automatic protection switching (APS)—ATM and POS SPAs
- Online insertion and removal (OIR) of the SIP and SPAs
- Nonstop Forwarding (NSF)
- Stateful switchover (SSO)

Cisco 7600 SIP-200 ATM Features

- Aggregate Weighted Random Early Detection (WRED)
- ATM Adaptation Layer 5 (AAL5) Subnetwork Access Protocol (SNAP)

- AAL5 over Multiprotocol Label Switching (MPLS)
- ATM virtual circuit (VC) bundles
- RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5,* Multipoint Bridging (MPB) on the 2-Port and 4-Port OC-3c/STM-1 ATM SPA
- VC bundle Class of Service (CoS) precedence mapping

For a comprehensive list of supported and unsupported ATM features, SIP-dependent features, and restrictions see Chapter 6, "Overview of the ATM SPAs."

Cisco 7600 SIP-200 Frame Relay Features

For additional Frame Relay features, see also the MPLS and Quality of Service (QoS) feature sections.

Note Based on your link configuration, Multilink PPP (MLPPP) and Multilink Frame Relay (MLFR) are either software-based on the Cisco 7600 SIP-200, or hardware-based on the 8-Port Channelized T1/E1 SPA and 2-Port and 4-Port Channelized T3 SPAs. For more information, see the corresponding configuration chapters for the SIPs and the serial SPAs.

- Distributed Multilink Frame Relay (dMLFR) (FRF.16)
- Distributed Link Fragmentation and Interleaving (dLFI) over Multilink PPP (MLPPP)
- dLFI with FRF.12
- Frame Relay over MPLS (FRoMPLS)
- Frame Relay VC bundles
- Frame Relay switching
- RFC 1490, *Multiprotocol Interconnect over Frame Relay*, Multipoint Bridging (MPB) on the 2-Port and 4-Port Clear Channel T3/E3 SPA, 2-Port and 4-Port Channelized T3 SPA, and the 8-Port Channelized T1/E1 SPA
- VC bundle Class of Service (CoS) precedence mapping

Cisco 7600 SIP-200 MPLS Features

- Explicit null
- Label disposition
- Label imposition
- Label swapping
- QoS tunneling
- Virtual private network (VPN) routing/forwarding (VRF) instance description
- MLPPP with MPLS on VPN
- Any Transport over MPLS (AToM) support, including:
  - ATM over MPLS (ATMoMPLS)—AAL5 VC mode
  - Ethernet over MPLS (EoMPLS)—(Single cell relay) VC mode
  - Frame Relay over MPLS (FRoMPLS)

  - High-Level Data Link Control (HDLC) over MPLS (HDLCoMPLS)
  - PPP over MPLS (PPPoMPLS)
- Hierarchical QoS for EoMPLS VCs

Beginning in Cisco IOS Release 12.2(33)SXH, the Cisco 7600 SIP-200 adds the following MPLS feature support:

- MPLS over RBE—ATM SPAs only

Cisco 7600 SIP-200 MPLS Classification

- Default copy of IP precedence to MPLS experimental (EXP) bit
- Match on MPLS EXP bit using Modular QoS CLI (MQC)

Cisco 7600 SIP-200 MPLS Congestion Management

- Low latency queueing (LLQ)
- Class-based weighted fair queueing (CBWFQ)

Cisco 7600 SIP-200 MPLS Encapsulations

- ATM AAL5 SNAP
- Frame Relay
- HDLC
- MLPPP
- PPP

Cisco 7600 SIP-200 MPLS Marking

- Set MPLS EXP bit using MQC

Cisco 7600 SIP-200 MPLS Traffic Shaping

- Traffic shaping using MQC

Cisco 7600 SIP-200 Multiservice Features

- Compressed Real-Time Protocol (cRTP)—Supported on the Cisco 7600 SIP-200 with the 8-Port Channelized T1/E1 SPA, 2-Port and 4-Port Channelized T3 SPA, and 2-Port and 4-Port Clear Channel T3/E3 SPA
- FRF.11

Cisco 7600 SIP-200 QoS Features

This section provides a list of the Quality of Service (QoS) features that are supported by the Cisco 7600 SIP-200.

Cisco 7600 SIP-200 ATM SPA QoS Implementation

For the 2-Port and 4-Port OC-3c/STM-1 ATM SPA, the following applies:

- In the ingress direction, all Quality of Service (QoS) features are supported by the Cisco 7600 SIP-200.
- In the egress direction:
  - All queueing based features (such as class-based weighted fair queueing [CBWFQ], and ATM per-VC WFQ) are implemented on the Segmentation and Reassembly (SAR) processor on the SPA.
  - Policing is implemented on the SIP.
  - Class queue shaping is not supported.

Cisco 7600 SIP-200 Packet Marking

- IP precedence
- Differentiated Services Code Point (DSCP)
- Class-based marking
- ATM cell loss priority (CLP) to EXP marking/Type of Service (ToS)/DSCP
- Frame relay discard eligibility (DE) to EXP marking/ToS/DSCP

Cisco 7600 SIP-200 Policing and Dropping

- Aggregate
- Dual rate
- Hierarchical
- DSCP Markdown
- Policing—Precedence, DSCP marking
- Policing—EXP marking
- Explicit Drop in Class
- Matching packet length

Cisco 7600 SIP-200 Classification Into a Queue

- MPLS EXP
- ACL number
- Configurable queue size
- Network-based application recognition (NBAR)/dSTILE

Cisco 7600 SIP-200 Congestion Management

- Weighted fair queueing (WFQ)
- Class-based weighted fair queueing (CBWFQ)
- Per-VC CBWFQ
- Allocation, DSCP, EXP and precedence matching
- LLQ or priority queueing (strict priority only)
- Configurable LLQ burst size

Cisco 7600 SIP-200 Congestion Avoidance

- Random early detection (RED)
- Weighted random early detection (WRED)
- Diffserv-compliant WRED
- Aggregate WRED—ATM SPAs only

Cisco 7600 SIP-200 Shaping

- Generic traffic shaping (GTS)/Distributed traffic shaping (DTS)
- Hierarchical service policy with GTS
- Hierarchical traffic shaping FR
- Hierarchical traffic shaping FR adaptive to FECN, BECN (Cisco 7600 SIP-200 only)
- Hierarchical traffic shaping for PPP and HDLC
- Ingress shaping
- Egress shaping

Note Egress shaping is not supported on the Cisco 7600 SIP-200 for the 2-Port and 4-Port OC-3c/STM-1 ATM SPA.

- Shaping by percentage

Cisco 7600 SIP-200 Other QoS Features

- Hierarchical QoS for EoMPLS VCs
- QoS with MLPPP

Beginning in Cisco IOS Release 12.2(33)SXH, the Cisco 7600 SIP-200 adds the following QoS feature support:

- QoS Support on Bridging Features

Cisco 7600 SIP-200 Fragmentation Features

- dLFI with ATM
- dLFI over MLPPP
- FRF.12

Cisco 7600 SIP-200 Layer 2 Protocols and Encapsulation

- AAL5 Network Layer protocol ID (NLPID)
- AAL5 SNAP
- Cisco Frame Relay
- IETF Frame Relay
- Frame Relay two-octet header
- Frame Relay BECN/FECN
- Frame Relay PVC
- Frame Relay UNI
- HDLC
- MLPPP
- PPP

Cisco 7600 SIP-200 Layer 2 Interworking

- ATM VC trunk emulation
- Bridged and routed RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*
- RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5,* Multipoint Bridging (MPB) on the 2-Port and 4-Port OC-3c/STM-1 ATM SPA
- RFC 1490, *Multiprotocol Interconnect over Frame Relay*, Multipoint Bridging (MPB) on the 2-Port and 4-Port Clear Channel T3/E3 SPA, 2-Port and 4-Port Channelized T3 SPA, and the 8-Port Channelized T1/E1 SPA
- Bridging of Routed Encapsulations (BRE)
- Routed bridged encapsulation (RBE)

Note RBE is not supported when using the Intermediate System-to-Intermediate System (IS-IS) routing protocol.

- RFC 3518, *Point-to-Point Protocol (PPP) Bridging Control Protocol (BCP)* on the 2-Port and 4-Port Clear Channel T3/E3 SPA, 2-Port and 4-Port Channelized T3 SPA, and the 8-Port Channelized T1/E1 SPA

Beginning in Cisco IOS Release 12.2(33)SXH, the Cisco 7600 SIP-200 adds the following Layer 2 interworking feature support:

- BCP Support Over MLPPP
- Multi-VC to VLAN scalability
- QoS Support on Bridging

Cisco 7600 SIP-400 Features

- FPD upgrade support

  The Cisco 7600 SIP-400 supports the standard FPD upgrade methods for the Catalyst 6500 Series switch. For more information about FPD support, see Chapter 31, "Upgrading Field-Programmable Devices."

Cisco 7600 SIP-400 High Availability Features

- Automatic protection switching (APS)—ATM and POS SPAs
- Online insertion and removal (OIR) of the SIP and SPAs
- Stateful switchover (SSO)

Cisco 7600 SIP-400 MPLS Features

Note For the Cisco 7600 SIP-400, the following MPLS features are implemented on the Supervisor Engine 720 PFC3B and Supervisor Engine 720 PFC3BXL: Label imposition, label swapping, label disposition, explicit null, default copy of IP precedence to EXP bit classification, and QoS tunneling. For more information about the requirements for Policy Feature Cards (PFCs) on the Catalyst 6500 Series switch, refer to the For more information about the requirements for Policy Feature Cards (PFCs) on the Catalyst 6500 Series switch, refer to the *Release Notes for Cisco IOS Release 12.2(33)SXH and Later Releases* at the following URL:
http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/release/notes/ol_14271.html#wp2561312

- VRF description
- Any Transport over MPLS (AToM) support, including:
  - ATMoMPLS—AAL0 mode (single cell relay only)
  - ATMoMPLS—AAL5 mode
  - EoMPLS—Port mode
  - EoMPLS—VLAN mode
  - FRoMPLS—DLCI mode

Beginning in Cisco IOS Release 12.2(33)SXH, the Cisco 7600 SIP-400 adds the following MPLS feature support:

- Ingress/Egress COS Classification with Ingress Policing per VLAN or EoMPLS VC
- Hierarchical VPLS (H-VPLS) with MPLS Edge
- VPLS Multiple VCs per Spoke
- Hierarchical QoS Support for Ethernet Over MPLS (EoMPLS) VCs

Cisco 7600 SIP-400 MPLS Congestion Management

- LLQ
- CBWFQ

Cisco 7600 SIP-400 MPLS Encapsulations

- ATM AAL5 SNAP
- Ethernet/802.1q
- Frame Relay
- HDLC
- Generic Routing Encapsulation (GRE)—2-Port Gigabit Ethernet SPA only
- PPP

Cisco 7600 SIP-400 MPLS Marking

- Set MPLS EXP bits at tag imposition using MQC (**set mpls-experiment** command)—Input IP interface
- Set MPLS EXP bits on topmost label (set EXP topmost) using MQC (**set mpls-experiment topmost** command)—Input and output MPLS interface
- Mapping Ethernet 802.1q priority bits to MPLS EXP bits for EoMPLS

Cisco 7600 SIP-400 QoS Features

This section provides a list of the Quality of Service (QoS) features that are supported by the Cisco 7600 SIP-400.

Cisco 7600 SIP-400 Packet Marking

- IP precedence (**set ip precedence** command)—Input and output
- DSCP (**set dscp** command)—Input and output
- Class-based marking
- DE to EXP marking/ToS/DSCP
- CLP to EXP marking/ToS/DSCP
- Ethernet 802.1q priority bits to EXP marking (EoMPLS)

Cisco 7600 SIP-400 Policing and Dropping

- Dual rate
- Hierarchical
- Dual-rate policer with three-color marker
- Policing—Percent
- Policing—Precedence, DSCP marking
- Policing—EXP marking
- Policing—Set ATM CLP, FR DE
- Policing—Set MPLS EXP bits on topmost label (set EXP topmost)
- Explicit Drop in Class

Cisco 7600 SIP-400 Classification Into a Queue

- Access control lists (IPv4 and IPv6)
  - Access group (**match access-group** command)—Input and output
  - Address (IPv6 compress mode only)
  - Name
  - Number
  - Source and destination port
  - TCP flag (IPv4 only)
- ATM CLP (**match atm clp** command)—Input ATM interface
- Configurable queue size
- CoS (**match cos** command)—Input and output dot1q tagged frames for 2-Port Gigabit Ethernet SPA only
- Frame Relay DE (**match fr-de** command)—Input Frame Relay interface
- IP DSCP (**match dscp** command)—Input and output
- IP precedence (**match ip precedence** command)—Input and output
- MPLS EXP (**match mpls experimental** command)—Input and output MPLS interface
- Multiple matches per class map (up to 8)

Beginning in Cisco IOS Release 12.2(33)SXH, the Cisco 7600 SIP-400 adds the following QoS classification feature support:

- Ingress/Egress COS Classification with Ingress Policing per VLAN or EoMPLS VC

Cisco 7600 SIP-400 Congestion Management

- CBWFQ
- Per-VC CBWFQ
- DSCP, EXP and Precedence matching
- LLQ or priority queueing (strict priority only)

Cisco 7600 SIP-400 Congestion Avoidance

- RED
- WRED
- Diffserv-compliant WRED
- Aggregate WRED—ATM SPAs only

Cisco 7600 SIP-400 Shaping

- Hierarchical traffic shaping using class-default (not supported for user-defined class)
- Hierarchical traffic shaping FR
- Hierarchical traffic shaping for PPP and HDLC
- Egress shaping

Cisco 7600 SIP-400 Fragmentation Features

- dLFI with ATM

Cisco 7600 SIP-400 Layer 2 Protocols and Encapsulation

- PPP
- AAL5 SNAP
- HDLC
- Cisco Frame Relay
- IETF Frame Relay
- Frame Relay two-octet header
- Frame Relay BECN/FECN
- Frame Relay PVC
- Frame Relay UNI

Cisco 7600 SIP-400 Layer 2 Interworking

- Bridged and routed RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*
- RFC 3518, *Point-to-Point Protocol (PPP) Bridging Control Protocol (BCP)* on the 2-Port and 4-Port Clear Channel T3/E3 SPA, 2-Port and 4-Port Channelized T3 SPA, and the 8-Port Channelized T1/E1 SPA

Beginning in Cisco IOS Release 12.2(33)SXH, the Cisco 7600 SIP-400 adds the following Layer 2 interworking feature support:

- QoS Support on Bridging Feature

Cisco 7600 SIP-600 Features

Note Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

Note The Cisco 7600 SIP-600 should not be used in the same chassis with an IPsec VPN SPA.

- FPD upgrade support

  The Cisco 7600 SIP-600 supports the standard FPD upgrade methods for the Catalyst 6500 Series switch. For more information about FPD support, see Chapter 31, "Upgrading Field-Programmable Devices."
- Layer 2 switch port
- EtherChannel and Link Aggregate Control Protocol (IEEE 802.3ad)
- Control Plane Policing (CPP)

Cisco 7600 SIP-600 High Availability Features

- Automatic protection switching (APS)
- Online insertion and removal (OIR) of the SIP and SPAs
- Nonstop Forwarding (NSF)
- Stateful switchover (SSO)

Cisco 7600 SIP-600 MPLS Features

- Unicast switching, with specific support for up to six label push operations, one label pop operation (2 label pop operation in case of Explicit Null), or one label swap with up to five label push operations, at each MPLS switch node.
- Support for Explicit Null label to preserve CoS information when forwarding packets from provider (P) to provider edge (PE) switches.
- Support for Implicit Null label to request that penultimate hop switch forward IP packets without labels to the switch at the end of the label switch path (LSP).
- VRF
- Traffic engineering
- Any Transport over MPLS (AToM) support—EoMPLS only
  - PFC-based (No MAC address learning)
  - SIP-based (MAC address learning, requires SIP as uplink)
  - Up to 4000 EoMPLS VCs per system
- Virtual Private LAN Service (VPLS) support, including:
  - H-VPLS on MPLS edge—H-VPLS with MPLS edge requires either an OSM module or Cisco 7600 SIP-600 in both the downlink (facing UPE) and uplink (MPLS core). For more information about configuring H-VPLS, see Chapter 10, “Configuring the Fast Ethernet and Gigabit Ethernet SPAs.”
  - H-VPLS with QinQ edge—Requires Cisco 7600 SIP-600 in the uplink, and any LAN port or Cisco 7600 SIP-600 on the downlink.
  - Up to 4000 VPLS domains
  - Up to 60 VPLS peers per domain
  - Up to 30,000 pseudo-wires, used in any combination of domains and peers up to the 4000-domain or 60-peer maximums. For example, support of up to 4000 domains with 7 peers or up to 60 peers in 500 domains.
- MPLS Operations and Maintenance (OAM) support, including:
  - LSP ping and traceroute
  - Virtual Circuit Connection Verification (VCCV)

Cisco 7600 SIP-600 Layer 2 Protocols and Encapsulation

- HDLC (Cisco Systems)
- PPP
- PPP over SONET/SDH

- Layer 2 Gigabit Ethernet support, including:
  - IEEE 802.3z 1000 Mbps Gigabit Ethernet
  - IEEE 802.3ab 1000BaseT Gigabit Ethernet
  - IEEE 802.3ae 10 Gbps Ethernet (1-Port 10-Gigabit Ethernet SPA only)
  - Jumbo frame (up to 9216 bytes)
  - ARPA, IEEE 802.3 SAP, IEEE 802.3 SNAP, QinQ
  - IEEE 802.1q VLANs
  - Autonegotiation support including IEEE 802.3 flow control and pause frames
  - Gigabit Ethernet Channel (GEC)
  - IEEE 802.3ad link aggregation
  - Address Resolution Protocol (ARP)/Reverse ARP (RARP
  - Hot Standby Router Protocol (HSRP)
  - Virtual Router Redundancy Protocol (VRRP)

Cisco 7600 SIP-600 QoS Features

This section provides a list of the Quality of Service (QoS) features that are supported by the Cisco 7600 SIP-600.

- MQC

Cisco 7600 SIP-600 Marking

- IP precedence (**set ip precedence** command)
- DSCP (**set dscp** command)
- MPLS EXP (**match mpls experimental** command)

Note Mapping 802.1p CoS values to MPLS EXP bits is supported using EoMPLS only.

Cisco 7600 SIP-600 Policing and Dropping

- Input policing on a per-port and per-VLAN basis

Cisco 7600 SIP-600 Classification Into a Queue

- Input and output ACLs on a per-port and per-VLAN basis
- Input VLAN (**match input vlan** command)
- IP DSCP (**match dscp** command)
- IP precedence (**match ip precedence** command)
- MPLS EXP (**match mpls experimental** command)
- QoS group (**match qos-group** command)
- VLAN (**match vlan** command)

Cisco 7600 SIP-600 Congestion Management

- CBWFQ
- LLQ

Cisco 7600 SIP-600 Congestion Avoidance

- WRED

Cisco 7600 SIP-600 Shaping

- Output shaping on a per-port and per-VLAN basis
- Output hierarchical traffic shaping—Two levels of shaping on an interface, subinterface, or group of subinterfaces

Supported SSC Features

The Cisco 7600 SSC-400 is a streamlined services card that provides a very high bandwidth data path between the Catalyst 6500 Series switch platform backplane and the high-speed interconnects on the IPsec VPN SPA.

For more information about the features and configuration supported by the IPsec VPN SPA with the Cisco 7600 SSC-400, see the related chapters in the IPsec VPN Shared Port Adapter section of this book.

Cisco 7600 SSC-400 Features

- Support of up to two IPsec VPN SPAs per slot
- Online insertion and removal (OIR) of the SSC and SPAs

Restrictions

This section documents unsupported features and feature restrictions for the SIPs and SSC on the Catalyst 6500 Series switch.

Cisco 7600 SIP-200 Restrictions

As of Cisco IOS Release 12.2(18)SXE, the Cisco 7600 SIP-200 has the following restrictions:

- The Cisco 7600 SIP-200 is not supported with a Supervisor Engine 1, Supervisor Engine 1A, Supervisor Engine 2, or Supervisor Engine 720A.
- A maximum number of 200 PVCs or SVCs using Link Fragmentation and Interleaving (LFI) is supported for all ATM SPAs (or other ATM modules) in a Catalyst 6500 Series switch.
- The following features are not supported:
  - Reliable PPP (RFC 1663, *PPP Reliable Transmission*)
  - Layer 2 Tunneling Protocol (L2TP) version 2

  - L2TP version 3
  - X.25, Link Access Procedure, Balanced (LAPB)
  - ATM LAN Emulation (LANE)
  - PPP over Ethernet (PPPoE)
  - STAC Compression
  - Legacy Priority Queueing and Custom Queueing
  - dLFI over Frame Relay (dLFIoFR)
  - dLFI with MPLS
- AToM (ATMoMPLS, FRoMPLS, HDLCoMPLS, and PPPoMPLs) on a SPA requires a Cisco 7600 SIP-200, FlexWAN, Enhanced FlexWAN, or OSM PXF interface as the core-facing interface.
- AToM (ATMoMPLS, FRoMPLS) on SIP-200 also are supported with a Cisco 7600 SIP-400 as the core-facing interface.

Cisco 7600 SIP-400 Restrictions

As of Cisco IOS Release 12.2(18)SXE, the Cisco 7600 SIP-400 has the following restrictions:

- The Cisco 7600 SIP-400 is not supported with a Supervisor Engine 1, Supervisor Engine 1A, or Supervisor Engine 2. It is also not supported with a Supervisor Engine 720 PFC3A, or in PFC3A mode.

  For more information about the requirements for Policy Feature Cards (PFCs) on the Catalyst 6500 Series switch, refer to the *Release Notes for Cisco IOS Release 12.2(33)SXH and Later Releases* at the following URL:
  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/release/notes/ol_14271.html#wp2561312
- The Cisco 7600 SIP-400 is not supported with PFC-2 based systems.
- A maximum number of 200 PVCs or SVCs using Link Fragmentation and Interleaving (LFI) is supported for all ATM SPAs (or other ATM modules) in a Catalyst 6500 Series switch.
- For AToM in releases prior to Cisco IOS Release 12.2(33)SXH, the Cisco 7600 SIP-400 does not support the following features when it is located in the data path. This means you should not configure the following features if the SIP is facing the customer edge (CE) or the MPLS core:
  - HDLCoMPLS
  - PPPoMPLS
  - Virtual Private LAN Service (VPLS)
- For AToM beginning in Cisco IOS Release 12.2(33)SXH, the Cisco 7600 SIP-400 supports the following features on CE-facing interfaces:
  - HDLCoMPLS
  - PPPoMPLS
  - Virtual Private LAN Service (VPLS)
- The Cisco 7600 SIP-400 supports EoMPLS with directly connected provider edge (PE) devices when the Cisco 7600 SIP-400 is on the MPLS core side of the network.

- The Cisco 7600 SIP-400 does not support the ability to enable or disable tunneling of Layer 2 packets, such as for the VLAN Trunking Protocol (VTP), Cisco Discovery Protocol (CDP), and bridge protocol data unit (BPDU). The Cisco 7600 SIP-400 tunnels BPDUs, and always blocks VTP and CDP packets from the tunnel.
- In ATMoMPLS AAL5 and cell mode, the Cisco 7600 SIP-400 supports non-matching VPIs/VCIs between PEs if the Cisco 7600 SIP-400 is on both sides of the network.
- The Cisco 7600 SIP-400 supports matching on FR-DE to set MPLS-EXP for FRoMPLS.
- The Cisco 7600 SIP-400 supports use of the **xconnect** command to configure AToM circuits for all AToM connection types except ATMoMPLS. For ATMoMPLS, you must use the **mpls l2 transport route** command.
- The Cisco 7600 SIP-400 supports local switching for Frame Relay and ATM interfaces.
- The Cisco 7600 SIP-400 does not support the following QoS classification features with AToM:
  - Matching on data-link connection identifier (DLCI) is unsupported.
  - Matching on virtual LAN (VLAN) is unsupported.
  - Matching on class of service (CoS) is unsupported is unsupported in Cisco IOS Release 12.2(18)SXE and Cisco IOS Release 12.2(18)SXE2 only. Beginning in Cisco IOS Release 12.2(18)SXF, it is supported with the 2-Port Gigabit Ethernet SPA.
  - Matching on input interface is unsupported.
  - Matching on packet length is unsupported.
  - Matching on media access control (MAC) address is unsupported.
  - Matching on protocol type, including Border Gateway Protocol (BGP), is unsupported.
- The Cisco 7600 SIP-400 does not support the following QoS classification features using MQC:
  - ACL IPv6 full address
  - ACL IPv6 TCP flags
  - Class map (**match class-map** command)
  - COS inner (**match cos inner** command)—Supported beginning in Cisco IOS Release 12.2(33)SXH on 2-Port Gigabit Ethernet SPA input and output interfaces and with bridging features.
  - Destination sensitive services (DSS)
  - Discard class (**match discard-class** command)
  - Frame Relay DLCI (**match fr-dlci** command)
  - Input interface (**match input-interface** command)
  - Input VLAN (**match input vlan** command)—Supported beginning in Cisco IOS Release 12.2(33)SXH on output interfaces only.
  - IP RTP (**match ip rtp** command)
  - IPv4 and IPv6 ToS
  - MAC address (**match mac** command)
  - Match protocol (**match protocol** command)—Support IP only
  - Packet length (**match packet length** command)
  - QoS group (**match qos-group** command)
  - Source and destination autonomous system (AS) (**match as** command)

  - Source and destination Border Gateway Protocol (BGP) community (**match bgp-community** command)
  - VLAN (**match vlan** command)
  - VLAN inner (**match vlan inner** command)—Supported beginning in Cisco IOS Release 12.2(33)SXH on input and output interfaces and with bridging features.
- The Cisco 7600 SIP-400 does not support the following QoS marking features:
  - CoS (**set cos** command)
  - CoS inner (**set cos inner** command)
- The Cisco 7600 SIP-400 does not support the following QoS marking features using MQC:
  - QoS group (**set qos-group** command)
  - Next-hop (**set next-hop** command)
  - Discard class (**set discard-class** command)
  - Table (**set table** command)
- The Cisco 7600 SIP-400 does not support the following QoS queueing actions using MQC:
  - Flow-based queueing
  - Adaptive shaping
- The Cisco 7600 SIP-400 does not support the following QoS policing feature:
  - Policing by Committed Information Rate (CIR) percentage (**police cir percent** command)—Supported as of Cisco IOS Release 12.2(18)SXF
- The Cisco 7600 SIP-400 does not support the following Frame Relay features:
  - Matching on DLCI is unsupported
  - Bridging encapsulation is unsupported
  - Multicast on multipoint interfaces is unsupported
  - FRF.5 is unsupported
  - FRF.8 is unsupported
  - FRF.12 fragmentation is unsupported
  - FRF.16 multilink support of four-octet extended addressing on an SVC is unsupported
  - NNI is unsupported
  - PVC bundling is unsupported
  - PPP over Frame Relay is unsupported
- The Cisco 7600 SIP-400 does not support RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5,* Multipoint Bridging (MPB). However, point-to-point bridging is supported.
- As of Cisco IOS Release 12.2(18)SXF, when using the Cisco 7600 SIP-400 with the 2-Port Gigabit Ethernet SPA or the 1-Port OC-48c/STM-16 ATM SPA, consider the following oversubscription guidelines:
  - The Cisco 7600 SIP-400 only supports installation of one 1-Port OC-48c/STM-16 ATM SPA without any other SPAs installed in the SIP.
  - The Cisco 7600 SIP-400 supports installation of up to two 2-Port Gigabit Ethernet SPAs without any other SPAs installed in the SIP.

  - The Cisco 7600 SIP-400 supports installation of any combination of OC-3 or OC-12 POS or ATM SPAs, up to a combined ingress bandwidth of OC-48 rates.
  - The Cisco 7600 SIP-400 supports installation of any combination of OC-3 or OC-12 POS or ATM SPAs up to a combined ingress bandwidth of OC-24 rates, when installed with a single 2-Port Gigabit Ethernet SPA.
- QinQ (the ability to map a single 802.1Q tag or a random double tag combination into a VPLS instance, a Layer 3 MPLS VPN, or an EoMPLS VC) is not supported.
- Cisco Discovery Protocol (CDP) is disabled by default on the 2-Port Gigabit Ethernet SPA interfaces and subinterfaces on the Cisco 7600 SIP-400.

Cisco 7600 SIP-600 Restrictions

Note Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

As of Cisco IOS Release 12.2(18)SXF, the Cisco 7600 SIP-600 has the following restrictions:

- The Cisco 7600 SIP-600 is not supported by the Supervisor Engine 32. The Cisco 7600 SIP-600 is supported by the Supervisor Engine 720 PFC3B and Supervisor Engine 720 PFC3BXL. It is not supported with a Supervisor Engine 720 PFC3A or in PFC3A mode.

  For more information about the requirements for Policy Feature Cards (PFCs) on the Catalyst 6500 Series switch, refer to the *Release Notes for Cisco IOS Release 12.2(33)SXH and Later Releases* at the following URL:
  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/release/notes/ol_14271.html#wp2561312
- The Cisco 7600 SIP-600 supports installation of only a single SPA in the first subslot.
- Removal of one type of SPA and reinsertion of a different type of SPA during OIR causes a reload of the Cisco 7600 SIP-600.
- QinQ (the ability to map a single 802.1Q tag or a random double tag combination into a VPLS instance, a Layer 3 MPLS VPN, or an EoMPLS VC) is not supported.
- H-VPLS with MPLS edge requires either an OSM module or Cisco 7600 SIP-600 in both the downlink (facing UPE) and uplink (MPLS core).
- Output policing is not supported.
- On any Cisco 7600 SIP-600 Ethernet port subinterface using VLANs, a unique VLAN ID must be assigned. This VLAN ID cannot be in use by any other interface on the Catalyst 6500 Series switch.

Note The Cisco 7600 SIP-600 should not be used in the same chassis with an IPsec VPN SPA.

Cisco 7600 SSC-400 Restrictions

As of Cisco IOS Release 12.2(18)SXE2, the Cisco 7600 SSC-400 has the following restrictions:

- The Cisco 7600 SSC-400 is only supported by the Supervisor Engine 720 (MSFC3 and PFC3).

For more information about the requirements for Policy Feature Cards (PFCs) on the Catalyst 6500 Series switch, refer to the For more information about the requirements for Policy Feature Cards (PFCs) on the Catalyst 6500 Series switch, refer to the *Release Notes for Cisco IOS Release 12.2(33)SXH and Later Releases* at the following URL:
http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/release/notes/ol_14271.html#wp2561312

- The Cisco 7600 SSC-400 only supports two IPsec VPN SPAs.

As of Cisco IOS Release 12.2(18)SXF, the Cisco 7600 SSC-400 has the following restrictions:

- The Cisco 7600 SSC-400 is not supported by the Supervisor Engine 32. The Cisco 7600 SSC-400 is only supported by the Supervisor Engine 720 (MSFC3 and PFC3).

  For more information about the requirements for Policy Feature Cards (PFCs) on the Catalyst 6500 Series switch, refer to the For more information about the requirements for Policy Feature Cards (PFCs) on the Catalyst 6500 Series switch, refer to the *Release Notes for Cisco IOS Release 12.2(33)SXH and Later Releases* at the following URL:
  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/release/notes/ol_14271.html#wp2561312

- The Cisco 7600 SSC-400 only supports two IPsec VPN SPAs.

Supported MIBs

The following MIBs are supported in Cisco IOS Release 12.2(18)SXE and later for the Cisco 7600 SIP-200 on a Catalyst 6500 Series switch:

- CISCO-ENTITY-ASSET-MIB
- CISCO-ENTITY-EXT-MIB
- CISCO-ENTITY-FRU-CONTROL-MIB
- ENTITY-MIB
- OLD-CISCO-CHASSIS-MIB

The following MIBs are supported in Cisco IOS Release 12.2(18)SXE and later for the Cisco 7600 SIP-400 on a Catalyst 6500 Series switch:

- ATM-ACCOUNTING-INFORMATION-MIB (RFC 2512)
- ATM-MIB (RFC 2515)
- ATM-SOFT-PVC-MIB
- ATM-TC-MIB
- ATM-TRACE-MIB
- CISCO-AAL5-MIB
- CISCO-ATM-CONN-MIB
- CISCO-ATM-RM-MIB
- CISCO-ATM TRAFFIC-MIB
- CISCO-CLASS-BASED-QOS-MIB
- CISCO-ENTITY-ASSET-MIB

- CISCO-ENTITY-EXT-MIB
- CISCO-ENTITY-FRU-CONTROL-MIB
- ENTITY-MIB
- IF-MIB
- OLD-CISCO-CHASSIS-MIB
- SONET MIB (RFC 2558)

The following MIBs are supported in Cisco IOS Release 12.2(18)SXF and later for the Cisco 7600 SIP-600 on a Catalyst 6500 Series switch:

- CISCO-ENTITY-ASSET-MIB
- CISCO-ENTITY-EXT-MIB
- CISCO-ENTITY-FRU-CONTROL-MIB
- ENTITY-MIB
- OLD-CISCO-CHASSIS-MIB

The following MIBs are supported in Cisco IOS Release 12.2(18)SXE2 and later for the Cisco 7600 SSC-400 on a Catalyst 6500 Series switch:

- CISCO-ENTITY-ASSET-MIB
- CISCO-ENTITY-EXT-MIB
- CISCO-ENTITY-FRU-CONTROL-MIB
- ENTITY-MIB
- OLD-CISCO-CHASSIS-MIB

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Displaying the SIP and SSC Hardware Type

To verify the SIP or SSC hardware type that is installed in your Catalyst 6500 Series switch, you can use the **show module** command. There are other commands on the Catalyst 6500 Series switch that also provide SIP hardware information, such as the **show idprom** command and **show diagbus** command.

Table 3-1 shows the hardware description that appears in the **show module** and **show idprom** command output for each type of SIP that is supported on the Catalyst 6500 Series switch.

Table 3-1 SIP Hardware Descriptions in show Commands

SIP	Description in show module and show idprom Commands
Cisco 7600 SIP-200	4-subslot SPA Interface Processor-200 / 7600-SIP-200
Cisco 7600 SIP-400	4-subslot SPA Interface Processor-400 / 7600-SIP-400
Cisco 7600 SIP-600	1-subslot SPA Interface Processor-600 / 7600-SIP-600
Cisco 7600 SSC-400	2-subslot Services SPA Carrier-400 / 7600-SSC-400

Example of the show module Command

The following example shows output from the **show module** command on the Catalyst 6500 Series switch with a Cisco 7600 SIP-400 installed in slot 13:

```
Router# show module 13
Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------

 13     0  4-subslot SPA Interface Processor-400  7600-SIP-400       JAB0851042X

Mod MAC addresses                       Hw     Fw           Sw           Status
--- ---------------------------------- ------ ------------ ------------ -------
 13  00e0.aabb.cc00 to 00e0.aabb.cc3f   0.525 12.2(PP_SPL_ 12.2(PP_SPL_ Ok

Mod Online Diag Status
--- -------------------
 13 Pass
```

Example of the show idprom Command

The following example shows sample output for a Cisco 7600 SIP-200 installed in slot 4 of the switch:

```
Router# show idprom module 4
IDPROM for module #4
  (FRU is '4-subslot SPA Interface Processor-200')
  OEM String = 'Cisco Systems'
  Product Number = '7600-SIP-200'
  Serial Number = 'SAD0738006Y'
  Manufacturing Assembly Number = '73-8272-03'
  Manufacturing Assembly Revision = '03'
  Hardware Revision = 0.333
  Current supplied (+) or consumed (-) = -4.77A
```

CHAPTER 4

Configuring the SIPs and SSC

This chapter provides information about configuring SIPs and SSCs on the Catalyst 6500 Series switch. It includes the following sections:

- Configuration Tasks, page 4-1
- Configuration Examples, page 4-61

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page -xlv.

Configuration Tasks

This section describes how to configure the SIPs and SSCs and includes information about verifying the configuration.

It includes the following topics:

- Required Configuration Tasks, page 4-2
- Identifying Slots and Subslots for SIPs, SSCs, and SPAs, page 4-2
- Configuring Compressed Real-Time Protocol, page 4-4
- Configuring Frame Relay Features, page 4-5
- Configuring Layer 2 Interworking Features on a SIP, page 4-17
- Configuring MPLS Features on a SIP, page 4-30
- Configuring QoS Features on a SIP, page 4-33
- Resetting a SIP, page 4-60

This section identifies those features that have SIP-specific configuration guidelines for you to consider and refers you to the supporting platform documentation.

Many of the Cisco IOS software features on the Catalyst 6500 Series switch that the FlexWAN and Enhanced FlexWAN modules support, the SIPs also support. Use this chapter while also referencing the list of supported features on the SIPs, in Chapter 3, "Overview of the SIPs and SSC."

Note When referring to the other platform documentation, be sure to note any SIP-specific configuration guidelines described in this document.

For information about configuring other features supported on the Catalyst 6500 Series switch but not discussed in this document, refer to the *Catalyst 6500 Series Cisco IOS Software Configuration Guide, 12.2SX* at the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/book.html

Required Configuration Tasks

As of Cisco IOS Release 12.2(18)SXE, there are no features that require direct configuration on the SIP or SSC. This means that you do not need to attach to the SIP or SSC itself to perform any configuration.

However, the Cisco 7600 SIP-200 and Cisco 7600 SIP-400 do implement and support certain features that are configurable at the system level on the Route Processor (RP).

Identifying Slots and Subslots for SIPs, SSCs, and SPAs

This section describes how to specify the physical locations of a SIP and SPA on the Catalyst 6500 Series switchs within the command-line interface (CLI) to configure or monitor those devices.

Note For simplicity, any reference to SIP in this section also applies to the SSC.

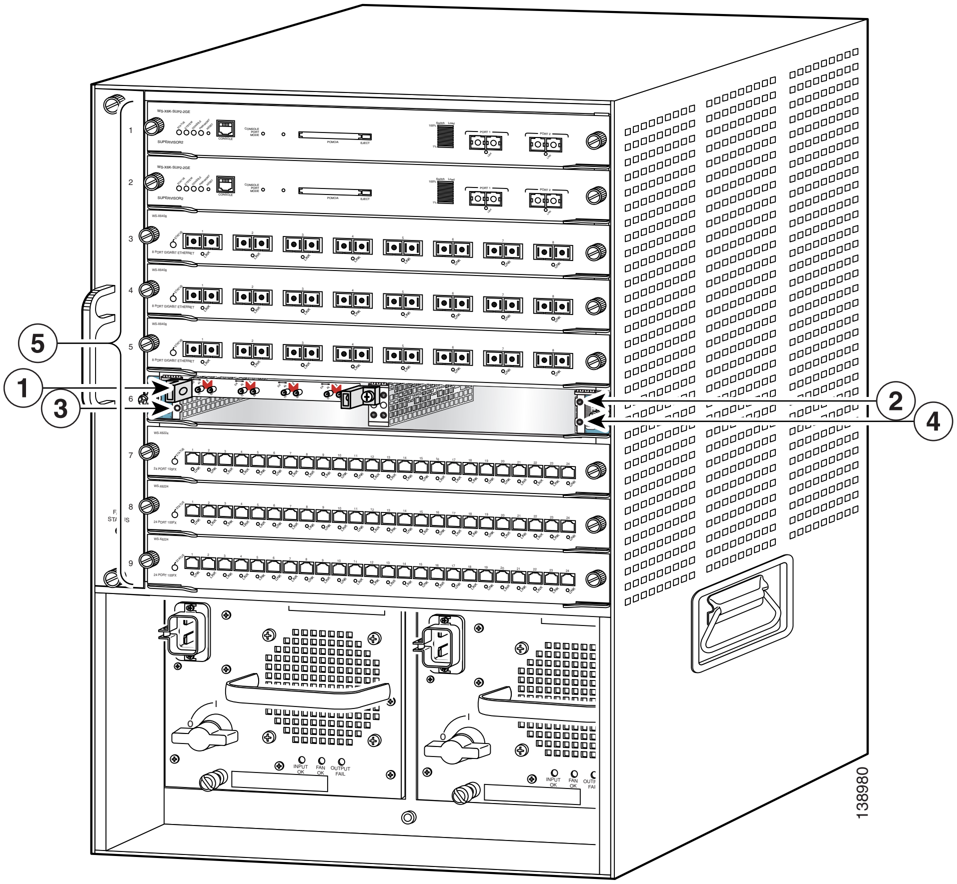
Specifying the Slot Location for a SIP or SSC

The Catalyst 6500 Series switch supports different chassis models, each of which supports a certain number of chassis slots.

Note The Catalyst 6500 Series switch SIPs are not supported with a Supervisor Engine 1, Supervisor Engine 1A, Supervisor Engine 2, or Supervisor Engine 720-3A.

Figure 4-1 shows an example of a SIP installed in slot 6 on a Catalyst 6509 switch. The Catalyst 6509 switch has nine horizontally-oriented chassis slots, which are numbered 1 to 9 from right to left.

Figure 4-1 SIP and SPA Installed in a Catalyst 6509 Switch



1	SIP subslot 0	**4**	SIP subslot 3
2	SIP subslot 1	**5**	Chassis slots 1–9 (numbered from top to bottom)
3	SIP subslot 2		

Some commands allow you to display information about the SIP itself, such as **show module**, **show sip-disk**, **show idprom module**, **show hw-module slot**, and **show diagbus**. These commands require you to specify the chassis slot location where the SIP that you want information about is installed.

For example, to display status and information about the SIP installed in slot 6 as shown in Figure 4-1, enter the following command:

```
Router# show module 6
```

For more information about SIP commands, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

Specifying the SIP or SSC Subslot Location for a SPA

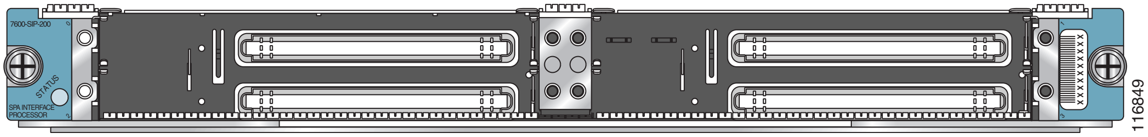
SIP subslots begin their numbering with 0 and have a horizontal or vertical orientation depending on the orientation of the SIP in the router chassis slot.

Figure 4-1 shows an example of a Cisco 7600 SIP-200 installed with a vertical orientation on a Cisco 7609 router. The Cisco 7600 SIP-200 supports four subslots for the installation of SPAs. In this example, the subslot locations are vertically oriented as follows:

- SIP subslot 0—Top–right subslot
- SIP subslot 1—Bottom–right subslot
- SIP subslot 2—Top–left subslot
- SIP subslot 3—Bottom–left subslot

Figure 4-2 shows the faceplate for the Cisco 7600 SIP-200 in a horizontal orientation.

Figure 4-2 Cisco 7600 SIP-200 Faceplate



In this view, the subslot locations in a horizontal orientation are as follows:

- SIP subslot 0—Top–left subslot
- SIP subslot 1—Top–right subslot
- SIP subslot 2—Bottom–left subslot
- SIP subslot 3—Bottom–right subslot

The SIP subslot numbering is indicated by a small numeric label beside the subslot on the faceplate.

As with the SIPs, some commands allow you to display information about the SPA itself, such as **show idprom module** and **show hw-module subslot**. These commands require you to specify both the physical location of the SIP and SPA in the format, *slot*/*subslot*, where:

- *slot*—Specifies the chassis slot number in the Catalyst 6500 Series switch where the SIP is installed.
- *subslot*—Specifies the secondary slot of the SIP where the SPA is installed.

For example, to display the operational status for the SPA installed in the first subslot of the SIP in chassis slot 6 shown in Figure 4-1, enter the following command:

```
Router# show hw-module subslot 6/0 oir
```

For more information about SPA commands, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

Configuring Compressed Real-Time Protocol

Compressed Real-Time Protocol (cRTP), from RFC 1889 (*RTP: A Transport Protocol for Real-Time Applications*), provides bandwidth efficiencies over low-speed links by compressing the UDP/RTP/IP header when transporting voice. With cRTP, the header for Voice over IP traffic can be reduced from 40 bytes to approximately 2 to 5 bytes offering substantial bandwidth efficiencies for low-speed links. cRTP is supported over Frame Relay, ATM, PPP, MLPPP, and HDLC encapsulated interfaces.

Note cRTP is supported only the Cisco 7600 SIP-200 with the 8-Port Channelized T1/E1 SPA, 2-Port and 4-Port Channelized T3 SPA, 2-Port and 4-Port Clear Channel T3/E3 SPA, and 1-Port Channelized OC-3/STM-1 SPA.

For information on configuring cRTP, see *Configuring Distributed Compressed Real-Time Protocol* at the following URL:

http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfdcrtp.html

Configuring Frame Relay Features

Many of the Frame Relay features supported on the FlexWAN and Enhanced FlexWAN modules on the Catalyst 6500 Series switch are also supported by the SIPs. For a list of the supported Frame Relay features on the SIPs, see Chapter 3, "Overview of the SIPs and SSC."

This section describes those Frame Relay features that have SIP-specific configuration guidelines. After you review the SIP-specific guidelines described in this document, then refer to the referenced URLs for more information about configuring Frame Relay features.

The Frame Relay features for SIPs and SPAs are qualified as *distributed features* because the processing for the feature is handled by the SIP or SPA, or a combination of both.

Configuring Distributed Multilink Frame Relay (FRF.16) on the Cisco 7600 SIP-200

The Distributed Multilink Frame Relay (dMLFR) feature provides a cost-effective way to increase bandwidth for particular applications by enabling multiple serial links to be aggregated into a single bundle of bandwidth. Multilink Frame Relay is supported on the User-Network Interface (UNI) and the Network-to-Network Interface (NNI) in Frame Relay networks.

Note Based on your link configuration, dMLFR can be either software-based on the Cisco 7600 SIP-200, or hardware-based on the 8-Port Channelized T1/E1 SPA, 2-Port and 4-Port Channelized T3 SPAs, and 1-Port Channelized OC-3/STM-1 SPA. For more information about the hardware-based configuration, see also refer to Chapter 15, "Configuring the 8-Port Channelized T1/E1 SPA," Chapter 17, "Configuring the 2-Port and 4-Port Channelized T3 SPAs," and Chapter 18, "Configuring the 1-Port Channelized OC-3/STM-1 SPA."

Table 4-1 provides information about where the dMLFR feature for SPA interfaces is supported.

Table 4-1 dMLFR Feature Compatibility by SIP and SPA Combination

Feature	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600
Hardware-based dMLFR	In Cisco IOS Release 12.2(18)SXE and later: • 8-Port Channelized T1/E1 SPA • 2-Port and 4-Port Channelized T3 SPA	Not supported.	Not supported.
Hardware- and software-based dMLFR	In Cisco IOS Release 12.2(33)SXH and later: • 8-Port Channelized T1/E1 SPA • 2-Port and 4-Port Channelized T3 SPA • 1-Port Channelized OC-3/STM-1 SPA	Not supported.	Not supported.

This section includes the following topics:

- Overview of dMLFR, page 4-6
- dMLFR Configuration Guidelines, page 4-7
- dMLFR Configuration Tasks, page 4-8
- Verifying dMLFR, page 4-10

Overview of dMLFR

The Distributed Multilink Frame Relay (dMLFR) feature enables you to create a virtual interface called a *bundle* or *bundle interface*. The bundle interface emulates a physical interface for the transport of frames. The Frame Relay data link runs on the bundle interface, and Frame Relay virtual circuits are built upon it.

The bundle is made up of multiple serial links, called *bundle links*. Each bundle link within a bundle corresponds to a physical interface. Bundle links are invisible to the Frame Relay data-link layer, so Frame Relay functionality cannot be configured on these interfaces. Regular Frame Relay functionality that you want to apply to these links must be configured on the bundle interface. Bundle links are visible to peer devices. The local switch and peer devices exchange link integrity protocol control messages to determine which bundle links are operational and to synchronize which bundle links should be associated with which bundles.

For link management, each end of a bundle link follows the dMLFR link integrity protocol and exchanges link control messages with its peer (the other end of the bundle link). To bring up a bundle link, both ends of the link must complete an exchange of ADD_LINK and ADD_LINK_ACK messages. To maintain the link, both ends periodically exchange HELLO and HELLO_ACK messages. This exchange of hello messages and acknowledgments serve as a keepalive mechanism for the link. If a switch is sending hello messages but not receiving acknowledgments, it will resend the hello message up to a configured maximum number of times. If the switch exhausts the maximum number of retries, the bundle link line protocol is considered down (unoperational).

The bundle link interface’s line protocol status is considered up (operational) when the peer device acknowledges that it will use the same link for the bundle. The line protocol remains up when the peer device acknowledges the hello messages from the local switch.

The bundle interface’s line status becomes up when at least one bundle link has its line protocol status up. The bundle interface's line status goes down when the last bundle link is no longer in the up state. This behavior complies with the Class A bandwidth requirement defined in FRF.16.

The bundle interface’s line protocol status is considered up when the Frame Relay data-link layer at the local switch and peer device synchronize using the Local Management Interface (LMI), when LMI is enabled. The bundle line protocol remains up as long as the LMI keepalives are successful.

dMLFR Configuration Guidelines

To support dMLFR on the Cisco 7600 SIP-200, consider the following guidelines:

- dMLFR must be configured on the peer device.
- The dMLFR peer device must not send frames that require assembly.
- The Cisco 7600 SIP-200 supports distributed links under the following conditions:
  - All links are on the same Cisco 7600 SIP-200.
  - T1 and E1 links cannot be mixed in a bundle.
  - T1 or E1 links in a bundle are recommended to have the same bandwidth.
- QoS is implemented on the Cisco 7600 SIP-200 for dMLFR.
- dMLFR is supported in software by the Cisco 7600 SIP-200, or in hardware on the 2-Port and 4-Port Channelized T3 SPA, the 8-Port Channelized T1/E1 SPA, and the 1-Port Channelized OC-3/STM-1 SPA. This support is determined by your link configuration.

Software-Based Guidelines

dMLFR will be implemented in the software if *any* of the following conditions are met:

- Any one bundle link member is a fractional T1 or E1 link.
- There are more than 12 T1 or E1 links in a bundle.
- Bundle links are configured across SPAs, but all links are on the same *type* of SPA. For example, links on a 8-Port Channelized T1/E1 SPA cannot be distributed with links on a 2-Port and 4-Port Channelized T3 SPA.

Hardware-Based Guidelines

dMLFR will be implemented in the hardware when *all* of the following conditions are met:

- All bundle link members are T1 or E1 only.
- All bundle links are on the same SPA.
- There are no more than 12 links in a bundle.

dMLFR Restrictions

When configuring dMLFR on the Cisco 7600 SIP-200, consider the following restrictions:

- FRF.9 hardware compression is not supported.
- Software compression is not supported.
- Encryption is not supported.
- The maximum differential delay supported is 50 ms when supported in hardware, and 100 ms when supported in software.
- Fragmentation is not supported on the transmit side.

- Frame Relay fragmentation (FRF.12) is not supported.

dMLFR Configuration Tasks

The following sections describe how to configure dMLFR:

- Enabling Distributed CEF Switching, page 4-8 (required)
- Creating a Multilink Frame Relay Bundle, page 4-8 (required)
- Assigning an Interface to a dMLFR Bundle, page 4-9 (required)

Enabling Distributed CEF Switching

To enable dMLFR, you must first enable distributed CEF (dCEF) switching. Distributed CEF switching is enabled by default on the Catalyst 6500 Series switch.

To enable dCEF, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip cef distributed`	Enables dCEF switching.

Creating a Multilink Frame Relay Bundle

To configure the bundle interface for dMLFR, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface mfr number`	Configures a multilink Frame Relay bundle interface and enters interface configuration mode, where: • *number*—Specifies the number for the Frame Relay bundle.
Step 2	`Router(config-if)# frame-relay multilink bid name`	(Optional) Assigns a bundle identification name to a multilink Frame Relay bundle, where: • *name*—Specifies the name for the Frame Relay bundle. **Note** The bundle identification (BID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shutdown** and **no shutdown** commands in interface configuration mode.
Step 3	`Router(config-if)# frame-relay intf-type dce`	Configures the switch to function as a digital communications equipment (DCE) device, or as a switch.

Assigning an Interface to a dMLFR Bundle

Note If you use this task to assign more than 12 T1 or E1 interface links as part of the same bundle, or if any of the T1/E1 interface links are fractional T1/E1, or any links reside on multiple SPAs as part of the same bundle, then software-based MLFR is implemented automatically by the Cisco 7600 SIP-200.

To configure an interface link and associate it as a member of a dMLFR bundle, perform this task beginning in global configuration mode. Repeat these steps to assign multiple links to the dMLFR bundle.

	Command	Purpose
Step 1	**2-Port and 4-Port Channelized T3 SPA** Router(config)# **interface serial** *slot/subslot/port/t1-number:channel-group* **8-Port Channelized T1/E1 SPA** Router(config)# **interface serial** *slot/subslot/port:channel-group*	Specifies a serial interface and enters interface configuration mode, where: • *slot*—Specifies the chassis slot number where the SIP is installed. • *subslot*—Specifies the secondary slot number on a SIP where a SPA is installed. • *port*—Specifies the number of the interface port on the SPA. • *t1-number*—Specifies the logical T1 number in channelized mode. • *channel-group*—Specifies the logical channel group assigned to the time slots within the T1/E1 group. **Note** If you configure a fractional T1/E1 interface on the SPA using a channel group and specify that fractional T1/E1 channel group as part of this task, then software-based dMLFR is implemented automatically by the Cisco 7600 SIP-200 when you assign the interface to the dMLFR bundle.
Step 2	Router(config-if)# **encapsulation frame-relay mfr** *number* [*name*]	Creates a multilink Frame Relay bundle link and associates the link with a bundle, where: • *number*—Specifies the number for the Frame Relay bundle. This number should match the dMLFR interface number specified in the **interface mfr** command. • *name*—(Optional) Specifies the name for the Frame Relay bundle.

	Command	Purpose
Step 3	`Router(config-if)# frame-relay multilink lid name`	(Optional) Assigns a bundle link identification name with a multilink Frame Relay bundle link, where: • *name*—Specifies the name for the Frame Relay bundle. **Note** The bundle link identification (LID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shutdown** and **no shutdown** commands in interface configuration mode.
Step 4	`Router(config-if)# frame-relay multilink hello seconds`	(Optional) Configures the interval at which a bundle link will send out hello messages, where: • *seconds*—Specifies the number of seconds between hello messages sent out over the multilink bundle. The default is 10 seconds.
Step 5	`Router(config-if)# frame-relay multilink ack seconds`	(Optional) Configures the number of seconds that a bundle link will wait for a hello message acknowledgment before resending the hello message, where: • *seconds*—Specifies the number of seconds a bundle link will wait for a hello message acknowledgment before resending the hello message. The default is 4 seconds.
Step 6	`Router(config-if)# frame-relay multilink retry number`	(Optional) Configures the maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment, where: • *number*—Specifies the maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment. The default is 2 tries.

Verifying dMLFR

To verify dMLFR configuration, use the **show frame-relay multilink** command. If you use the **show frame-relay multilink** command without any options, information for all bundles and bundle links is displayed.

The following examples show output for the **show frame-relay multilink** command with the **serial** *number* and **detailed** options. Detailed information about the specified bundle links is displayed.

```
Router# show frame-relay multilink serial6 detailed

Bundle: MFR49, State = down, class = A, fragmentation disabled
 BID = MFR49
 No. of bundle links = 1, Peer's bundle-id =
 Bundle links:

  Serial6/0/0:0, HW state = up, link state = Add_sent, LID = test
    Cause code = none, Ack timer = 4, Hello timer = 10,
    Max retry count = 2, Current count = 0,
    Peer LID = , RTT = 0 ms
    Statistics:
    Add_link sent = 21, Add_link rcv'd = 0,
    Add_link ack sent = 0, Add_link ack rcv'd = 0,
    Add_link rej sent = 0, Add_link rej rcv'd = 0,
    Remove_link sent = 0, Remove_link rcv'd = 0,
    Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,
    Hello sent = 0, Hello rcv'd = 0,
    Hello_ack sent = 0, Hello_ack rcv'd = 0,
    outgoing pak dropped = 0, incoming pak dropped = 0
```

Configuring Distributed Multilink PPP on the Cisco 7600 SIP-200

The Distributed Multilink Point-to-Point Protocol (dMLPPP) feature allows you to combine T1/E1 lines into a bundle that has the combined bandwidth of multiple T1/E1 lines. This is done by using a dMLPPP link. You choose the number of bundles and the number of T1/E1 lines in each bundle. This allows you to increase the bandwidth of your network links beyond that of a single T1/E1 line without having to purchase a T3 line.

Note Based on your link configuration, dMLPPP can be either software-based on the Cisco 7600 SIP-200, or hardware-based on the 8-Port Channelized T1/E1 SPA and 2-Port and 4-Port Channelized T3 SPAs. For more information about the hardware-based configuration, see also refer to Chapter 15, "Configuring the 8-Port Channelized T1/E1 SPA," Chapter 17, "Configuring the 2-Port and 4-Port Channelized T3 SPAs," and Chapter 18, "Configuring the 1-Port Channelized OC-3/STM-1 SPA."

This section includes the following topics:

- dMLPPP Configuration Guidelines, page 4-11
- dMLPPP Configuration Tasks, page 4-12
- Verifying MLPPP, page 4-15

dMLPPP Configuration Guidelines

dMLPPP is supported in software by the Cisco 7600 SIP-200, or in hardware on the 2-Port and 4-Port Channelized T3 SPA, the 8-Port Channelized T1/E1 SPA, and the 1-Port Channelized OC-3/STM-1 SPA. This support is determined by your link configuration.

The Cisco 7600 SIP-200 supports distributed links under the following conditions:

- All links are on the same Cisco 7600 SIP-200.
- T1 and E1 links cannot be mixed in a bundle.
- T1 or E1 links in a bundle are recommended to have the same bandwidth.
- QoS is implemented on the Cisco 7600 SIP-200 for dMLPPP.

Software-Based Guidelines

dMLPPP will be implemented in the software if *any* of the following conditions are met:

- Any one bundle link member is a fractional T1 or E1 link.
- There are more than 12 T1 or E1 links in a bundle.
- Bundle links are configured across SPAs.
- To enable fragmentation for software-based dMLPPP, you must configure the **ppp multilink interleave** command. This command is not required to enable fragmentation for hardware-based dMLPPP.

Hardware-Based Guidelines

dMLPPP will be implemented in the hardware when all of the following conditions are met:

- All bundle link members are T1 or E1 only.
- All bundle links are on the same SPA.
- There are no more than 12 links in a bundle.

dMLPPP Restrictions

When configuring dMLPPP on the Cisco 7600 SIP-200, consider the following restrictions:

- dMLPPP across SPAs is not supported.
- Hardware and software compression is not supported.
- Encryption is not supported.
- The maximum differential delay supported is 50 ms when supported in hardware, and 100 ms when supported in software.

dMLPPP Configuration Tasks

The following sections describe how to configure MLPPP:

- Enabling Distributed CEF Switching, page 4-12 (required)
- Creating a dMLPPP Bundle, page 4-13 (required)
- Assigning an Interface to a dMLPPP Bundle, page 4-13 (required)
- Configuring Link Fragmentation and Interleaving over dMLPPP, page 4-14 (optional)

Enabling Distributed CEF Switching

To enable dMLPPP, you must first enable distributed CEF switching. Distributed CEF switching is enabled by default on the Cisco 7600 series router.

To enable dCEF, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip cef distributed`	Enables distributed CEF switching.

Creating a dMLPPP Bundle

To configure a dMLPPP bundle, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface multilink group-number`	Creates a multilink interface and enters interface configuration mode, where: • *group-number*—Specifies the group number for the multilink bundle.
Step 2	`Router(config-if)# ip address ip-address mask`	Sets the IP address for the multilink group, where: • *ip-address*—Specifies the IP address for the interface. • *mask*—Specifies the mask for the associated IP subnet.
Step 3	`Router(config-if)# ppp multilink interleave`	(Optional—Software-based LFI) Enables fragmentation for the interfaces assigned to the multilink bundle. Fragmentation is disabled by default in software-based LFI.
Step 4	`Router(config-if)#  ppp multilink fragment-delay delay`	(Optional) Sets the fragmentation size satisfying the configured delay on the multilink bundle, where: • *delay*—Specifies the delay in milliseconds.

Assigning an Interface to a dMLPPP Bundle

Note If you use this task to assign more than 12 T1 or E1 interface links as part of the same bundle, or if any of the T1/E1 interface links are fractional T1/E1, or any links reside on multiple SPAs as part of the same bundle, then software-based dMLPPP is implemented automatically by the Cisco 7600 SIP-200.

To configure an interface PPP link and associate it as a member of a multilink bundle, perform this task beginning in global configuration mode. Repeat these steps to assign multiple links to the dMLPPP bundle.

	Command	Purpose

Step 1	**2-Port and 4-Port Channelized T3 SPA**
	`Router(config)# `**`interface serial`** *`slot/subslot/port/t1-number:channel-group`*
	8-Port Channelized T1/E1 SPA
	`Router(config)# `**`interface serial`** *`slot/subslot/port:channel-group`*
		Specifies a serial interface and enters interface configuration mode, where:
		• *slot*—Specifies the chassis slot number where the SIP is installed.
		• *subslot*—Specifies the secondary slot number on a SIP where a SPA is installed.
		• *port*—Specifies the number of the interface port on the SPA.
		• *t1-number*—Specifies the logical T1 number in channelized mode.
		• *channel-group*—Specifies the logical channel group assigned to the time slots within the T1 or E1 group.
		Note If you configure a fractional T1/E1 interface on the SPA using a channel group and specify that fractional T1/E1 channel group as part of this task, then software-based MLPPP is implemented automatically by the Cisco 7600 SIP-200 when you assign the interface to the MLPPP bundle.

Step 2	`Router(config-if)# `**`encapsulation ppp`**	Enables PPP encapsulation.
Step 3	`Router(config-if)# `**`ppp multilink`**	(Optional) Enables MLPPP on the interface.
Step 4	`Router(config-if)# `**`multilink-group`** *`group-number`*	Assigns the interface to a multilink bundle, where: • *group-number*—Specifies the group number for the multilink bundle. This number should match the MLPPP interface number specified in the **interface multilink** command.
Step 5	`Router(config-if)# `**`ppp authentication chap`**	(Optional) Enables Challenge Handshake Authentication Protocol (CHAP) authentication.

Configuring Link Fragmentation and Interleaving over dMLPPP

Link fragmentation and interleaving (LFI) over dMLPPP is supported in software on the Cisco 7600 SIP-200, or in hardware on the 2-Port and 4-Port Channelized T3 SPA and the 8-Port Channelized T1/E1 SPA. This support is determined by your link configuration.

Software-Based Guidelines

When configuring LFI over dMLPPP, consider the following guidelines for software-based LFI:

- LFI over dMLPPP will be configured in software if there is more than one link assigned to the dMLPPP bundle.
- LFI is disabled by default in software-based LFI. To enable LFI on the multilink interface, use the **ppp multilink interleave** command.
- Fragmentation size is calculated from the delay configured and the member link bandwidth.

- You must configure a policy map with a priority class under the multilink interface.

Hardware-Based Guidelines

When configuring LFI over dMLPPP, consider the following guidelines for hardware-based LFI:

- LFI over dMLPPP will configured in hardware if you only assign one link (either T1/E1 or fractional T1/E1) to the MLPPP bundle.
- LFI is enabled by default in hardware-based LFI with a default size of 512 bytes. To enable LFI on the serial interface, use the **ppp multilink interleave** command.
- A policy-map having a priority class needs to be applied to the multilink interface.

Verifying MLPPP

To verify dMLPPP configuration, use the **show ppp multilink** command, as shown in the following example:

```
Router# show ppp multilink

Multilink2, bundle name is group2
  Bundle up for 00:01:21
  Bundle is Distributed
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received, 1/255 load
  0x0 received sequence, 0x0 sent sequence
  Member links: 2 active, 0 inactive (max not set, min not set)
    Se4/3/0/1:0, since 00:01:21, no frags rcvd
    Se4/3/0/1:1, since 00:01:19, no frags rcvd
```

If hardware-based MLPPP is configured on the SPA, the **show ppp multilink** command displays “Multilink in Hardware” as shown in the following example:

```
Router# show ppp multilink

Multilink1, bundle name is group1
  Bundle up for 00:00:13
  Bundle is Distributed
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received, 206/255 load
  0x0 received sequence, 0x0 sent sequence
  Member links: 2 active, 0 inactive (max not set, min not set)
    Se4/2/0/1:0, since 00:00:13, no frags rcvd
    Se4/2/0/2:0, since 00:00:10, no frags rcvd
  Distributed fragmentation on. Fragment size 512.  Multilink in Hardware.
```

Configuring Distributed Link Fragmentation and Interleaving for Frame Relay and ATM Interfaces

The Distributed Link Fragmentation and Interleaving (dLFI) feature supports the transport of real-time traffic, such as voice, and non-real-time traffic, such as data, on lower-speed Frame Relay and ATM virtual circuits (VCs) and on leased lines without causing excessive delay to the real-time traffic.

This feature is implemented using dMLPPP over Frame Relay, ATM, and leased lines. The feature enables delay-sensitive real-time packets and non-real-time packets to share the same link by fragmenting the large data packets into a sequence of smaller data packets (fragments). The fragments are then interleaved with the real-time packets. On the receiving side of the link, the fragments are reassembled and the packets reconstructed.

The dLFI feature is often useful in networks that send real-time traffic using Distributed Low Latency Queueing, such as voice, but have bandwidth problems that delay this real-time traffic due to the transport of large, less time-sensitive data packets. The dLFI feature can be used in these networks to disassemble the large data packets into multiple segments. The real-time traffic packets then can be sent between these segments of the data packets. In this scenario, the real-time traffic does not experience a lengthy delay waiting for the low-priority data packets to traverse the network. The data packets are reassembled at the receiving side of the link, so the data is delivered intact.

The ability to configure quality of service (QoS) using the modular QoS CLI while also using dMLPPP is also introduced as part of the dLFI feature.

For specific information about configuring dLFI, refer to the *FlexWAN and Enhanced FlexWAN Module Installation and Configuration Note* located at the following URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexwan-config-guide.html

For information about configuring dLFI on ATM SPAs, see the "Configuring Link Fragmentation and Interleaving with Virtual Templates" section on page 7-45 in Chapter 7, "Configuring the ATM SPAs."

Table 4-2 provides information about where the dLFI feature for SPA interfaces is supported.

Table 4-2 dLFI Feature Compatibility by SIP and SPA Combination

Feature	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600
Hardware-based dLFI	In Cisco IOS Release 12.2(18)SXE and later: • 8-Port Channelized T1/E1 SPA • 2-Port and 4-Port Channelized T3 SPA	Not supported.	Not supported.
Hardware- and software-based dLFI	In Cisco IOS Release 12.2(33)SXH and later: • 8-Port Channelized T1/E1 SPA • 2-Port and 4-Port Channelized T3 SPA • 1-Port Channelized OC-3/STM-1 SPA	Not supported.	Not supported.
dLFI with MPLS	Not supported.	Not supported.	Not supported.

Catalyst 6500 Series Switch LFI Restrictions

When configuring LFI on the Catalyst 6500 Series switch, consider the following restrictions:

- A maximum number of 200 PVCs or SVCs using Link Fragmentation and Interleaving (LFI) is supported for all ATM SPAs (or other ATM modules) in a Catalyst 6500 Series switch.
- LFI using FRF.12 is supported in hardware only for the 2-Port and 4-Port Channelized T3 SPA and 8-Port Channelized T1/E1 SPA.
- LFI over dMLPPP is supported in software or hardware depending on your link configuration. For more information about software-based LFI over MLPPP, see the "Configuring Link Fragmentation and Interleaving over dMLPPP" section on page 4-14. For more information about hardware-based LFI over dMLPPP, refer to Chapter 15, "Configuring the 8-Port Channelized T1/E1 SPA," Chapter 17, "Configuring the 2-Port and 4-Port Channelized T3 SPAs," and Chapter 18, "Configuring the 1-Port Channelized OC-3/STM-1 SPA."
- QoS is implemented on the Cisco 7600 SIP-200 for dLFI.

Configuring Voice over Frame Relay FRF.11 and FRF.12

Voice over Frame Relay (VoFR) enables a switch to carry voice traffic (for example, telephone calls and faxes) over a Frame Relay network using the FRF.11 protocol. This specification defines multiplexed data, voice, fax, dual-tone multi-frequency (DTMF) digit-relay, and channel-associated signaling (CAS)/robbed-bit signaling frame formats. The Frame Relay backbone must be configured to include the map class and Local Management Interface (LMI).

The Cisco VoFR implementation enables dynamic- and tandem-switched calls and Cisco trunk calls. Dynamic-switched calls have dial-plan information included that processes and routes calls based on the telephone numbers. The dial-plan information is contained within dial-peer entries.

Note Because the Catalyst 6500 Series switch does not support voice modules, the Catalyst 6500 Series switch can act only as a VoFR tandem switch when FRF.11 or FRF.12 is configured on the SIPs.

Tandem-switched calls are switched from incoming VoFR to an outgoing VoFR-enabled data-link connection identifier (DLCI) and tandem nodes enable the process. The nodes also switch Cisco trunk calls.

Permanent calls are processed over Cisco private-line trunks and static FRF.11 trunks that specify the frame format and coder types for voice traffic over a Frame Relay network.

VoFR connections depend on the hardware platform and type of call. The types of calls are:

- Switched (user dialed or auto-ringdown and tandem)
- Permanent (Cisco trunk or static FRF.11 trunk)

For specific information about configuring voice over Frame Relay FRF.11 and FRF.12, refer to the *Cisco IOS Voice, Video, and Fax Configuration Guide* located at the following URL:

http://www.cisco.com/en/US/docs/ios/12_2/voice/configuration/guide/vvfvofr.html

Configuring Layer 2 Interworking Features on a SIP

This section provides SIP-specific information about configuring the Layer 2 interworking features on the Catalyst 6500 Series switch. It includes the following topics:

- Configuring Multipoint Bridging, page 4-17
- Configuring PPP Bridging Control Protocol Support, page 4-18
- Configuring Virtual Private LAN Service (VPLS), page 4-23
- Configuring Asymmetric Carrier Delay, page 4-28

Configuring Multipoint Bridging

Note As of Cisco IOS Release 12.2(18)SXE, MPB is supported on the Catalyst 6500 Series switch with the 2-Port and 4-Port OC-3c/STM-1 ATM SPA and the Cisco 7600 SIP-200, and the serial SPAs with the Cisco 7600 SIP-200, including the 2-Port and 4-Port Clear Channel T3/E3 SPA, 2-Port and 4-Port Channelized T3 SPA, the 8-Port Channelized T1/E1 SPA, and the 1-Port Channelized OC-3/STM-1 SPA.

Multipoint bridging (MPB) enables point-to-multipoint bridging for ATM permanent virtual circuits (PVCs) and Frame Relay data-link connection identifiers (DLCIs). This feature allows the use of multiple VCs or DLCIs per VLAN for bridging on the supported WAN line cards. Multipoint bridging allows service providers to add support for Ethernet-based Layer 2 services to the proven technology of their existing ATM and Frame Relay legacy networks. Customers can then use their current VLAN-based networks over the ATM or Frame Relay cloud. This also allows service providers to gradually update their core networks to the latest Gigabit Ethernet optical technologies, while still supporting their existing customer base.

ATM interfaces use RFC 1483 bridging, and Frame Relay interfaces use RFC 1490 bridging, both of which provide an encapsulation method to allow the transport of Ethernet frames over each type of Layer 2 network.

Note RFC 1483 has been obsoleted and superseded by RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. RFC 1490 has been obsoleted and superseded by RFC 2427, *Multiprotocol Interconnect over Frame Relay.* To avoid confusion, this document continues to use the original RFC numbers.

For specific information about configuring MPB, refer to the *FlexWAN and Enhanced FlexWAN Module Installation and Configuration Note* located at the following URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexwan-config-guide.html

Configuring PPP Bridging Control Protocol Support

The Bridging Control Protocol (BCP) feature on the SIPs and SPAs enables forwarding of Ethernet frames over serial and SONET networks, and provides a high-speed extension of enterprise LAN backbone traffic through a metropolitan area. The implementation of BCP on the SPAs includes support for IEEE 802.1D Spanning Tree Protocol, IEEE 802.1Q Virtual LAN (VLAN), and high-speed switched LANs.

The Bridging Control Protocol (BCP) feature provides support for BCP to Cisco devices, as described in RFC 3518, *Point-to-Point Protocol (PPP) Bridging Control Protocol (BCP)*. The Cisco implementation of BCP is a VLAN infrastructure that does not require the use of subinterfaces to group Ethernet 802.1Q trunks and the corresponding PPP links. This approach enables users to process VLAN encapsulated packets without having to configure subinterfaces for every possible VLAN configuration.

In Cisco IOS Release 12.2(33)SXH and later releases, BCP is supported over dMLPPP links on the Cisco 7600 SIP-200 with the 2-Port and 4-Port Channelized T3 SPA and the 8-Port Channelized T1/E1 SPA. BCP over dMLPPP is supported in trunk mode (**switchport**) only, in which a single BCP link can carry multiple VLANs.

BCP Configuration Guidelines

When configuring BCP support for SPAs on the Cisco 7600 SIP-200, consider the following guidelines:

- In Cisco IOS Release 12.2(33)SXH and later releases, QoS is supported on bridged interfaces. In earlier releases, QoS is not supported on bridged interfaces.

Configuring BCP in Trunk Mode

When BCP is configured in trunk mode, a single BCP link can carry multiple VLANs. BCP trunk mode operation is consistent with that of normal Ethernet trunk ports.

Trunk Mode BCP Configuration Guidelines

When configuring BCP support in trunk mode for SPAs on the Cisco 7600 SIP-200, consider the following guidelines:

- There are some differences between the Ethernet trunk ports and BCP trunk ports.
  - Ethernet trunk ports support ISL and 802.1Q encapsulation, but BCP trunk ports support only 802.1Q.
  - Ethernet trunk ports support Dynamic Trunk Protocol (DTP), which is used to automatically determine the trunking status of the link. BCP trunk ports are always in trunk state and no DTP negotiation is performed.
  - The default behavior of Ethernet trunk ports is to allow all VLANs on the trunk. The default behavior of BCP trunks is to disallow all VLANs. This means that VLANs that need to be allowed have to be explicitly configured on the BCP trunk port.
- Use the **switchport** command under the WAN interface when configuring trunk mode BCP.
- The SIPs support the following maximum number of BCP ports on any given VLAN:
  - In Cisco IOS Release 12.2(18)SXE and later—Maximum of 60 BCP ports
  - In Cisco IOS Release 12.2(33)SXH and later—Maximum of 112 BCP ports on Cisco 7600 SIP-200.
- To use VLANs in trunk mode BCP, you must use the **vlan** command to manually add the VLANs to the VLAN database. The default behavior for trunk mode BCP allows no VLANs.
- Trunk mode BCP is not supported on VLAN IDs 0, 1006–1023, and 1025.
- The native VLAN (1) has the following restrictions for trunk mode BCP:
  - In Cisco IOS 12.2SX software releases—The native VLAN is not supported.
  - In Cisco IOS Release 12.2(33)SXH and later releases—The native VLAN is supported.
- For trunk mode BCP (switch port), STP interoperability is the same as that of Ethernet switch ports. The STP path cost of WAN links can be changed and other STP functionality such as BPDU Guard and PortFast will work on the WAN links. However, we recommend that you do not change the default values.
- VLAN Trunking Protocol (VTP) is supported.

Note The management VLAN, VLAN 1, must be explicitly enabled on the trunk to send VTP advertisements.

To configure BCP in trunk mode, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vlan dot1q tag native`	(Optional) Enables dot1q tagging for all VLANs in a trunk. By default, packets on the native VLAN are sent untagged. When you enable dot1q tagging, packets are tagged with the native VLAN ID.

Step 2	**2-Port and 4-Port Channelized T3 SPA**
	`Router(config)# interface serial slot/subslot/port/t1-number:channel-group`
	8-Port Channelized T1/E1 SPA
	`Router(config)# interface serial slot/subslot/port:channel-group`
		Specifies an interface and enters interface configuration mode, where:
		• *slot*—Specifies the chassis slot number where the SIP is installed.
		• *subslot*—Specifies the secondary slot number on a SIP where a SPA is installed.
		• *port*—Specifies the number of the interface port on the SPA.
		• *t1-number*—Specifies the logical T1 number in channelized mode.
		• *channel-group*—Specifies the logical channel group assigned to the time slots within the T1 or E1 group.

Step 3	`Router(config-if)# switchport`	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration. PPP encapsulation is automatically configured, and the interface is automatically configured for trunk mode and nonegotiate status.
Step 4	`Router(config-if)# shutdown`	Disables the interface.

	Command	Purpose
Step 5	`Router(config-if)# no shutdown`	Restarts the disabled interface.
Step 6	`Router(config-if)# switchport trunk allowed vlan {all | {add | remove | except} vlan-list [,vlan-list...] | vlan-list [,vlan-list...]}`	(Optional) Controls which VLANs can receive and transmit traffic on the trunk, where: • **all**—Enables all applicable VLANs. • **add** *vlan-list* [,*vlan-list*...]—Appends the specified list of VLANs to those currently set instead of replacing the list. • **remove** *vlan-list* [,*vlan-list*...]—Removes the specified list of VLANs from those currently set instead of replacing the list. • **except** *vlan-list* [,*vlan-list*...]—Excludes the specified list of VLANs from those currently set instead of replacing the list. • *vlan-list* [,*vlan-list*...]—Specifies a single VLAN number from 1 to 4094, or a continuous range of VLANs that are described by two VLAN numbers from 1 to 4094. You can specify multiple VLAN numbers or ranges using a comma-separated list. To specify a range of VLANs, enter the smaller VLAN number first, separated by a hyphen and the larger VLAN number at the end of the range. **Note** Do not enable the reserved VLAN range (1006 to 1024) on trunks when connecting a Cisco 7600 series router running the Cisco IOS software on both the supervisor engine and the MSFC to a Cisco 7600 series router running the Catalyst operating system. These VLANs are reserved in Cisco 7600 series routers running the Catalyst operating system. If enabled, Cisco 7600 series routers running the Catalyst operating system may error-disable the ports if there is a trunking channel between these systems.

Verifying BCP in Trunk Mode

Because the PPP link has to flap (be brought down and renegotiated), it is important that you run the following **show** commands after you configure BCP in trunk mode to confirm the configuration:

Command	Purpose

2-Port and 4-Port Channelized T3 SPA
`Router# show interfaces [serial slot/subslot/port/t1-number:channel-group] trunk [module number]`
8-Port Channelized T1/E1 SPA
`Router# show interfaces [serial slot/subslot/port:channel-group] trunk [module number]`
	Displays the interface trunk information, where:
	• *slot*—Specifies the chassis slot number where the SIP is installed.
	• *subslot*—Specifies the secondary slot number on a SIP where a SPA is installed.
	• *port*—Specifies the number of the interface port on the SPA.
	• *t1-number*—Specifies the logical T1 number in channelized mode.
	• *channel-group*—Specifies the logical channel group assigned to the time slots within the T1 or E1 group.
	• **module** *number*—(Optional) Specifies the chassis slot number of the SIP and displays information for all interfaces of the SPAs in that SIP.

Command	Purpose

2-Port and 4-Port Channelized T3 SPA
`Router# show interfaces [serial slot/subslot/port/t1-number:channel-group] switchport [module number]`
8-Port Channelized T1/E1 SPA
`Router# show interfaces [serial slot/subslot/port:channel-group] switchport [module number]`
	Displays the administrative and operational status of a switching (nonrouting) port, where:
	• *slot*—Specifies the chassis slot number where the SIP is installed.
	• *subslot*—Specifies the secondary slot number on a SIP where a SPA is installed.
	• *port*—Specifies the number of the interface port on the SPA.
	• *t1-number*—Specifies the logical T1 number in channelized mode.
	• *channel-group*—Specifies the logical channel group assigned to the time slots within the T1 or E1 group.
	• **module** *number*—(Optional) Specifies the chassis slot number of the SIP and displays information for all interfaces of the SPAs in that SIP.

The following output of the **show interfaces** commands provide an example of the information that is displayed when BCP is configured in trunk mode:

Note When switch port is configured, the encapsulation is automatically changed to PPP.

```
Router# show interfaces trunk
Port      Mode         Encapsulation  Status        Native vlan
PO4/1/0   on           802.1q         trunking      1

Port      Vlans allowed on trunk
PO4/1/0   1-1005,1025-1026,1028-4094

Port      Vlans allowed and active in management domain
PO4/1/0   1,100,200

Port      Vlans in spanning tree forwarding state and not pruned
PO4/1/0   1,100,200

Router# show interfaces switchport

Name: PO4/1/0
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: down
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: 100
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
```

Configuring Virtual Private LAN Service (VPLS)

Virtual Private LAN Service (VPLS) uses the provider core to simulate a virtual bridge that joins geographically separate LAN segments together. From a customer point of view, there is no topology for VPLS. All of the CE devices appear to connect to a logical bridge emulated by the provider core.

VPLS enables geographically separate LAN segments to be interconnected as a single bridged domain over a packet-switched network, such as IP, MPLS, or a hybrid of both.

For information about configuring VPLS on the SIPs, refer to the “Virtual Private LAN Services on the Optical Services Modules” section of the *OSM Configuration Note* for the Cisco 7600 series router at the following URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SR_OSM_config/mpls.html#wp1423607

Full-mesh, hub and spoke, and Hierarchical VPLS (H-VPLS) with MPLS edge configurations are available.

Full-Mesh Configuration

The full-mesh configuration requires a full mesh of tunnel label switched paths (LSPs) between all the PEs that participate in the VPLS. With full-mesh, signaling overhead and packet replication requirements for each provisioned VC on a PE can be high.

You set up a VPLS by first creating a virtual forwarding instance (VFI) on each participating PE router. The VFI specifies the VPN ID of a VPLS domain, the addresses of other PE routers in the domain, and the type of tunnel signaling and encapsulation mechanism for each peer PE router.

The set of VFIs formed by the interconnection of the emulated VCs is called a VPLS instance; it is the VPLS instance that forms the logic bridge over a packet-switched network. The VPLS instance is assigned a unique VPN ID.

The PE routers use the VFI to establish a full-mesh LSP of emulated VCs to all the other PE routers in the VPLS instance. PE routers obtain the membership of a VPLS instance through static configuration using the Cisco IOS CLI.

Hub and Spoke

In a hub-and-spoke model, the PE router that acts as the hub establishes a point-to-multipoint forwarding relationship with all PE routers at the spoke sites. An Ethernet or VLAN packet received from the customer network on the hub PE can be forwarded to one or more emulated VCs.

The PE routers that act as the spoke establish a point-to-point connection to the PE at the hub site. Ethernet or VLAN packets received from the customer network on the spoke PE are forwarded to the VFI or VPLS instance at the hub.

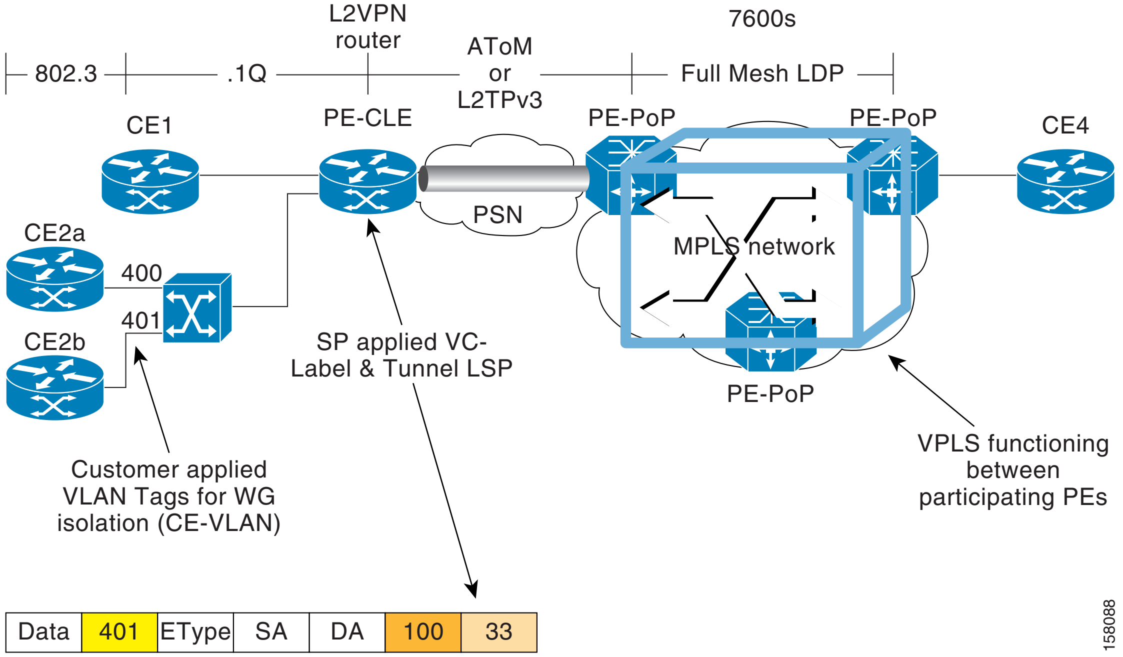
In Cisco IOS Release 12.2(33)SXH and later releases, if there are a number of customer sites connecting to the spoke, you can terminate multiple VCs per spoke into the same VFI or VPLS instance at the hub.

Hierarchical Virtual Private LAN Service (H-VPLS) with MPLS to the Edge

In a flat or non-hierarchical VPLS configuration, a full mesh of pseudowires (PWs) is needed between all PE nodes. A *pseudowire* defines a VLAN and its corresponding pseudoport.

H-VPLS reduces both signaling and replication overhead by using a combination of full-mesh and hub-and-spoke configurations. Hub-and-spoke configurations operate with split horizon to allow packets to be switched between PWs, which effectively reduce the number of PWs between PEs.

Figure 4-3 H-VPLS with MPLS to the Edge Network



In the H-VPLS with MPLS to the edge architecture, Ethernet Access Islands (EAIs) work in combination with a VPLS core network with MPLS as the underlying transport mechanism. EAIs operate like standard Ethernet networks. In Figure 4-3, devices CE1, CE2a, and CE2b reside in an EAI. Traffic from any CE devices within the EAI are switched locally within the EAI by the user-facing provider edge (UPE) device along the computed spanning-tree path. Each user-facing provider edge (UPE) device is connected to one or more network-facing provider edge (NPE) devices using PWs. The traffic local to the UPE is not forward to any network-facing provider edge (NPE) devices.

VPLS Configuration Guidelines

When configuring VPLS on a SIP, consider the following guidelines:

- For support of specific VPLS features by SIP, see Table 4-3.
- The SIPs support up to 4000 VPLS domains per Catalyst 6500 Series switch.
- The SIPs support up to 60 VPLS peers per domain per Catalyst 6500 Series switch.
- The SIPs support up to 30,000 pseudowires, used in any combination of domains and peers up to the 4000-domain or 60-peer maximums. For example, support of up to 4000 domains with 7 peers, or up to 60 peers in 500 domains.
- When configuring VPLS on a Cisco 7600 SIP-600, consider the following guidelines:
  - Q-in-Q (the ability to map a single 802.1Q tag or a random double tag combination into a VPLS instance, a Layer 3 MPLS VPN, or an EoMPLS VC) is not supported.
  - H-VPLS with Q-in-Q edge—Requires a Cisco 7600 SIP-600 in the uplink, and any LAN port or Cisco 7600 SIP-600 on the downlink.
- H-VPLS with MPLS edge requires either an OSM module, Cisco 7600 SIP-600, or Cisco 7600 SIP-400 in both the downlink (facing UPE) and uplink (MPLS core).

- The Cisco 7600 SIP-400 and Cisco 7600 SIP-600 provide Transparent LAN Services (TLS) and Ethernet Virtual Connection Services (EVCS).
- The Cisco 7600 SIP-400 does not support redundant PW links from a UPE to multiple NPEs.
- For information about configuring VPLS on the SIPs, consider the guidelines in this document and then refer to the “Virtual Private LAN Services on the Optical Services Modules” section of the *OSM Configuration Note* for the Cisco 7600 series router at the following URL:

  http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SR_OSM_config/mpls.html#wp1423607

VPLS Feature Compatibility

Table 4-3 provides information about where the VPLS features are supported.

Table 4-3 VPLS Feature Compatibility by SIP and SPA Combination

Feature	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
H-VPLS with MPLS edge	Not supported.	In Cisco IOS Release 12.2(33)SXH and later: • 2-Port Gigabit Ethernet SPA • 2-Port and 4-Port OC-3c/STM-1 POS SPA • 1-Port OC-12c/STM-4 POS SPA • 1-Port OC-48c/STM-16 POS SPA	In Cisco IOS Release 12.2(18)SXF[1] and later: • 1-Port 10-Gigabit Ethernet SPA • 5-Port Gigabit Ethernet SPA • 10-Port Gigabit Ethernet SPA • 1-Port OC-192c/STM-64 POS/RPR SPA • 2-Port and 4-Port OC-48c/STM-16 POS SPA Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
H-VPLS with Q-in-Q edge	Not supported.	Not supported.	In Cisco IOS Release 12.2(18)SXF[1] and later: • 1-Port 10-Gigabit Ethernet SPA • 5-Port Gigabit Ethernet SPA • 10-Port Gigabit Ethernet SPA • 1-Port OC-192c/STM-64 POS/RPR SPA • 2-Port and 4-Port OC-48c/STM-16 POS SPA Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

Table 4-3 VPLS Feature Compatibility by SIP and SPA Combination (continued)

Feature	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
VPLS multiple VCs per spoke		Added in Cisco IOS Release 12.2(33)SXH.	Added in Cisco IOS Release 12.2(33)SXI.
VPLS with point-to-multipoint EoMPLS and fully-meshed PE configuration	Not supported.	In Cisco IOS Release 12.2(33)SXH and later: • 2-Port Gigabit Ethernet SPA • 2-Port and 4-Port OC-3c/STM-1 POS SPA • 1-Port OC-12c/STM-4 POS SPA • 1-Port OC-48c/STM-16 POS SPA	In Cisco IOS Release 12.2(18)SXF[1] and later: • 1-Port 10-Gigabit Ethernet SPA • 5-Port Gigabit Ethernet SPA • 10-Port Gigabit Ethernet SPA • 1-Port OC-192c/STM-64 POS/RPR SPA • 2-Port and 4-Port OC-48c/STM-16 POS SPA Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

[1] Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

Configuring Asymmetric Carrier Delay

After a switchover of redundant links, a local link or port may be declared as link-up before the port is ready to forward data. This condition can result in erroneous routing table convergence and traffic loss. In Cisco IOS Release 12.2(33)SXI and later releases, asymmetric carrier delay (ACD) allows you to configure separate delay values for link-up and link-down event notification for SIP-200 or SIP-400 physical interfaces. With this feature, link-down events can be notified quickly while link-up events can be notified after a delay of sufficient time to ensure that a rebooted port is ready.

ACD Restrictions and Guidelines

When configuring ACD, consider the following restrictions and guidelines:

- ACD cannot be configured on an interface if conventional carrier delay (the **carrier-delay** command without an **up** or **down** keyword) is configured on the interface.
- Link-up carrier delay times are configured in seconds. Link-down carrier delay times are configured in either milliseconds, using the **msec** keyword, or seconds.
- The line card (LC) implements a 4-second debounce timer for link-up events. A configured link-up carrier delay will execute concurrently with the LC debounce timer, and must be 4 seconds or more.
- The route processor (RP) implements a 2-second delay for link-up and link-down events. Configuring a link-down carrier delay time cancels the 2-second RP delay for link-up and link-down events.
- The Fast Link and carrier delay features are mutually exclusive. If you configure either feature on an interface, the other is disabled.

- Administrative shutdown of an interface will force an immediate link-down event regardless of any carrier delay configuration.
- Table 4-4 describes the resulting carrier delay for each configuration and interface event.

Table 4-4 ACD Behavior

ACD Configuration	Interface Event	Total Carrier Delay
carrier-delay down *t_down* (Because a link-down delay is configured, the RP delay is cancelled.)	Transition to down state	*t_down*
	Transition to up state	4 seconds (LC debounce timer)
	Administrative shutdown	0 (immediate shutdown)
	Administrative bring up	4 seconds (LC debounce timer)
carrier-delay up *t_up* (Because a link-down delay is not configured, the RP delay is applied.)	Transition to down state	2 seconds (RP delay)
	Transition to up state	*t_up* + 2 seconds (RP delay)
	Administrative shutdown	0 (immediate shutdown)
	Administrative bring up	*t_up* + 2 seconds (RP delay), minimum 4 seconds
carrier-delay down *t_down* **carrier-delay up** *t_up* (Because a link-down delay is configured, the RP delay is cancelled.)	Transition to down state	*t_down*
	Transition to up state	*t_up*, minimum 4 seconds
	Administrative shutdown	0 (immediate shutdown)
	Administrative bring up	*t_up*, minimum 4 seconds

ACD Configuration Procedure

To configure separate carrier delay values for link-up and link-down events on a SIP-200 or SIP-400 physical interface, perform this task:

	Command or Action	Purpose
Step 1	Router(config)# **interface** *type slot/subslot/port*	Selects the interface to configure.
Step 2	Router(config-if)# **carrier-delay** {**up** *seconds* | **down** {*seconds* | **msec** *milliseconds*}}	Configures the ACD up or down notification delay. • **up** —Specifies the link-up notification delay. • **down**—Specifies the link-down notification delay. • *seconds*—Time, in seconds, to wait for the system to change states. The range is from 0 to 60. The default is 4 seconds for transitions to the up state and 2 seconds for transitions to the down state. • **msec** *milliseconds*—Specifies the link-down notification delay time in milliseconds.
Step 3	Router(config-if)# **end**	Exits the configuration mode.

The following example shows how to configure a carrier delay of 8 seconds for link-up transitions and 50 milliseconds for link-down transitions:

```
Router(config)# interface Fa2/0/0
Router(config-if)# carrier-delay up 8
Router(config-if)# carrier-delay down msec 50
```

Verifying ACD Configuration

To display the carrier delay configuration on a SIP-200 or SIP-400 physical interface, enter the **show running-config** command:

```
Router# show running-config interface Fa2/0/0
Building configuration...

Current configuration: 219 bytes
!
interface FastEthernet2/0/0
ip address 32.0.0.1 255.255.255.0
logging event link-status
carrier-delay up 8
carrier-delay down msec 50
end
```

Configuring MPLS Features on a SIP

Many of the MPLS features supported on the FlexWAN and Enhanced FlexWAN modules on the Catalyst 6500 Series switch, are also supported by the SIPs. For a list of the supported MPLS features on the SIPs, see Chapter 3, "Overview of the SIPs and SSC."

This section describes those MPLS features that have SIP-specific configuration guidelines. After you review the SIP-specific guidelines described in this document, then refer to the following URL for more information about configuring MPLS features:

http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexmpls.html

Configuring Any Transport over MPLS on a SIP

Any Transport over MPLS (AToM) transports Layer 2 packets over a Multiprotocol Label Switching (MPLS) backbone. AToM uses a directed Label Distribution Protocol (LDP) session between edge switches for setting up and maintaining connections. Forwarding occurs through the use of two levels of labels, switching between the edge switches. The external label (tunnel label) routes the packet over the MPLS backbone to the egress Provider Edge (PE) at the ingress PE. The VC label is a demuxing label that determines the connection at the tunnel endpoint (the particular egress interface on the egress PE as well as the virtual path identifier [VPI]/virtual channel identifier [VCI] value for an ATM Adaptation Layer 5 [AAL5] protocol data unit [PDU], the data-link connection identifier [DLCI] value for a Frame Relay PDU, or the virtual LAN [VLAN] identifier for an Ethernet frame).

For specific information about configuring AToM features, refer to the *FlexWAN and Enhanced FlexWAN Module Installation and Configuration Note* located at the following URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexmpls.html

Note When referring to the FlexWAN documentation, be sure to note any SIP-specific configuration guidelines described in this document.

Cisco 7600 SIP-200 AToM Features

The Cisco 7600 SIP-200 supports the following AToM features:

- ATM over MPLS (ATMoMPLS)—AAL5 mode

- Ethernet over MPLS (EoMPLS)—Port mode
- EoMPLS-VLAN mode
- Frame Relay over MPLS (FRoMPLS)
- Hierarchical QoS for EoMPLS VCs

Cisco 7600 SIP-400 AToM Features

The Cisco 7600 SIP-400 supports the following AToM features:

- ATMoMPLS—AAL0 mode (single cell relay only)
- ATMoMPLS—AAL5 mode
- EoMPLS—Port mode
- EoMPLS—VLAN mode
- FRoMPLS—DLCI mode
- In Cisco IOS Release 12.2(33)SXH and later releases:
  - Hierarchical QoS for EoMPLS VCs
  - HDLCoMPLS
  - PPPoMPLS
- In Cisco IOS Release 12.2(33)SXI and later releases:
  - AToM over GRE

Cisco 7600 SIP-400 AToM Configuration Guidelines

When configuring AToM with a Cisco 7600 SIP-400, consider the following guidelines:

- The Cisco 7600 SIP-400 is not supported with a Supervisor Engine 1, Supervisor Engine 1A, Supervisor Engine 2, or Supervisor Engine 720 PFC3A.
- The Cisco 7600 SIP-400 is not supported with PFC-2 based systems.
- For AToM in releases prior to Cisco IOS Release 12.2(33)SXH, the Cisco 7600 SIP-400 does not support the following features when it is located in the data path. You should not configure the following features if the SIP is facing the customer edge (CE) or the MPLS core:
  - HDLCoMPLS
  - PPPoMPLS
  - VPLS
- For AToM in Cisco IOS Release 12.2(33)SXH and later releases, the Cisco 7600 SIP-400 supports the following features on CE-facing interfaces:
  - HDLCoMPLS
  - PPPoMPLS
  - VPLS
- The Cisco 7600 SIP-400 supports EoMPLS with directly connected provider edge (PE) devices when the Cisco 7600 SIP-400 is on the MPLS core side of the network.
- In Cisco IOS Release 12.2(33)SXH and later releases, the Cisco 7600 SIP-400 supports AToM over GRE.

- The Cisco 7600 SIP-400 does not support the ability to enable or disable tunneling of Layer 2 packets, such as for the VLAN Trunking Protocol (VTP), Cisco Discovery Protocol (CDP), and bridge protocol data unit (BPDU). The Cisco 7600 SIP-400 tunnels BPDUs, and always blocks VTP and CDP packets from the tunnel.
- In ATMoMPLS AAL5 and cell mode, the Cisco 7600 SIP-400 supports non-matching VPIs/VCIs between PEs if the Cisco 7600 SIP-400 is on both sides of the network.
- The Cisco 7600 SIP-400 supports matching on FR-DE to set MPLS-EXP for FRoMPLS.
- The Cisco 7600 SIP-400 supports use of the **xconnect** command to configure AToM circuits for all AToM connection types except ATMoMPLS. For ATMoMPLS, you must use the **mpls l2 transport route** command.

  For information about configuring the **xconnect** command for AToM circuits, refer to the MPLS examples using the **xconnect** command at the following URL:

  http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexmpls.html
- The Cisco 7600 SIP-400 does not support local switching for ATM interfaces, but does support local switching for Frame Relay interfaces.
- The Cisco 7600 SIP-400 does not support the following QoS classification features with AToM:
  - Matching on data-link connection identifier (DLCI) is unsupported
  - Matching on virtual LAN (VLAN) is unsupported
  - Matching on class of service (CoS) is unsupported in Cisco IOS Release 12.2(18)SXE and Cisco IOS Release 12.2(18)SXE2 only. Beginning in Cisco IOS Release 12.2(18)SXF, it is supported with the 2-Port Gigabit Ethernet SPA.
  - Matching on input interface is unsupported
  - Matching on packet length is unsupported
  - Matching on media access control (MAC) address is unsupported
  - Matching on protocol type, including Border Gateway Protocol (BGP), is unsupported

Understanding MPLS Imposition on the Cisco 7600 SIP-400 to Set MPLS Experimental Bits

The MPLS imposition function encapsulates non-MPLS frames (such as Ethernet, VLAN, Frame Relay, ATM or IP) into MPLS frames. MPLS disposition performs the reverse function.

An input QoS policy map is applied to ingress packets *before* MPLS imposition takes place. This means that the packets are treated as non-MPLS frames, so any MPLS-related matches have no effect. In the case of marking experimental (EXP) bits using the **set mpls experimental** command, the information is passed to the AToM or MPLS component to set the EXP bits. After imposition takes place, the frame becomes an MPLS frame and an output QoS policy map (if it exists) can apply MPLS-related criteria.

On the egress side, an output QoS policy map is applied to the egress packets *after* MPLS disposition takes place. This means that packets are treated as non-MPLS frames, so any MPLS-related criteria has no effect. Before disposition, the frame is an MPLS frame and the input QoS policy map (if it exists) can apply MPLS-related criteria.

The Encoded Address Recognition Logic (EARL) is a centralized processing engine for learning and forwarding packets based upon MAC address on the Catalyst 6500 Series switch supervisor engines. The EARL stores the VLAN, MAC address, and port relationships. These relationships are used to make switching decisions in hardware. The EARL engine also performs MPLS imposition, and the MPLS EXP bits are copied either from the IP TOS field (using **trust dscp** or **trust precedence** mode), or from the DBUS header QoS field (using **trust cos** mode).

When using the 2-Port Gigabit Ethernet SPA with the Cisco 7600 SIP-400 as the customer-side interface configured for 802.1Q encapsulation for IP imposition with MPLS, the Layer 2 CoS value is not automatically copied into the corresponding MPLS packet's EXP bits. Instead, the value in the IP precedence bits is copied.

To maintain the 802.1Q CoS values, classify the imposition traffic on the customer-facing Gigabit Ethernet interface in the input direction to match on CoS value, and then set the MPLS experimental action for that class as shown in the following example:

```
Router(config)# class-map cos0
Router(config-cmap)# match cos 0
Router(config-cmap)# exit
!
Router(config)# class-map cos1
Router(config-cmap)# match cos 1
Router(config-cmap)# exit
!
Router(config)# policy-map policy1
Router(config-pmap)# class cos0
Router(config-pmap-c)# set mpls experimental imposition 0
Router(config-pmap-c)# exit
Router(config-pmap)# class cos1
Router(config-pmap-c)# set mpls experimental imposition 1
```

Cisco 7600 SIP-600 AToM Features

The Cisco 7600 SIP-600 supports the following AToM features:

- Any Transport over MPLS (AToM) support—EoMPLS only (Encoded Address Recognition Logic [EARL]-based and SIP-based EoMPLS)

Configuring Hierarchical Virtual Private LAN Service (H-VPLS) with MPLS to the Edge

The Cisco 7600 SIP-400 and Cisco 7600 SIP-600 support the H-VPLS with MPLS to the Edge feature. For more information about VPLS support on the SIPs, see the "Configuring Virtual Private LAN Service (VPLS)" section on page 4-23.

Configuring QoS Features on a SIP

This section describes configuration of the SIP-specific QoS features. Many of the QoS features supported on the FlexWAN and Enhanced FlexWAN modules on the Catalyst 6500 Series switch are also supported by the SIPs. For a list of the supported QoS features on the SIPs, see Chapter 3, "Overview of the SIPs and SSC."

This section describes those QoS features that have SIP-specific configuration guidelines. After you review the SIP-specific guidelines described in this document, then refer to the *FlexWAN and Enhanced FlexWAN Module Installation and Configuration Note* located at the following URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexwan-config-guide.html

This section includes the following topics:

- General QoS Feature Configuration Guidelines, page 4-34
- Configuring QoS Features Using MQC, page 4-35
- Configuring QoS Traffic Classes on a SIP, page 4-35

- Configuring QoS Class-Based Marking Policies on a SIP, page 4-41
- Configuring QoS Congestion Management and Avoidance Policies on a SIP, page 4-44
- Configuring Dual Priority Queuing on a Cisco 7600 SIP-400, page 4-47
- Configuring QoS Traffic Shaping Policies on a SIP, page 4-49
- Configuring QoS Traffic Policing Policies on a SIP, page 4-50
- Attaching a QoS Traffic Policy to an Interface, page 4-55
- Configuring Network-Based Application Recognition and Distributed Network-Based Application Recognition, page 4-56
- Configuring Hierarchical QoS on a SIP, page 4-58
- Configuring PFC QoS on a Cisco 7600 SIP-600, page 4-60

General QoS Feature Configuration Guidelines

This section identifies some general QoS feature guidelines for certain types of SPAs. You can find other feature-specific SIP and SPA configuration guidelines and restrictions in the other QoS sections of this chapter.

ATM SPA QoS Configuration Guidelines

For the 2-Port and 4-Port OC-3c/STM-1 ATM SPA, the following applies:

- In the ingress direction, all Quality of Service (QoS) features are supported by the Cisco 7600 SIP-200.
- In the egress direction:
  - All queueing based features (such as class-based weighted fair queueing [CBWFQ], and ATM per-VC WFQ, and WRED) are implemented on the Segmentation and Reassembly (SAR) processor on the SPA.
  - Policing is implemented on the SIP.
  - Class queue shaping is not supported.

Gigabit Ethernet SPA QoS Configuration Guidelines

For the 2-Port Gigabit Ethernet SPA, the following QoS behavior applies:

- In both the ingress and egress directions, all QoS features calculate packet size similarly to how packet size calculation is performed by the FlexWAN and Enhanced FlexWAN modules on the Catalyst 6500 Series switch.
- Specifically, all features consider the IEEE 802.3 Layer-2 headers and the Layer-3 protocol payload. The CRC, interframe gap, and preamble are not included in the packet size calculations.

Note For Fast Ethernet SPAs, QoS cannot change the speed of an interface (for example, Fast Ethernet SPAs cannot change QoS settings whenever an interface speed is changed between 100 Mbps to 10 Mbps). When the speed is changed, the user must also adjust the QoS setting accordingly.

Configuring QoS Features Using MQC

The Modular QoS CLI (MQC) is a CLI structure that allows users to create traffic policies and attach these policies to interfaces. A traffic policy contains a traffic class and one or more QoS features. A traffic class is used to select traffic, while the QoS features in the traffic policy determine how to treat the classified traffic.

If you apply a traffic policy at a main interface that also contains subinterfaces, then all of the traffic that goes through the subinterfaces is processed according to the policy at the main interface. For example, if you configure a traffic shaping policy at the main interface, all of the traffic going through the subinterfaces is aggregated and shaped to the rate defined in the traffic shaping policy at the main interface.

To configure QoS features using the Modular QoS CLI on the SIPs, complete the following basic steps:

Step 1 Define a traffic class using the **class-map** command.

Step 2 Create a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).

Step 3 Attach the traffic policy to the interface using the **service-policy** command.

For a complete discussion about MQC, refer to the "Modular Quality of Service Command-Line Interface Overview" chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2* publication at:

http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfmdcli.html

Configuring QoS Traffic Classes on a SIP

Use the QoS classification features to select your network traffic and categorize it into classes for further QoS processing based on matching certain criteria. The default class, named class-default, is the class to which traffic is directed for any traffic that does not match any of the selection criteria in the configured class maps.

QoS Traffic Class Configuration Guidelines

When configuring traffic classes on a SIP, consider the following guidelines:

- You can define up to 256 unique class maps.
- A single class map can contain up to 8 different **match** command statements.
- For ATM bridging, Frame Relay bridging, MPB, and BCP features, the following matching features are supported on bridged frames in Cisco IOS Release 12.2(33)SXH and later releases:
  - Matching on ATM CLP bit (input interface only)
  - Matching on COS
  - Matching on Frame Relay DE bit (input interface only)
  - Matching on Frame Relay DLCI
  - Matching on inner COS
  - Matching on inner VLAN
  - Matching on IP DSCP

  - Matching on IP precedence
  - Matching on VLAN
- The Cisco 7600 SIP-600 does not support combining matches on QoS group or input VLAN with other types of matching criteria (for example, access control lists [ACLs]) in the same class or policy map.
- The Cisco 7600 SIP-400 supports matching on ACLs for routed traffic only. Matching on ACLs is not supported for bridged traffic.
- When configuring hierarchical QoS on the Cisco 7600 SIP-600, if you configure matching on an input VLAN in a parent policy, then only matching on a QoS group is supported in the child policy.
- For support of specific matching criteria by SIP, see Table 4-5.

To create a user-defined QoS traffic class, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# class-map [match-all | match-any] class-name`	Creates a traffic class, where: • **match-all**—(Optional) Specifies that all match criteria in the class map must be matched, using a logical AND of all matching statements defined under the class. This is the default. • **match-any**—(Optional) Specifies that one or more match criteria must match, using a logical OR of all matching statements defined under the class. • *class-name*—Specifies the user-defined name of the class. **Note** You can define up to 256 unique class maps.
Step 2	`Router(config-cmap)# match type`	Specifies the matching criterion to be applied to the traffic, where *type* represents one of the forms of the **match** command supported by the SIP as shown in Table 4-5. **Note** A single class-map can contain up to 8 different **match** command statements.

Table 4-5 provides information about which QoS classification features are supported for SIPs on the Catalyst 6500 Series switch. For more information about most of the commands documented in this table, refer to the *Cisco IOS Quality of Service Solutions Command Reference.*

Table 4-5 QoS Classification Feature Compatibility by SIP

Feature (match command)	**Cisco 7600 SIP-200**	**Cisco 7600 SIP-400**	**Cisco 7600 SIP-600**[1]
Matching on access list (ACL) number (**match access-group** command)	Supported for all SPAs with the following types of ACLs: • Protocols—ICMP, IGMP, EIGRP, OSPF, PIM, and GRE • Source and destination port • TCP flags • ToS (DSCP and precedence)	Supported for all SPAs with the following types of ACLs: • Source and destination port • TCP flag (IPv4 only) • IP address (IPv6 compress mode only)	Supported for all SPAs[1] with the following types of ACLs: • IPv4 and IPv6 • Protocols—ICMP, IGMP, UDP, and MAC • Source and destination ports • TCP flags • ToS Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Matching on ACL name (**match access-group name** command)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs.[1] Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Match on any packet (**match any** command) **Note** Not supported for user-defined class maps.	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs.[1] Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Matching on ATM cell loss (CLP) (**match atm clp** command)	• Supported for all ATM SPAs. • Cisco IOS Release 12.2(33)SXH—Support added for ATM CLP matching with RFC 1483 bridging features.	• Supported for all ATM SPAs on ATM input interface only. • Cisco IOS Release 12.2(33)SXH—Support added for ATM CLP matching with RFC 1483 bridging features on ATM input interface only.	Not supported.

Table 4-5 QoS Classification Feature Compatibility by SIP (continued)

Feature (match command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Matching on class map (**match class-map** command)	Supported for all SPAs.	Not supported.	Not supported.
Matching on Class of Service (COS) (**match cos** command)	Supported in Cisco IOS Release 12.2(33)SXH on the 4-Port and 8-Port Fast Ethernet SPA using dot1q encapsulation.	• 2-Port Gigabit Ethernet SPA only—Input and output 802.1Q tagged frames. • Cisco IOS Release 12.2(33)SXH—Support added for inner COS matching with bridging features.	Not supported.
Matching on Inner COS (**match cos inner** command)	• Supported for all SPAs. • Cisco IOS Release 12.2(33)SXH—Support added for inner COS matching with bridging features.	Supported in Cisco IOS Release 12.2(33)SXH on the 2-Port Gigabit Ethernet SPA: • Input and output interfaces. • Inner COS matching with bridging features.	Not supported.
Match on Frame Relay discard eligibility (DE) bit (**match fr-de** command)	• Supported for Frame Relay input and output interfaces. • Cisco IOS Release 12.2(33)SXH—Support added for Frame Relay DE matching with Frame Relay bridging features.	• Supported for a Frame Relay input interface only. • Cisco IOS Release 12.2(33)SXH—Support added for Frame Relay DE matching with Frame Relay bridging features on input Frame Relay interface only. **Note** Since the Cisco 7600 SIP-400 acts as a Frame Relay data terminal equipment (DTE) device only, and not a data communications equipment (DCE) device, the Cisco 7600 SIP-400 does not support dropping of frames that match on FR DE bits; however, other QoS actions are supported.	Not supported.

Table 4-5 QoS Classification Feature Compatibility by SIP (continued)

Feature (match command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Match on Frame Relay data-link connection identifier (DLCI) (**match fr-dlci** command)	• Supported for Frame Relay input and output interfaces. • Cisco IOS Release 12.2(33)SXH—Support added for Frame Relay DLCI matching with Frame Relay bridging features.	Supported in Cisco IOS Release 12.2(33)SXH on Frame Relay input and output interfaces, and with Frame Relay bridging features.	Not supported.
Match on input VLAN (**match input vlan** command—Matches the VLAN from an input interface.)	Supported for EoMPLS interfaces.	Supported in Cisco IOS Release 12.2(33)SXH—Output interface only, and with bridging features. **Note** Service policy is applied on the output interface of the Cisco 7600 SIP-400 to match the VLAN from the input interface.	Not supported.
Match on IP DSCP (**match ip dscp** command)	• Supported for all SPAs. • Cisco IOS Release 12.2(33)SXH—Support added for IP DSCP matching with bridging features on an input interface only.	• Supported for all SPAs. • Cisco IOS Release 12.2(33)SXH—Support added for IP DSCP matching with bridging features.	Supported for all SPAs.[1] Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Match on IP precedence (**match ip precedence** command)	Supported for all SPAs.	• Supported for all SPAs. • Cisco IOS Release 12.2(33)SXH—Support added for IP precedence matching with bridging features.	Supported for all SPAs.[1] Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Match on IP Real-Time Protocol (RTP) (**match ip rtp** command)	Supported for all SPAs.	Not supported.	Not supported.
Match on MAC address for an ACL name (**match mac address** command)	Not supported.	Not supported.	Not supported.

Table 4-5 QoS Classification Feature Compatibility by SIP (continued)

Feature (match command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Match on destination MAC address (**match destination-address mac** command)	Not supported.	Not supported.	Not supported.
Match on source MAC address (**match source-address mac** command)	Not supported.	Not supported.	Not supported.
Match on MPLS experimental (EXP) bit (**match mpls experimental** command)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs.
Match on Layer 3 packet length in IP header (**match packet length** command)	Supported for all SPAs.	Not supported.	Not supported.
Match on QoS group (**match qos-group** command)	Not supported.	Supported in Cisco IOS Release 12.2(33)SXH—Output interface only.	Supported in software-based EoMPLS configurations only using hierarchical QoS, where the parent policy configures matching on input VLAN and the child policy configures matching on QoS group.[1] Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Match on protocol (**match protocol** command	Supported for NBAR.	Not supported.	Supports matching on IP and IPv6.[1] Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

Table 4-5 QoS Classification Feature Compatibility by SIP (continued)

Feature (match command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Match on VLAN (**match vlan** command—Matches the outer VLAN of a Layer 2 802.1Q frame)	Not supported.	Supported in Cisco IOS Release 12.2(33)SXH: • Input and output interfaces. • Outer VLAN ID matching for 802.1Q tagged frames.	Not supported.
Match on VLAN Inner (**match vlan inner** command—Matches the innermost VLAN of the 802.1Q tag in the Layer 2 frame)	• Supported for all SPAs. • Cisco IOS Release 12.2(33)SXH—Support added for inner VLAN ID matching with bridging features.	Supported in Cisco IOS Release 12.2(33)SXH: • Input and output interface. • Inner VLAN ID matching with bridging features.	Not supported.
No match on specified criteria (**match not** command)	Supported for all SPAs.	Supported for all SPAs.	Not supported.

[1] Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

Configuring QoS Class-Based Marking Policies on a SIP

After you have created your traffic classes, you can configure traffic policies to configure marking features to apply certain actions to the selected traffic in those classes.

In most cases, the purpose of a packet mark is identification. After a packet is marked, downstream devices identify traffic based on the marking and categorize the traffic according to network needs. This categorization occurs when the **match** commands in the traffic class are configured to identify the packets by the mark (for example, **match ip precedence**, **match ip dscp**, **match cos**, and so on). The traffic policy using this traffic class can then set the appropriate QoS features for the marked traffic.

In some cases, the markings can be used for purposes besides identification. Distributed WRED, for instance, can use the IP precedence, IP DSCP, or MPLS EXP values to detect and drop packets. In ATM networks, the CLP bit of the packet is used to determine the priority of packet in a congested environment. If congestion occurs in the ATM network, packets with the CLP bit set to 1 are dropped before packets with the CLP bit set to 0. Similarly, the DE bit of a Frame Relay frame is used to determine the priority of a frame in a congested Frame Relay network. In Frame Relay networks, frames with the DE bit set to 1 are dropped before frames with the DE bit set to 0.

QoS Class-Based Marking Policy Configuration Guidelines

When configuring class-based marking on a SIP, consider the following guidelines:

- Packet marking is supported on interfaces, subinterfaces, and ATM virtual circuits (VCs). In an ATM PVC, you can configure packet marking in the same traffic policy where you configure the queueing actions, on a per-VC basis. However, only PVC configuration of service policies is supported for classes using multipoint bridging (MPB) match criteria.
- For ATM bridging, Frame Relay bridging, MPB, and BCP features, the following marking features are supported on bridged frames in Cisco IOS Release 12.2(33)SXH and later releases:
  - Set ATM CLP bit (output interface only)
  - Set Frame Relay DE bit (output interface only)

  - Set inner COS

- If a service policy configures both class-based marking and marking as part of a policing action, then the marking using policing takes precedence over any class-based marking.
- The Cisco 7600 SIP-600 supports marking on input interfaces only.
- For support of specific marking criteria by SIP, see Table 4-6.

To configure a QoS traffic policy with class-based marking, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map-name`	Creates or modifies a traffic policy and enters policy map configuration mode, where: • *policy-map-name*—Specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
Step 2	`Router (config-pmap)# class {class-name | class-default}`	Specifies the name of the traffic class to which this policy applies and enters policy-map class configuration mode, where: • *class-name*—Specifies that the policy applies to a user-defined class name previously configured. • **class-default**—Specifies that the policy applies to the default traffic class.
Step 3	`Router(config-pmap-c)# set type`	Specifies the marking action to be applied to the traffic, where *type* represents one of the forms of the **set** command supported by the SIP as shown in Table 4-6.

Table 4-6 provides information about which QoS class-based marking features are supported for SIPs on the Catalyst 6500 Series switch.

Table 4-6 QoS Class-Based Marking Feature Compatibility by SIP

Marking Feature (set command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Set ATM CLP bit (**set atm-clp** command—Mark the ATM cell loss bit with value of 1)	• Supported for ATM output interfaces only. • Cisco IOS Release 12.2(33)SXH—Support added for ATM CLP marking on output interfaces only with RFC 1483 bridging features.	Supported for ATM SPA output interfaces only.	Not supported.
Set discard class (**set discard-class** command—Marks the packet with a discard class value for per-hop behavior)	Not supported.	Not supported.	Not supported.

Table 4-6 QoS Class-Based Marking Feature Compatibility by SIP (continued)

Marking Feature (set command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Set Frame Relay DE bit (**set fr-de** command—Mark the Frame Relay discard eligibility bit with value of 1)	• Supported for Frame Relay output interfaces only. • Cisco IOS Release 12.2(33)SXH—Support added for Frame Relay DE marking on output interfaces only with Frame Relay bridging features.	Supported for Frame Relay output interfaces only.	Not supported.
Set IP DSCP (**set ip dscp** command—Marks the IP differentiated services code point (DSCP) in the type of service (ToS) byte with a value from 0–63.)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs on an input interface. Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Set IP precedence (**set ip precedence** command—Marks the precedence value in the IP header with a value from 0–7.)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs on an input interface. Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Set Layer 2 802.1Q COS (**set cos** command—Marks the COS value from 0–7 in an 802.1Q tagged frame.)	• Supported for all SPAs. • In Cisco IOS Release 12.2(33)SXH—Not supported with **set cos-inner** command on the same interface.	Supported in Cisco IOS Release 12.2(33)SXH.	Not supported.
Set Layer 2 802.1Q COS (**set cos-inner** command—Marks the inner COS field from 0–7 in a bridged frame.)	Supported in Cisco IOS Release 12.2(33)SXH with bridging features on the 4-Port and 8-Port Fast Ethernet SPA.	Supported in Cisco IOS Release 12.2(33)SXH with bridging features.	Not supported.

Table 4-6 QoS Class-Based Marking Feature Compatibility by SIP (continued)

Marking Feature (set command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Set MPLS experimental (EXP) bit on label imposition (**set mpls experimental imposition** command)	Supported for all SPAs.	Supported for any SPA IP input interface. **Note** The **table** keyword is not supported.	Supported for all SPAs on an input interface.[1] Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Set MPLS EXP topmost (**set mpls experimental topmost** command)	Supported for all SPAs.	Supported for any SPA MPLS interface.	Not supported.
Set QoS group (**set qos-group** command—Marks the packet with a QoS group association.)	Not supported.	Not supported.	Supported only for software-based EoMPLS on an input SPA switchport interface.[1] Note: Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH.

[1] Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

For more detailed information about configuring class-based marking features, refer to the *Class-Based Marking* document located at the following URL:

http://www.cisco.com/en/US/docs/ios/12_1t/12_1t5/feature/guide/cbpmark2.html

Note When referring to the class-based marking documentation, be sure to note any SIP-specific configuration guidelines described in this document.

Configuring QoS Congestion Management and Avoidance Policies on a SIP

This section describes SIP- and SPA-specific information for configuring QoS traffic policies for congestion management and avoidance features. These features are generally referred to as queueing features.

QoS Congestion Management and Avoidance Policy Configuration Guidelines

When configuring queueing features on a SIP, consider the following guidelines:

- The Catalyst 6500 Series switch supports different forms of queueing features. See Table 4-7 to determine which queueing features are supported by SIP type.
- The Cisco 7600 SIP-200 and Cisco 7600 SIP-400 do not support ingress queueing features.
- When configuring queueing on the Cisco 7600 SIP-400, consider the following guidelines:

  - A queue on the Cisco 7600 SIP-400 is not assured any minimum bandwidth.
  - You cannot configure bandwidth or shaping with queueing under the same class in a service policy on the Cisco 7600 SIP-400.
  - If you want to define bandwidth parameters under different classes in the same service policy on the Cisco 7600 SIP-400, then you only can use the **bandwidth remaining percent** command. The Cisco 7600 SIP-400 does not support other forms of the **bandwidth** command with queueing in the same service policy.
- You can use policing with queueing to limit the traffic rate.
- On the Cisco 7600 SIP-400, WRED is supported on bridged VCs with classification on precedence and DSCP values. On other SIPs, WRED does not work on bridged VCs (for example, VCs that implement MPB).
- When configuring WRED on the Cisco 7600 SIP-400, consider the following guidelines:
  - WRED is supported on bridged VCs with classification on precedence and DSCP values.
  - WRED explicit congestion notification (ECN) is not supported for output traffic on ATM SPAs.
  - ECN is supported for IP traffic on output POS interfaces only.
  - You can use the low-order TOS bits in the IP header for explicit congestion notification (ECN) for WRED. If you configure **random-detect ecn** in a service policy and apply it to either a POS interface or a VC on a POS interface, then if at least one of the ECN bits is set and the packet is a candidate for dropping, the Cisco 7600 SIP-400 marks both ECN bits. If either one of the ECN bits is set, the Cisco 7600 SIP-400 will not drop the packet.
  - WRED ECN is not support for MPLS packets.
- On the Cisco 7600 SIP-400, the default queue limit is calculated based on the number of 250-byte packets that the SIP can transmit in one half of a second. For example, for an OC-3 SPA with a rate of 155 Mbps, the default queue limit is 38,750 packets (155000000 x 0.5 / 250 x 8).
- For more detailed information about configuring congestion management features, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide* document corresponding to your Cisco IOS software release.

Table 4-7 provides information about which QoS queueing features are supported for SIPs on the Catalyst 6500 Series switch:

Table 4-7 QoS Congestion Management and Avoidance Features by SIP and SPA Combination

Congestion Management and Avoidance Feature	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Aggregate Weighted Random Early Detection (**random-detect aggregate**, **random-detect dscp (aggregate)**, and **random-detect precedence (aggregate)** commands)	Supported for ATM SPA PVCs only—Cisco IOS Release 12.2(18)SXE and later.	Supported for ATM SPA PVCs only—Cisco IOS Release 12.2(18)SXE and later.	Supported for all SPAs.[1] For more information on configuring aggregate WRED, see the "Configuring Aggregate WRED for PVCs" section on page 7-26.
Class-based Weighted Fair Queueing (CBWFQ) (**bandwidth**, **queue-limit** commands)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs.[1]

Table 4-7 QoS Congestion Management and Avoidance Features by SIP and SPA Combination (continued)

Congestion Management and Avoidance Feature	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Dual-Queue Support (**priority** and **priority level** commands)	Not supported.	Supported for all SPAs except ATM SPAs—Cisco IOS Release 12.2(33)SXI and later.	Not supported.
Flow-based Queueing (fair queueing/WFQ) (**fair-queue** command)	Supported for all SPAs.	Not supported.	Not supported.
Low Latency Queueing (LLQ)/ Queueing (**bandwidth** command)	Strict priority only—Supported for all SPAs.	Strict priority only—Supported for all SPAs.	Supported for all SPAs.[1]
Random Early Detection (RED) (**random-detect** commands)	Supported for all SPAs.	Supported for all SPAs. • ATM SPAs—Up to 106 unique WRED minimum threshold (min-th), maximum threshold (max-th), and mark probability profiles supported. • Other SPAs—Up to 128 unique WRED min-th, max-th, and mark probability profiles supported.	Not supported.
Weighted RED (WRED)	Supported for all SPAs, with the following exception: • WRED is not supported on bridged VCs.	Supported for all SPAs, with the following restriction: • WRED is supported on bridged VCs with classification on precedence and DSCP values.	Not supported.

[1] Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

To configure a QoS CBWFQ policy, perform the following task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`policy-map`** *`policy-map-name`*	Creates or modifies a traffic policy and enters policy map configuration mode, where: • *policy-map-name*—Specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
Step 2	`Router (config-pmap)#` **`class`** `{`*`class-name`* `|` **`class-default`**`}`	Specifies the name of the traffic class to which this policy applies and enters policy-map class configuration mode, where: • *class-name—*Specifies that the policy applies to a user-defined class name previously configured. • **class-default**—Specifies that the policy applies to the default traffic class.
Step 3	`Router(config-pmap-c)#` **`bandwidth`** `{`*`bandwidth-kbps`* `|` **`percent`** *`percent`*`}`	Specifies the bandwidth allocated to a class belonging to a policy map. **Note** The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead. • *bandwidth-kbps*—Specifies the amount of bandwidth, in number of kbps, to be assigned to a class. • **percent**—Specifies the amount of guaranteed bandwidth, based on the absolute percent of available bandwidth. • *percentage*—Used in conjunction with the percent keyword, the percentage of the total available bandwidth to be set aside for the priority classes.
Step 4	`Router(config-pmap-c)#` **`queue-limit`** *`number-of-packets`*	Specifies the maximum number of packets the queue can hold for a class policy configured in a policy map. • *number-of-packets*—A number in the range 1 to 64 specifying the maximum number of packets that the queue for this class can accumulate.

Configuring Dual Priority Queuing on a Cisco 7600 SIP-400

When configuring Dual Priority Queuing, consider the following guidelines:

- Only two priority levels are supported.
- Level 1 is higher than level 2.
- Propagation is supported on both levels.
- A priority without a level is mapped to level 1.
- The sum of bandwidth percentage and another queues' bandwidth reservation must not exceed 100% bandwidth.

- The police rate includes a Layer 2 header but not cyclic redundancy check (CRC), preamble, or interframe gap.
- Dual priority queuing is not supported on ATM SPAs.

To configure dual priority queuing, perform the following task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **policy-map** *policy-map-name*	Creates or modifies a traffic policy and enters policy map configuration mode, where: • *policy-map-name*—Specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
Step 2	Router (config-pmap)# **class** {*class-name* | **class-default**}	Specifies the name of the traffic class to which this policy applies and enters policy-map class configuration mode, where: • *class-name*—Specifies that the policy applies to a user-defined class name previously configured. • **class-default**—Specifies that the policy applies to the default traffic class.
Step 3	Router(config-pmap-c)# **priority level** [**1** | **2**]	Gives priority to a class of traffic belonging to a policy map. Two priority levels are supported—1 (higher priority) and 2 (lower priority). If no level is specified, the default priority of 1 is assigned.
	Router(config-pmap-c)# **priority** [**level 1** | **2**] *kbps* [*burst*]	Enables conditional policing rate as a data rate to be given to a class of traffic. Conditional policing is used if the logical or physical link is congested. • *kbps*—Specifies the rate in kbps, from 1 to 2480000 kbps. • *burst*—(Optional) Specifies the burst size in bytes, from 18 to 2000000 bytes. The burst size configures the network to accommodate temporary bursts of traffic.
	Router(config-pmap-c)# **priority** [**level 1** | **2**] **percent** *percentage* [*burst*]	Enables conditional policing rate as a percentage of total bandwidth to be given to a class of traffic. Conditional policing is used if the logical or physical link is congested. • *percentage*—Specifies the percentage of total bandwidth, from 1 to 100 percent. • *burst*—(Optional) Specifies the burst size in bytes, from 18 to 2000000 bytes. The burst size configures the network to accommodate temporary bursts of traffic.

The **level** keyword can be combined with the policing configuration, as in the following examples:

```
Router(config-pmap-c)# priority level 2 1024 10000
Router(config-pmap-c)# priority level 2 percent 20 2000
```

Configuring QoS Traffic Shaping Policies on a SIP

This section describes SIP- and SPA-specific information for configuring QoS traffic policies for shaping traffic.

QoS Traffic Shaping Policy Configuration Guidelines

When configuring queueing features on a SIP, consider the following guidelines:

- The Catalyst 6500 Series switch supports different forms of queueing features. See Table 4-8 to determine which traffic shaping features are supported by SIP type.
- Use a hierarchical policy if you want to achieve minimum bandwidth guarantees using CBWFQ with a Frame Relay map class. First, configure a parent policy to shape to the total bandwidth required (on the Cisco 7600 SIP-400, use the class-default in Cisco IOS Release 12.2(18)SXF, or a user-defined class in Cisco IOS Release 12.2(33)SXH and later releases). Then, define a child policy using CBWFQ for the minimum bandwidth percentages.
- ATM SPAs do not support MQC-based traffic shaping. You need to configure traffic shaping for ATM interfaces using ATM Layer 2 VC shaping.
- For more detailed information about configuring congestion management features, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide* document corresponding to your Cisco IOS software release.

Table 4-8 provides information about which QoS traffic shaping features are supported for SIPs on the Catalyst 6500 Series switch.

Table 4-8 QoS Traffic Shaping Feature Compatibility by SIP and SPA Combination

Traffic Shaping Feature (shape command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Adaptive shaping for Frame Relay (**shape adaptive** command)	Supported for all SPAs.	Not supported.	Not supported.
Class-based shaping (**shape average**, **shape peak** commands)	Supported for all SPAs.	Supported for all SPAs, with the following exceptions: • Committed burst (bc)—Not supported. • Excess burst (be)—Not supported.	Supports **shape average** only for all SPAs.[1]
Policy-map class shaping of average-rate of traffic by percentage of bandwidth (**shape average percent command**)	Not supported.	Supported for all SPAs.	Not supported.
Policy-map class shaping with adaptation to backward explicit congestion notification (BECN) (**shape adaptive** command)	Supported for all SPAs.	Not supported.	Not supported.

Table 4-8 QoS Traffic Shaping Feature Compatibility by SIP and SPA Combination (continued)

Traffic Shaping Feature (shape command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Policy-map class shaping with reflection of forward explicit congestion notification (FECN) as BECN (**shape fecn-adapt** command)	Supported for all SPAs.	Not supported.	Not supported.
Policy-map class shaping of peak-rate of traffic by percentage of bandwidth (**shape peak percent** command)	Not supported.	Not supported.	Not supported.

[1] Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

Configuring QoS Traffic Policing Policies on a SIP

This section describes SIP- and SPA-specific information for configuring QoS traffic policing policies.

QoS Traffic Policing Policy Configuration Guidelines

When configuring traffic policing on a SIP, consider the following guidelines:

- The Catalyst 6500 Series switch supports different forms of policing using the **police** command. See Table 4-9 to determine which policing features are supported by SIP type.
- When configuring policing on the Cisco 7600 SIP-600, consider the following guidelines:
  - The Cisco 7600 SIP-600 supports **conform-action** policing on input interfaces only, unless it is being implemented with queueing.
  - The Cisco 7600 SIP-600 does not support any policing actions (shown in Table 4-10) using the **exceed-action** or **violate-action** keywords on an input interface.
  - The Cisco 7600 SIP-600 supports **exceed-action** policing on an output interface with a **drop** action only, when the policing is being implemented with queueing.
  - The Cisco 7600 SIP-600 supports marking for **exceed-action** policing only using the **set-dscp-transmit** command.
- When configuring a policing service policy and specifying the CIR in bits per second without specifying the optional conform (bc) or peak (be) burst in bytes, the Cisco 7600 SIP-400 calculates the burst size based on the number of bytes that it can transmit in 250 ms using the CIR value. For example, a CIR of 1 Mbps (or 1,000,000 bps) is equivalent to 125,000 bytes per second, which is 125 bytes per millisecond. The calculated burst is 250 x 125 = 31250 bytes. If the calculated burst is less than the interface maximum transmission unit (MTU), then the interface MTU is used as the burst size.
- You can use policing with queueing to limit the traffic rate.
- If a service policy configures both class-based marking and marking as part of a policing action, then the marking using policing takes precedence over any class-based marking.
- When configuring policing with MPB features on the Cisco 7600 SIP-200 and Cisco 7600 SIP-400, the **set-cos-inner-transmit** command is supported in Cisco IOS Release 12.2(33)SXH and later releases.

Table 4-9 provides information about which policing features are supported for SIPs on the Catalyst 6500 Series switch.

Table 4-9 QoS Policing Feature Compatibility by SIP and SPA Combination

Policing Feature (police command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Policing by aggregate policer (**police aggregate** command)	Not supported.	Not supported.	Supported for all SPAs.[1]
Policing by bandwidth using token bucket algorithm (**police** command)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAS.[1]
Policing by committed information rate (CIR) percentage (**police (percent)** command—**police cir percent** form)	Supported for all SPAs.	Supported for all SPAs.	Not supported.
Policing with 2-color marker (CIR and peak information rate [PIR]) (**police (two rates)** command—**police cir pir** form)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs.[1]
Policing by flow mask (**police flow mask** command)	Not supported.	Not supported.	Supported for all SPAs.[1]
Policing by microflow (**police flow** command)	Not supported.	Not supported.	Supported for all SPAs.[1]

[1] Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

To create QoS traffic policies with policing, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map-name`	Creates or modifies a traffic policy and enters policy map configuration mode, where: • *policy-map-name*—Specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
Step 2	`Router (config-pmap)# class {class-name | class-default}`	Specifies the name of the traffic class to which this policy applies and enters policy-map class configuration mode, where: • *class-name*—Specifies that the policy applies to a user-defined class name previously configured. • **class-default**—Specifies that the policy applies to the default traffic class.
	Use one of the following forms of **police** commands to evaluate traffic for the specified class. See Table 4-9 to determine which SIPs support the different policing features.	

	Command	Purpose
Step 3	Router(config-pmap-c)# **police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* **violate-action** *action*	Specifies a maximum bandwidth usage by a traffic class through the use of a token bucket algorithm, where: • *bps*—Specifies the average rate in bits per second. Valid values are 8000 to 200000000. • *burst-normal*—(Optional) Specifies the normal burst size in bytes. Valid values are 1000 to 51200000. The default normal burst size is 1500 bytes. • *burst-max*—(Optional) Specifies the excess burst size in bytes. Valid values are 1000 to 51200000. • *action*—Specifies the policing command (as shown in Table 4-10) for the action to be applied to the corresponding conforming, exceeding, or violating traffic.
Step 4	Router(config-pmap-c)# **police cir percent** *percentage* [*burst-in-msec*] [**bc** *conform-burst-in-msec*] [**pir percent** *percentage*] [**be** *peak-burst-in-msec*] [**conform-action** *action* [**exceed-action** *action* [**violate-action** *action*]]]	Configures traffic policing on the basis of a percentage of bandwidth available on an interface, where: • **cir percent** *percentage*—Specifies the committed information rate (CIR) bandwidth percentage. Valid values are 1 to 100. • *burst-in-msec*—(Optional) Burst in milliseconds. Valid values are 1 to 2000. • **bc** *conform-burst-in-msec*—(Optional) Specifies the conform burst (bc) size used by the first token bucket for policing traffic in milliseconds. Valid values are 1 to 2000. • **pir percent** *percentage*—(Optional) Specifies the peak information rate (PIR) bandwidth percentage. Valid values are 1 to 100. • **be** *peak-burst-in-msec*—(Optional) Specifies the peak burst (be) size used by the second token bucket for policing traffic in milliseconds. Valid values are 1 to 2000. • *action*—Specifies the policing command (as shown in Table 4-10) for the action to be applied to the corresponding conforming, exceeding, or violating traffic.

	Command	Purpose
Step 5	Router(config-pmap-c)# **police** {**cir** *cir*} [**bc** *conform-burst*] {**pir** *pir*} [**be** *peak-burst*] [**conform-action** *action* [**exceed-action** *action* [**violate-action** *action*]]]	Configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR), where: • **cir** *cir*—Specifies the CIR at which the first token bucket is updated as a value in bits per second. Valid values are 8000 to 200000000. • **bc** *conform-burst*—(Optional) Specifies the conform burst (bc) size in bytes used by the first token bucket for policing. Valid values are 1000 to 51200000. • **pir** *pir*—Specifies the PIR at which the second token bucket is updated as a value in bits per second. Valid values are 8000 to 200000000. • **be** *peak-burst*—(Optional) Specifies the peak burst (be) size in bytes used by the second token bucket for policing. The size varies according to the interface and platform in use. • *action*—(Optional) Specifies the policing command (as shown in Table 4-10) for the action to be applied to the corresponding conforming, exceeding, or violating traffic.
Step 6	Router(config-pmap-c)# **police flow** {*bits-per-second* [*normal-burst-bytes*] [*maximum-burst-bytes*] [**pir** *peak-rate-bps*]} | [**conform-action** *action*] [**exceed-action** *action*] [**violate-action** *action*]	Configures a microflow policer, where: • *bits-per-second*—Specifies the CIR in bits per second. Valid values are from 32000 to 4000000000 bits per second. • *normal-burst-bytes*—(Optional) Specifies the CIR token bucket size. Valid values are from 1000 to 512000000 bytes. • *maximum-burst-bytes*—(Optional) Specifies the PIR token-bucket size. Valid values are from 1000 to 32000000 bytes. • **pir** *peak-rate-bps*—(Optional) Specifies the PIR in bits per second. Valid values are from 32000 to 4000000000 bits per second. • *action*—Specifies the policing command (as shown in Table 4-10) for the action to be applied to the corresponding conforming, exceeding, or violating traffic.

	Command	Purpose
Step 7	Router(config-pmap-c)# **police flow mask** {**dest-only** | **full-flow** | **src-only**} {*bits-per-second* [*normal-burst-bytes*] [*maximum-burst-bytes*]} [**conform-action** *action*] [**exceed-action** *action*]	Configures a flow mask to be used for policing, where: • **dest-only**—Specifies the destination-only flow mask. • **full-flow**—Specifies the full-flow mask. • **src-only**—Specifies the source-only flow mask. • *bits-per-second*—Specifies the CIR in bits per second. Valid values are from 32000 to 4000000000 bits per second. • *normal-burst-bytes*—(Optional) Specifies the CIR token bucket size. Valid values are from 1000 to 512000000 bytes. • *maximum-burst-bytes*—(Optional) Specifies the PIR token bucket size. Valid values are from 1000 to 32000000 bytes. • *action*—Specifies the policing command (as shown in Table 4-10) for the action to be applied to the corresponding conforming or exceeding traffic.
Step 8	Router(config-pmap-c)# **police aggregate** *name*	Specifies a previously defined aggregate policer name and configures the policy-map class to use the specified *name* of the aggregate policer.

Table 4-10 provides information about which policing actions are supported for SIPs on the Catalyst 6500 Series switch.

Note For restrictions on use of certain marking features with different types of policing actions (conform, exceed, or violate actions), be sure to see the "QoS Traffic Policing Policy Configuration Guidelines" section on page 4-50.

Table 4-10 QoS Policing Action Compatibility by SIP and SPA Combination

Policing Action (set command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Drop the packet (**drop** command)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs—Input interface only.
Set the ATM CLP bit to 1 and transmit (**set-clp-transmit** command)	Supported for all SPAs.	Supported for all SPAs.	Not supported.
Set the inner CoS value and transmit (**set-cos-inner-transmit** command)	Supported in Cisco IOS Release 12.2(33)SXH with bridging features.	Supported in Cisco IOS Release 12.2(33)SXH with bridging features.	Not supported.
Set the Frame Relay DE bit to 1 and transmit (**set-frde-transmit** command)	Supported for all SPAs.	Supported for all SPAs.	Not supported.

Table 4-10 QoS Policing Action Compatibility by SIP and SPA Combination (continued)

Policing Action (set command)	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Set the IP precedence and transmit (**set-prec-transmit** command)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs—Input interface only.[1]
Set the IP DSCP and transmit (**set-dscp-transmit** command)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs—Input interface only.[1]
Set the MPLS EXP bit (0–7) on imposition and transmit (**set-mpls-experimental-imposition-transmit** command	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs.[1]
Set the MPLS EXP bit in the topmost label and transmit (**set-mpls-experimental-topmost-tr ansmit** command)	Supported for all SPAs.	Supported for all SPAs.	Supported for all SPAs.[1]
Transmit all packets without alteration (**transmit** command)	Supported for all SPAs.	Supported for all SPAs	Supported for all SPAs.[1]

[1] Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

Attaching a QoS Traffic Policy to an Interface

Before a traffic policy can be enabled for a class of traffic, it must be configured on an interface. A traffic policy also can be attached to an ATM permanent virtual circuit (PVC) subinterface, Frame Relay data-link connection identifier (DLCI), and Ethernet subinterfaces.

Traffic policies can be applied for traffic coming into an interface (input), and for traffic leaving that interface (output).

Attaching a QoS Traffic Policy for an Input Interface

When you attach a traffic policy to an input interface, the policy is applied to traffic coming into that interface. To attach a traffic policy for an input interface, use the following command beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# service-policy input policy-map-name`	Attaches a traffic policy to the input direction of an interface, where: • *policy-map-name*—Specifies the name of the traffic policy to configure.

Attaching a QoS Traffic Policy to an Output Interface

When you attach a traffic policy to an output interface, the policy is applied to traffic leaving that interface. To attach a traffic policy to an output interface, use the following command beginning in interface configuration mode:

Command	Purpose
`Router(config-if)#` **`service-policy output`** *`policy-map-name`*	Attaches a traffic policy to the output direction of an interface, where: • *policy-map-name*—Specifies the name of the traffic policy to configure.

Configuring Network-Based Application Recognition and Distributed Network-Based Application Recognition

Note Network-Based Application Recognition (NBAR) and Distributed Network-Based Application Recognition (dNBAR) are supported on the Cisco 7600 SIP-200 only.

The purpose of IP quality of service (QoS) is to provide appropriate network resources (bandwidth, delay, jitter, and packet loss) to applications. QoS maximizes the return on investments on network infrastructure by ensuring that mission critical applications get the required performance and noncritical applications do not hamper the performance of critical applications.

IP QoS can be deployed by defining classes or categories of applications. These classes are defined by using various classification techniques available in Cisco IOS software. After these classes are defined and attached to an interface, the desired QoS features, such as marking, congestion management, congestion avoidance, link efficiency mechanisms, or policing and shaping can then be applied to the classified traffic to provide the appropriate network resources amongst the defined classes.

Classification, therefore, is an important first step in configuring QoS in a network infrastructure.

NBAR is a classification engine that recognizes a wide variety of applications, including web-based and other difficult-to-classify protocols that utilize dynamic TCP/UDP port assignments. When an application is recognized and classified by NBAR, a network can invoke services for that specific application. NBAR ensures that network bandwidth is used efficiently by classifying packets and then applying QoS to the classified traffic. Some examples of class-based QoS features that can be used on traffic after the traffic is classified by NBAR include:

- Class-based marking (the **set** command)
- Class-based weighted fair queueing (the **bandwidth** and **queue-limit** commands)
- Low latency queueing (the **priority** command)
- Traffic policing (the **police** command)
- Traffic shaping (the **shape** command)

Note The NBAR feature is used for classifying traffic by protocol. The other class-based QoS features determine how the classified traffic is forwarded and are documented separately from NBAR.

NBAR is not the only method of classifying network traffic so that QoS features can be applied to classified traffic.

For information on the class-based features that can be used to forward NBAR-classified traffic, see the individual feature modules for the particular class-based feature as well as the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Many of the non-NBAR classification options for QoS are documented in the "Modular Quality of Service Command-Line Interface" section of the *Cisco IOS Quality of Service Solutions Configuration Guide*. These commands are configured using the **match** command in class map configuration mode.

NBAR introduces several new classification features that identify applications and protocols from Layer 4 through Layer 7:

- Statically assigned TCP and UDP port numbers
- Non-UDP and non-TCP IP protocols
- Dynamically assigned TCP and UDP port numbers. Classification of such applications requires stateful inspection; that is, the ability to discover the data connections to be classified by parsing the connections where the port assignments are made.
- Sub-port classification or classification based on deep packet inspection; that is, classification by looking deeper into the packet.

NBAR can classify static port protocols. Although access control lists (ACLs) can also be used for this purpose, NBAR is easier to configure and can provide classification statistics that are not available when using ACLs.

NBAR includes a Protocol Discovery feature that provides an easy way to discover application protocols that are transversing an interface. The Protocol Discovery feature discovers any protocol traffic supported by NBAR. Protocol Discovery maintains the following per-protocol statistics for enabled interfaces: total number of input and output packets and bytes, and input and output bit rates. The Protocol Discovery feature captures key statistics associated with each protocol in a network that can be used to define traffic classes and QoS policies for each traffic class.

For specific information about configuring NBAR and dNBAR, refer to the *Network-Based Application Recognition and Distributed Network-Based Application Recognition* feature documentation located at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/dtnbarad.htm

Configuring Hierarchical QoS on a SIP

Table 4-11 provides information about where the hierarchical QoS features for SPA interfaces are supported.

Table 4-11 Hierarchical QoS Feature Compatibility by SIP and SPA Combination

Feature	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600[1]
Hierarchical QoS for EoMPLS VCs	Supported for all SPAs in Cisco IOS Release 12.2(18)SXE and later, and in Cisco IOS Release 12.2(33)SXH.	Supported for all SPAs beginning in Cisco IOS Release 12.2(33)SXH.	Supported for all SPAs in Cisco IOS Release 12.2(18)SXF. [1] Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Hierarchical QoS—Tiered policy maps with parent policy using class-default only on the main interface.	Not applicable.	Supported for all SPAs in Cisco IOS Release 12.2(18)SXF and later.	Supported in Cisco IOS Release 12.2(18)SXF.[1] Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.
Hierarchical QoS—Tiered policy maps with parent policy in user-defined or class-default classes on the main interface.	Supported for all SPAs in Cisco IOS Release 12.2(18)SXF and later, and in Cisco IOS Release 12.2(33)SXH.	Supported for all SPAs in Cisco IOS Release 12.2(33)SXH.	Not supported.

[1] Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

Configuring Hierarchical QoS with Tiered Policy Maps

Hierarchical QoS with tiered policy maps is a configuration where the actions associated with a class contain a queuing action (such as shaping) and a nested service policy, which in itself is a policy map with classes and actions. This hierarchy of the QoS policy map is then translated into a corresponding hierarchy of queues.

Hierarchical QoS with Tiered Policy Maps Configuration Guidelines

When configuring hierarchical QoS with tiered policy maps on a SIP, consider the following guidelines:

- For information about where hierarchical QoS with tiered policy maps is supported, see Table 4-11 on page 4-58.
- You can configure up to three levels of hierarchy within the policy maps.

- The parent policy map has the following restrictions on a main interface:
  - In Cisco IOS Release 12.2(18)SXF and later—Supports the shape queueing action in the default class (class-default) only.
  - In Cisco IOS Release 12.2(33)SXH and later—Supports VLAN or ACL matching, and shape or bandwidth queueing actions in any class, user-defined and class-default.
- When configuring hierarchical QoS for software-based EoMPLS on the Cisco 7600 SIP-600, if you configure **match input vlan** in the parent policy, then you can only configure **match qos-group** in the child policy.
- In hierarchical QoS, you cannot configure just a **set** command in the parent policy. The **set** command works only if you configure other commands in the policy.
- The child policy map supports shape, bandwidth, and WRED QoS features.
- With hierarchical QoS on a subinterface, the parent policy map supports hierarchical QoS using the **shape average** command as a queueing action in the default class (class-default) only.
- If you configure shaping at both the parent policy and the child policy, the traffic is shaped first according to the parameters defined in the parent policy, followed by the parameters of the child policy.
- If you configure service policies at the main interface, subinterface, and VC levels, then the policy applied at the VC level takes precedence over a policy at the interface.
- In a Frame Relay configuration, if you need to define service policies at the interface, subinterface, and PVC at the same time, then you can use a map class.
- For a POS subinterface with a Frame Relay PVC, a service policy can be applied either at the subinterface or at the PVC, but not both.
- Use a hierarchical policy if you want to achieve minimum bandwidth guarantees using CBWFQ with a map class. First, configure a parent policy to shape to the total bandwidth required (use the class-default in Cisco IOS Release 12.2(18)SXF, or a user-defined class in Cisco IOS Release 12.2(33)SXH and later releases). Then, define a child policy using CBWFQ for the minimum bandwidth percentages.
- You can configure hierarchical QoS up to the following limits, according to the current Cisco IOS software limits:
  - Up to 1024 class maps
  - Up to 1024 policy maps
  - Up to 256 classes within a policy map

Configuring Hierarchical QoS for EoMPLS VCs

The Hierarchical Quality of Service (HQoS) for EoMPLS VCs feature extends support for hierarchical, parent and child relationships in QoS policy maps. This feature also provides EoMPLS per-VC QoS for point-to-point VCs.

The new feature adds the ability to match the virtual LAN (VLAN) IDs that were present on a packet when the packet was originally received by the switch. It also supports the ability to match on a QoS group that is set to the same value of the IP precedence or 802.1P class of service (CoS) bits that are received on the incoming interface. This allows service providers to classify traffic easily for all or part of a particular EoMPLS network, as well as to preserve the customer's original differentiated services (DiffServ) QoS values.

In EoMPLS applications, the parent policy map typically specifies the maximum or the minimum bandwidth for a group of specific VCs in an EoMPLS network. Then child policy maps in the policy can implement a different bandwidth or perform other QoS operations (such as traffic shaping) on a subset of the selected VCs.

This feature enables service providers to provide more granular QoS services to their customers. It also gives service providers the ability to preserve customer IP precedence or CoS values in the network.

Note For information about where hierarchical QoS for EoMPLS VCs is supported, see Table 4-11 on page 4-58.

For more information about configuring hierarchical QoS for EoMPLS VCs, refer to the *Optical Services Module Configuration Note* located at the following URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SR_OSM_config/122srosm.html

Configuring PFC QoS on a Cisco 7600 SIP-600

Note Support for the Cisco 7600 SIP-600 was removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

The Cisco 7600 SIP-600 supports most of the same QoS features as those supported by the Policy Feature Card on the Catalyst 6500 Series switch.

This section describes those QoS features that have SIP-specific configuration guidelines. After you review the SIP-specific guidelines described in this document, then refer to the *Cisco 7600 Series Cisco IOS Software Configuration Guide, 12.2SX* located at the following URL:

http://www.cisco.com/en/US/docs/routers/7600/ios/12.2SXF/configuration/guide/swcg.html

PFC QoS Configuration Guidelines for the Cisco 7600 SIP-600

For the Cisco 7600 SIP-600 the following applies:

- Output policing is not supported.

Resetting a SIP

To reset a SIP, use the following command in privileged EXEC configuration mode:

Command	Purpose
`Router# hw-module module slot reset`	Turns power off and on to the SIP in the specified slot, where: • *slot*—Specifies the chassis slot number where the SIP is installed.

Configuration Examples

This section includes the following examples for configuring SIPs installed in a Catalyst 6500 Series switch:

- BCP in Trunk Mode Configuration Example, page 4-61
- QoS Configuration Examples, page 4-62

BCP in Trunk Mode Configuration Example

The following example shows how to configure BCP in trunk mode:

```
! Enter global configuration mode.
!
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
!
! Specify the interface address.
!
Router(config)# interface pos4/1/0
!
! Put the interface in Layer 2 mode for Layer 2 configuration.

Router(config-if)# switchport
%Please shut/no shut POS4/1/0 to bring up BCP
!
! When the switchport command is configured, the interface is automatically configured for
! trunk mode and nonegotiate status.
! Restart the interface to enable BCP.
!
Router(config-if)# shutdown
Router(config-if)# no shutdown
!
! Enable all VLANs for receiving and transmitting traffic on the trunk.
!
Router(config-if)# switchport trunk allowed vlan all
%Internal vlans not available for bridging:1006-1018,1021
```

The following example shows sample output from the **show running-config** command for this configuration. The **switchport mode trunk** and **switchport nonegotiate** commands are automatically generated when the **switchport** command is configured:

```
Router# show running-config interface pos4/1/0
Building configuration...
Current configuration : 191 bytes
!
interface POS4/1/0
switchport
switchport trunk allowed vlan all
switchport mode trunk
switchport nonegotiate
no ip address
encapsulation ppp
clock source internal
end
```

QoS Configuration Examples

This section includes the following QoS configuration examples:

- QoS with Multipoint Bridging Configuration Examples, page 4-62
- Hierarchical QoS with 2-Level Policy Map Configuration Examples, page 4-66

QoS with Multipoint Bridging Configuration Examples

The SIPs and SPAs support a subset of QoS features with MPB configurations.

- For ATM bridging, Frame Relay bridging, MPB, and BCP features on the Cisco 7600 SIP-200 and Cisco 7600 SIP-400, the following matching features are supported on bridged frames in Cisco IOS Release 12.2(33)SXH and later releases:
  - Matching on ATM CLP bit
  - Matching on Frame Relay DE bit
  - Matching on Frame Relay DLCI
  - Matching on inner VLAN
  - Matching on inner COS
  - Matching on IP DSCP (input interface only)
- For ATM bridging, Frame Relay bridging, MPB, and BCP features on the Cisco 7600 SIP-200 and Cisco 7600 SIP-400, the following marking features are supported on bridged frames in Cisco IOS Release 12.2(33)SXH and later releases:
  - Set ATM CLP bit (output interface only)
  - Set Frame Relay DE bit (output interface only)
  - Set inner COS
- For ATM bridging, Frame Relay bridging, MPB, and BCP features on the Cisco 7600 SIP-200 and Cisco 7600 SIP-400, the following marking features with policing are supported on bridged frames in Cisco IOS Release 12.2(33)SXH and later releases:
  - Set inner COS

For more information about configuring QoS on SIPs and SPAs, see the "Configuring QoS Features on a SIP" section on page 4-33.

This section includes the following QoS with MPB configuration examples:

- Matching All Traffic on an Inner VLAN Tag with MPB on SIPs and SPAs Example, page 4-62
- Marking the Inner COS Value with MPB on SIPs and SPAs Example, page 4-63
- Configuring QoS Matching, Shaping, and Marking with MPB on SIPs and SPAs Example, page 4-63
- Setting the Inner CoS Value as a Policing Action for SIPs and SPAs Example, page 4-65

Matching All Traffic on an Inner VLAN Tag with MPB on SIPs and SPAs Example

You can match traffic on an inner VLAN ID of a packet when you are using bridging features on a SPA. The following example shows configuration of a QoS class that filters all bridged traffic for VLAN 100 into a class named vlan-inner-100. An output service policy is then applied to the SPA interface that bridges all outgoing traffic for the vlan-inner-100 class into VLAN 100.

```
! Configure the class maps with your matching criteria.
```

```
!
Router(config)# class-map match-all vlan-inner-100
Router(config-cmap)# match vlan inner 100
!
! Apply the service policy to an input or output bridged interface or VC.
!
Router(config)# interface atm3/0/0
Router(config-if)# pvc 100/100
Router(config-if-atm-vc)# bridge-domain 100 dot1q
Router(config-if-atm-vc)# service-policy output vlan-inner-100
Router(config-if)# end
```

Marking the Inner COS Value with MPB on SIPs and SPAs Example

The following example shows configuration of a QoS class that filters all traffic matching on VLAN 100 into a class named vlan-inner-100. The configuration shows the definition of a policy-map (also named vlan-inner-100) that marks the inner CoS with a value of 3 for traffic in the vlan-inner-100 class. Since marking of the inner CoS value is only supported with bridging features, the configuration also shows the service policy being applied as an output policy to a serial SPA interface that bridges traffic into VLAN 100 using the **bridge-domain** command.

```
! Configure the class maps with your matching criteria.
!
Router(config)# class-map match-all vlan-inner-100
Router(config-cmap)# match vlan inner 100
Router(config-cmap)# exit
!
! Configure the policy map to mark all traffic in a class.
!
Router(config)# policy-map vlan-inner-100
Router(config-pmap)# class vlan-inner-100
Router(config-pmap-c)# set cos-inner 3
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
! Apply the service policy to an input or output bridged interface or VC.
!
Router(config)# interface serial3/0/0
Router(config-if)# no ip address
Router(config_if)# encapsulation ppp
Router(config-if)# bridge-domain 100 dot1q
Router(config-if)# service-policy output vlan-inner-100
Router(config-if)# shutdown
Router(config-if)# no shutdown
Router(config-if)# end
```

Configuring QoS Matching, Shaping, and Marking with MPB on SIPs and SPAs Example

The following example shows a complete QoS configuration of matching, shaping, and marking with MPB on SIPs and SPAs:

```
! Configure the class maps with your matching criteria.
! The following class maps configure matching on the inner VLAN ID.
!
Router(config)# class-map match-all vlan100
Router(config-cmap)# match vlan inner 100
Router(config-cmap)# exit
Router(config)# class-map match-all vlan200
Router(config-cmap)# match vlan inner 200
Router(config-cmap)# exit
Router(config)# class-map match-all vlan300
Router(config-cmap)# match vlan inner 300
Router(config-cmap)# exit
!
! The following class maps configure matching on the inner COS value.
!
Router(config)# class-map match-all cos0
Router(config-cmap)# match cos inner 0
Router(config-cmap)# exit
Router(config)# class-map match-all cos1
Router(config-cmap)# match cos inner 1
Router(config-cmap)# exit
Router(config)# class-map match-all cos2
Router(config-cmap)# match cos inner 2
Router(config-cmap)# exit
Router(config)# class-map match-all cos7
Router(config-cmap)# match cos inner 7
Router(config-cmap)# exit
!
! Configure a policy map for the defined classes.
! The following policies define shaping characteristics for classes
! on different VLANs
!
Router(config)# policy-map vlan100
Router(config-pmap)# class cos1
Router(config-pmap-c)# bandwidth percent 10
Router(config-pmap-c)# exit
Router(config-pmap)# class cos2
Router(config-pmap-c)# bandwidth percent 20
Router(config-pmap-c)# exit
Router(config-pmap)# class cos7
Router(config-pmap-c)#  percent 30
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# policy-map vlan200
Router(config-pmap)# class cos1
Router(config-pmap-c)# bandwidth percent 10
Router(config-pmap-c)# exit
Router(config-pmap)# class cos2
Router(config-pmap-c)# bandwidth percent 20
Router(config-pmap-c)# exit
Router(config-pmap)# class cos7
Router(config-pmap-c)#  percent 30
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
! The following policy map defines criteria for an output interface using MPB
!
Router(config)# policy-map egress_mpb
Router(config-pmap)# class vlan100
Router(config-pmap-c)# bandwidth percent 30
Router(config-pmap-c)# service-policy vlan100
Router(config-pmap-c)# exit
Router(config-pmap)# class vlan200
Router(config-pmap-c)# bandwidth percent 40
Router(config-pmap-c)# service-policy vlan200
!
! The following policy map defines criteria for an input interface using MPB
!
Router(config)# policy-map ingress_mpb
Router(config-pmap)# class vlan100
Router(config-pmap-c)# set cos-inner 5
Router(config-pmap-c)# exit
Router(config-pmap)# class vlan200
Router(config-pmap-c)# set cos-inner 3
!
```

```
! The following policy map defines criteria for an ATM output interface using MPB
! Note: You can only mark ATM CLP on an ATM output interface with MPB
!
Router(config)# policy-map atm_clp
Router(config-pmap)# class cos1
Router(config-pmap-c)# set atm-clp
Router(config-pmap-c)# exit
Router(config-pmap)# class cos2
Router(config-pmap-c)# set atm-clp
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
! Configure an interface for MPB and apply the service policies.
! The following example configures a POS interface in BCP trunk mode and applies two
! different service policies for the output and input traffic on the interface.
!
Router(config)# interface POS3/0/0
Router(config-if)# switchport
Router(config-if)# shutdown
Router(config-if)# no shutdown
Router(config-if)# switchport trunk allowed vlan 100,200,300
Router(config-if)# service-policy output egress_mpb
Router(config-if)# service-policy input ingress_mpb
!
! The following example configures an ATM interface with bridging on VLAN 100
! and applies a service policy for setting the ATM CLP for the output traffic.
!
Router(config)# interface ATM 4/1/0
Router(config-if)# pvc 1/100
Router(config-if-atm-vc)# bridge-domain 100
Router(config-if-atm-vc)# service-policy output atm-clp
```

Setting the Inner CoS Value as a Policing Action for SIPs and SPAs Example

The following example shows configuration of a QoS class that filters all traffic for virtual LAN (VLAN) 100 into a class named vlan-inner-100, and establishes a traffic shaping policy for the vlan-inner-100 class. The service policy limits traffic to a CIR of 20 percent and a PIR of 40 percent, with an conform burst (bc) of 300 ms, and peak burst (be) of 400 ms, and sets the inner CoS value to 3. Because the inner CoS value is only supported with bridging features, the configuration also shows the service policy being applied as an output policy for an ATM SPA interface permanent virtual circuit (PVC) that bridges traffic into VLAN 100 using the **bridge-domain** command.

```
! Configure the class maps with your matching criteria
!
Router(config)# class-map match-all vlan-inner-100
Router(config-cmap)# match vlan inner 100
Router(config-cmap)# exit
!
! Configure the policy map to police all traffic in a class and mark conforming traffic
! (marking traffic whose rate is less than the conform burst)
!
Router(config)# policy-map vlan-inner-100
Router(config-pmap-c)# police cir percent 20 bc 300 ms be 400 ms pir percent 40
conform-action set-cos-inner-transmit 3
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
! Apply the service policy to an input or output bridged interface or VC.
!
Router(config)# interface atm3/0/0
Router(config-if)# pvc 100/100
Router(config-if-atm-vc)# bridge-domain 100 dot1q
Router(config-if-atm-vc)# service-policy output vlan-inner-100
Router(config-if)# end
```

Hierarchical QoS with 2-Level Policy Map Configuration Examples

The following example shows configuration of hierarchical QoS that maps to two levels of hierarchical queues (you can configure up to three levels). The first-level policy (the parent policy) configures the aggregated data rate to be shaped to 1 Mbps for the class-default class. The second-level policy (the child policy) configures the traffic in User-A class for 40 percent of the bandwidth and traffic in User-B class for 60 percent of the bandwidth.

Because this example shows the parent policy applying to the class-default class, it is supported in Cisco IOS Release 12.2(33)SXF and later, as well as in Cisco IOS Release 12.2(33)SXH and later releases.

```
! Configure the class maps with your matching criteria
!
Router(config)# class-map match-any User-A
Router(config-cmap)# match access-group A
Router(config-cmap)# exit
Router(config)# class-map match-any User-B
Router(config-cmap)# match access-group B
Router(config-cmap)# exit
!
! Configure the parent policy for class-default to shape
! all traffic in that class and apply a second-level policy.
!
Router(config)# policy-map parent
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape 1000000
Router(config-pmap-c)# service-policy child
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
! Configure the child policy to allocate different percentages of
! bandwidth by class.
!
Router(config)# policy-map Child
Router(config-pmap)# class User-A
Router(config-pmap-c)# bandwidth percent 40
Router(config-pmap-c)# exit
Router(config-pmap)# class User-B
Router(config-pmap-c)# bandwidth percent 60
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
! Apply the parent service policy to an input or output interface.
!
Router(config)# interface GigabitEthernet 2/0/0
Router(config-if)# service-policy output parent
```

The following example shows configuration of hierarchical QoS that maps to two levels of hierarchical queues, where the parent policy configures average traffic shaping rates on both user-defined classes as well as the class-default class, which is supported in Cisco IOS Release 12.2(33)SXH and later releases. This configuration does not show the corresponding class-map configuration, which is also required to support these policy maps.

```
! Configure the parent policy for user-defined and class-default classes to shape
! traffic in those classes and apply a second-level policy.
!
Router(config)# policy-map parent
Router(config-pmap)# class input-vlan100
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service-policy child-pm
Router(config-pmap-c)# exit
Router(config-pmap)# class input-vlan200
Router(config-pmap-c)# shape average 100000
Router(config-pmap-c)# service-policy child-pm
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 200000
Router(config-pmap-c)# service-policy child-pm
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
! Configure the child policy to allocate different percentages of
! bandwidth by class.
!
Router(config)# policy-map child-pm
Router(config-pmap)# class cos0
Router(config-pmap-c)# bandwidth percent 10
Router(config-pmap-c)# exit
Router(config-pmap)# class cos1
Router(config-pmap-c)# bandwidth percent 10
Router(config-pmap-c)# exit
Router(config-pmap)# exit
!
! Apply the parent service policy to an input or output interface.
!
Router(config)# interface gigabitethernet 2/0/0
Router(config-if)# service-policy output parent-pm
```

CHAPTER 5

Troubleshooting the SIPs and SSC

This chapter describes techniques that you can use to troubleshoot the operation of your SIPs.

It includes the following sections:

- General Troubleshooting Information, page 5-1
- Using the Cisco IOS Event Tracer to Troubleshoot Problems, page 5-2
- Troubleshooting Oversubscription on the Cisco 7600 SIP-400, page 5-3
- Preparing for Online Insertion and Removal of SIPs, SSCs, and SPAs, page 5-3

The first section provides information about basic interface troubleshooting. If you are having a problem with your SPA, use the steps in the "Using the Cisco IOS Event Tracer to Troubleshoot Problems" section to begin your investigation of a possible interface configuration problem.

To perform more advanced troubleshooting, see the other sections in this chapter.

General Troubleshooting Information

This section describes general information for troubleshooting SIPs, SSCs, and SPAs. It includes the following sections:

- Interpreting Console Error Messages, page 5-1
- Using debug Commands, page 5-2
- Using show Commands, page 5-2

Interpreting Console Error Messages

To view the explanations and recommended actions for Catalyst 6500 Series switch error messages, including messages related to Catalyst 6500 Series switch SIPs and SSCs, see the *Catalyst 6500 Series Cisco IOS System Message Guide, 12.2SX*.

System error messages are organized in the documentation according to the particular system facility that produces the messages. The SIP and SSC error messages use the following facility names:

- Cisco 7600 SIP-200—C7600_SIP200
- Cisco 7600 SIP-400—SIP400
- Cisco 7600 SIP-600—SIP600
- Cisco 7600 SSC-400—C7600_SSC400

Using debug Commands

Along with the other **debug** commands supported on the Catalyst 6500 Series switch, you can obtain specific debug information for SIPs and SSCs on the Catalyst 6500 Series switch using the **debug hw-module** privileged exec command.

The **debug hw-module** command is intended for use by Cisco Systems technical support personnel. For more information about the **debug hw-module subslot** command and other **debug** commands, see the *Cisco IOS Debug Command Reference, Release 12.2*.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

For information about other **debug** commands supported on the Catalyst 6500 Series switch, the *Cisco IOS Debug Command Reference, Release 12.2*. For more information about other commands that can be used on a Catalyst 6500 Series switch, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

Using show Commands

There are several **show** commands that you can use to monitor and troubleshoot the SIPs and SSCs on the Catalyst 6500 Series switch. This chapter describes using the **show hw-module slot** commands to perform troubleshooting of your SPA.

For more information about **show** commands to verify and monitor SIPs and SSCs, see Chapter 4, "Configuring the SIPs and SSC."

Using the Cisco IOS Event Tracer to Troubleshoot Problems

Note This feature is intended for use as a software diagnostic tool and should be configured only under the direction of a Cisco Technical Assistance Center (TAC) representative.

The Event Tracer feature provides a binary trace facility for troubleshooting Cisco IOS software. This feature gives Cisco service representatives additional insight into the operation of the Cisco IOS software and can be useful in helping to diagnose problems in the unlikely event of an operating system malfunction or, in the case of redundant systems, Route Processor switchover.

Event tracing works by reading informational messages from specific Cisco IOS software subsystem components that have been preprogrammed to work with event tracing, and by logging messages from those components into system memory. Trace messages stored in memory can be displayed on the screen or saved to a file for later analysis.

The SPAs currently support the "spa" component to trace SPA OIR-related events.

For more information about using the Event Tracer feature, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/evnttrcr.html

Troubleshooting Oversubscription on the Cisco 7600 SIP-400

As of Cisco IOS Release 12.2(18)SXF, when using the Cisco 7600 SIP-400 with the 2-Port Gigabit Ethernet SPA or the 1-Port OC-48c/STM-16 ATM SPA, consider the following oversubscription guidelines:

- The Cisco 7600 SIP-400 only supports installation of one 1-Port OC-48c/STM-16 ATM SPA without any other SPAs installed in the SIP.
- The Cisco 7600 SIP-400 supports installation of up to two 2-Port Gigabit Ethernet SPAs without any other SPAs installed in the SIP.
- The Cisco 7600 SIP-400 supports installation of any combination of OC-3 or OC-12 POS or ATM SPAs, up to a combined ingress bandwidth of OC-48 rates.
- The Cisco 7600 SIP-400 supports installation of any combination of OC-3 or OC-12 POS or ATM SPAs up to a combined ingress bandwidth of OC-24 rates, when installed with a single 2-Port Gigabit Ethernet SPA.

Configurations on the Cisco 7600 SIP-400 with an unsupported aggregate SPA bandwidth greater than OC-48 generates the following error message:

Error Message `SLOT 3: 00:00:05: %SIPSPA-4-MAX_BANDWIDTH: Total SPA bandwidth exceeds line card capacity of 2488 Mbps`

Preparing for Online Insertion and Removal of SIPs, SSCs, and SPAs

The Catalyst 6500 Series switch supports online insertion and removal (OIR) of the SPA interface processor (SIP) or SPA services card (SSC), in addition to each of the shared port adapters (SPAs). Therefore, you can remove a SIP or SSC with its SPAs still intact, or you can remove a SPA independently from the SIP or SSC, leaving the SIP or SSC installed in the switch.

This section includes the following topics on OIR support:

- Preparing for Online Removal of a SIP or SSC, page 5-3
- Verifying Deactivation and Activation of a SIP or SSC, page 5-5
- Preparing for Online Removal of a SPA, page 5-5
- Verifying Deactivation and Activation of a SPA, page 5-7
- Deactivation and Activation Configuration Examples, page 5-7

Note For simplicity, any reference to "SIP" in this section also applies to the SSC.

Preparing for Online Removal of a SIP or SSC

To perform OIR of a SIP or SSC, power down a SIP (which automatically deactivates any installed SPAs) and remove the SIP with the SPAs still intact.

Although graceful deactivation of a SIP by using the **no power enable module** command is preferred, the Catalyst 6500 Series switch does support removal of the SIP without deactivating it first. If you plan to remove a SIP, you can deactivate the SIP first, using the **no power enable module** global configuration command. When you deactivate a SIP using this command, it automatically deactivates each of the SPAs that are installed in that SIP. Therefore, it is not necessary to deactivate each of the SPAs prior to deactivating the SIP.

Either a blank filler plate or a functional SPA should reside in every subslot of a SIP during normal operation.

For more information about the recommended procedures for physical removal of the SIP, refer to the *Cisco 7600 Series Router SIP, SSC, and SPA Hardware Installation Guide*.

Deactivating a SIP or SSC

To deactivate a SIP or SSC and its installed SPAs prior to removal of the SIP, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`no power enable module`** *`slot`*	Shuts down any installed interfaces, and deactivates the SIP in the specified slot, where: • *slot*—Specifies the chassis slot number where the SIP is installed.

For more information about chassis slot numbering, refer to the “Identifying Slots and Subslots for SIPs, SSCs, and SPAs” section in this guide.

Reactivating a SIP or SSC

Once you deactivate a SIP or SSC, whether or not you have performed an OIR, you must use the **power enable module** global configuration command to reactivate the SIP.

If you did not issue a command to deactivate the SPAs installed in a SIP, but you did deactivate the SIP using the **no power enable module** command, then you do not need to reactivate the SPAs after an OIR of the SIP. The installed SPAs automatically reactivate upon reactivation of the SIP in the switch.

For example, if you remove a SIP from the switch to replace it with another SIP, you will reinstall the same SPAs into the new SIP. When you enter the **power enable module** command on the switch, the SPAs will automatically reactivate with the new SIP.

To activate a SIP and its installed SPAs after the SIP has been deactivated, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`power enable module`** *`slot`*	Activates the SIP in the specified slot and its installed SPAs, where: • *slot*—Specifies the chassis slot number where the SIP is installed.

For more information about chassis slot numbering, refer to the “Identifying Slots and Subslots for SIPs, SSCs, and SPAs” section in this guide.

Verifying Deactivation and Activation of a SIP or SSC

To verify the deactivation of a SIP or SSC, enter the **show module** command in privileged EXEC configuration mode. Observe the Status field associated with the SIP that you want to verify.

The following example shows that the Cisco 7600 SIP-400 located in slot 13 is deactivated. This is indicated by its “PwrDown” status.

```
Router# show module 13
Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------

 13    0  4-subslot SPA Interface Processor-400  7600-SIP-400       JAB0851042X

Mod MAC addresses                       Hw    Fw           Sw           Status
--- ---------------------------------- ------ ------------ ------------ -------
 13  00e0.aabb.cc00 to 00e0.aabb.cc3f   0.525 12.2(PP_SPL_ 12.2(PP_SPL_ Ok

Mod Online Diag Status
--- -------------------
 13 PwrDown
```

To verify activation and proper operation of a SIP, enter the **show module** command and observe “Ok” in the Status field as shown in the following example:

```
Router# show module 2
Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------
  2     0 4-subslot SPA Interface Processor-200   7600-SIP-200       JAB074905S1

Mod MAC addresses                       Hw    Fw           Sw           Status
--- ---------------------------------- ------ ------------ ------------ -------
  2  0000.0000.0000 to 0000.0000.003f   0.232 12.2(2004082 12.2(2004082 Ok

Mod Online Diag Status
--- -------------------
  2 Pass
```

Preparing for Online Removal of a SPA

The Catalyst 6500 Series switch supports OIR of a SPA independently of removing the SIP or SSC. This means that a SIP can remain installed in the switch with one SPA remaining active, while you remove another SPA from one of the SIP subslots. If you are not planning to immediately replace a SPA into the SIP, then be sure to install a blank filler plate in the subslot. The SIP should always be fully installed with either functional SPAs or blank filler plates.

The interface configuration is retained (recalled) if a SIP or SPA is removed and then replaced with one of the same type. This is not the case if you replace a Cisco 7600 SIP-200 with a Cisco 7600 SIP-400 or vice versa.

If you are planning to remove a SIP along with its SPAs, then you do not need to follow the instructions in this section. To remove a SIP, see the “Preparing for Online Removal of a SIP or SSC” section on page 5-3.

Deactivating a SPA

Although graceful deactivation of a SPA is preferred using the **hw-module subslot shutdown** command, the Catalyst 6500 Series switch does support removal of the SPA without deactivating it first. Before deactivating a SPA, ensure that the SIP is seated securely into the slot before removing the SPA itself.

Note If you are preparing for an OIR of a SPA, it is not necessary to independently shut down each of the interfaces prior to deactivation of the SPA. The **hw-module subslot shutdown** command automatically stops traffic on the interfaces and deactivates them along with the SPA in preparation for OIR. You also do not need to independently restart any interfaces on a SPA after OIR of a SPA or SIP.

To deactivate a SPA and all of its interfaces prior to removal of the SPA, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`hw-module subslot`** *`slot/subslot`* **`shutdown`** `[`**`powered`** `|` **`unpowered`**`]`	Deactivates the SPA in the specified slot and subslot of the SIP, where: • *slot*—Specifies the chassis slot number where the SIP is installed. • *subslot*—Specifies subslot number on a SIP where a SPA is installed. • **powered**—(Optional) Shuts down the SPA and all of its interfaces, and leaves them in an administratively down state with power enabled. This is the default state. • **unpowered**—(Optional) Shuts down the SPA and all of its interfaces, and leaves them in an administratively down state without power.

For more information about chassis slot and SIP subslot numbering, refer to the "Identifying Slots and Subslots for SIPs, SSCs, and SPAs" section in this guide.

Reactivating a SPA

Note You do not need to reactivate a SPA after an OIR of either the SIP or a SPA if you did not deactivate the SPA prior to removal. If the switch is running, then the SPAs automatically start upon insertion into the SIP or with insertion of a SIP into the switch.

If you deactivate a SPA using the **hw-module subslot shutdown** global configuration command and need to reactivate it without performing an OIR, you need to use the **no hw-module subslot shutdown** global configuration command to reactivate the SPA and its interfaces.

To activate a SPA and its interfaces after the SPA has been deactivated, use the following command in global configuration mode:

Command	Purpose
`Router(config)# **no hw-module subslot** *slot/subslot* **shutdown**`	Activates the SPA and its interfaces in the specified slot and subslot of the SIP, where: • *slot*—Specifies the chassis slot number where the SIP is installed. • *subslot*—Specifies subslot number on a SIP where a SPA is installed.

Verifying Deactivation and Activation of a SPA

When you deactivate a SPA, the corresponding interfaces are also deactivated. This means that these interfaces will no longer appear in the output of the **show interface** command.

To verify the deactivation of a SPA, enter the **show hw-module subslot all oir** command in privileged EXEC configuration mode. Observe the Operational Status field associated with the SPA that you want to verify.

In the following example, the SPA located in subslot 1 of the SIP in slot 2 of the switch is administratively down from the **hw-module subslot shutdown** command:

```
Router# show hw-module subslot all oir
Module          Model               Operational Status
-------------- ------------------ -------------------------
subslot 2/0     SPA-4XOC3-POS       ok
subslot 2/1     SPA-4XOC3-ATM       admin down
```

To verify activation and proper operation of a SPA, enter the **show hw-module subslot all oir** command and observe "ok" in the Operational Status field as shown in the following example:

```
Router# show hw-module subslot all oir
Module          Model               Operational Status
-------------- ------------------ -------------------------
subslot 2/0     SPA-4XOC3-POS       ok
subslot 2/1     SPA-4XOC3-ATM       ok
```

Deactivation and Activation Configuration Examples

This section provides the following examples of deactivating and activating SIPs and SPAs:

- Deactivation of a SIP Configuration Example, page 5-8
- Activation of a SIP Configuration Example, page 5-8
- Deactivation of a SPA Configuration Example, page 5-8
- Activation of a SPA Configuration Example, page 5-8

Deactivation of a SIP Configuration Example

Deactivate a SIP when you want to perform OIR of the SIP. The following example deactivates the SIP that is installed in slot 5 of the switch, its SPAs, and all of the interfaces. The corresponding console messages are shown:

```
Router# configure terminal
Router(config)# no power enable module 5
1w4d: %OIR-6-REMCARD: Card removed from slot 5, interfaces disabled
1w4d: %C6KPWR-SP-4-DISABLED: power to module in slot 5 set off (admin request)
```

Activation of a SIP Configuration Example

Activate a SIP if you have previously deactivated it. If you did not deactivate the SPAs, the SPAs automatically reactivate with reactivation of the SIP.

The following example activates the SIP that is installed in slot 5 of the switch, its SPA, and all of the interfaces (as long as the **hw-module subslot shutdown** command was not issued to also deactivate the SPA):

```
Router# configure terminal
Router(config)# power enable module 5
```

Notice that there are no corresponding console messages shown with the activation. If you reenter the **power enable module** command, a message is displayed indicating that the module is already enabled:

```
Router(config)# power enable module 5
% module is already enabled
```

Deactivation of a SPA Configuration Example

Deactivate a SPA when you want to perform OIR of that SPA. The following example deactivates the SPA (and its interfaces) that is installed in subslot 0 of the SIP located in slot 2 of the switch and removes power to the SPA. Notice that no corresponding console messages are shown.

```
Router# configure terminal
Router(config)# hw-module subslot 2/0 shutdown unpowered
```

Activation of a SPA Configuration Example

Activate a SPA if you have previously deactivated it. If you have not deactivated a SPA and its interfaces during OIR of a SIP, then the SPA is automatically reactivated upon reactivation of the SIP.

The following example activates the SPA that is installed in slot 2 of the switch and all of its interfaces:

```
Router# configure terminal
Router(config)# no hw-module subslot 2/0 shutdown
Router#
```

PART 3

ATM Shared Port Adapters

CHAPTER 6

Overview of the ATM SPAs

This chapter provides an overview of the release history, features, and MIB support for the 1-Port OC-48c/STM-16 ATM SPA, 1-Port OC-12c/STM-4 ATM SPA, and the 2-Port and 4-Port OC-3c/STM-1 ATM SPA. This chapter includes the following sections:

Release History

Release	Modification
12.2(33)SXI	Support was restored for the ATM SPAs.
12.2(33)SXH	Support was temporarily removed for the ATM SPAs.
12.2(18)SXF2	• Support for the "Enhancements to RFC 1483 Spanning Tree Interoperability" feature was added for ATM SPAs on the Cisco 7600 series router and Catalyst 6500 series switch. • Documentation of a workaround for ATM SPA configuration on the Cisco 7600 SIP-200 has been added in Chapter 7, "Configuring the ATM SPAs" to address a Routed Bridge Encapsulation (RBE) limitation where only one remote MAC address is supported.
12.2(18)SXF	Support was introduced for the 1-Port OC-48c/STM-16 ATM SPA on the Cisco 7600 SIP-400 on the Cisco 7600 series router and Catalyst 6500 series switch.
12.2(18)SXE	• Support was introduced for the 2-Port and 4-Port OC-3c/STM-1 ATM SPAs on the Cisco 7600 SIP-200 and Cisco 7600 SIP-400 SPA interface processors (SIPs) on the Cisco 7600 series router and Catalyst 6500 series switch. • Support was introduced for the 1-Port OC-12c/STM-4 ATM SPA on the Cisco 7600 SIP-400 carrier card on the Cisco 7600 series router and Catalyst 6500 series switch.

Overview

The ATM SPAs are single-width, double-height, cross-platform Optical Carrier (OC) ATM adapter cards that provide OC-3c/STM-1c (155.52 Mbps), OC-12c/STM-4c (622.080 Mbps), or OC-48/STM-16 (2488 Mbps) connectivity and can be used in a Catalyst 6500 Series switch. The ATM SPAs come in the following models:

- 2-Port and 4-Port OC-3c/STM-1 ATM SPA (SPA-2XOC3-ATM=, SPA-4XOC3-ATM=)
- 1-Port OC-12c/STM-4 ATM SPA (SPA-1XOC12-ATM=)
- 1-Port OC-48c/STM-16 ATM SPA (SPA-1XOC48-ATM=)

The OC-3c ATM SPAs must be installed in a Cisco 7600 SIP-200 or Cisco 7600 SIP-400 SPA interface processor (SIP) before they can be used in the Catalyst 6500 Series switch. The 1-Port OC-12c/STM-4 ATM SPA and 1-Port OC-48c/STM-16 ATM SPA card must be installed in a Cisco 7600 SIP-400 before it can be used in the Catalyst 6500 Series switch.

You can install the SPA in the SIP carrier before or after you insert the SIP into the switch chassis. This allows you to perform online insertion and removal (OIR) operations either by removing individual SPAs from the SIP, or by removing the entire SIP (and its contained SPAs) from the switch chassis.

The ATM SPAs provide cost-effective wide area networking (WAN) connectivity for service providers across their existing ATM networks. Using a highly modular approach, the SPA and SIP form factors maximize the flexibility of an existing Catalyst 6500 Series switch, allowing service providers to mix and match SPAs to more easily meet evolving port-density and networking media needs.

The ATM SPAs also use small form-factor pluggable (SFP) optical transceivers, giving service providers port-level flexibility for different types of optical media (such as single mode and multimode). Changing the type of optical network involves simply replacing the transceiver, not the SPAs or SIP.

Note A maximum of two ATM SPAs can be installed in each SIP, and these SPAs can be different models (such as 2-Port OC-3c/STM-1 ATM SPA and 1-Port OC-12c/STM-4 ATM SPA). You can also mix SPAs of different types, such as ATM and POS, in a SIP, depending on the space requirements of the SIPs. An exception is that only one 1-Port OC-48c/STM-16 ATM SPA can be installed in a SIP; the other bay should be left empty.

See the following sections for more information about the ATM SPAs:

- ATM Overview, page 6-3
- PVC and SVC Encapsulations, page 6-3
- PVC and SVC Service Classes, page 6-4
- Advanced Quality of Service, page 6-5

ATM Overview

Asynchronous Transfer Mode (ATM) uses cell-switching and multiplexing technology that combines the benefits of circuit switching (constant transmission delay and guaranteed capacity) with those of packet switching (flexibility and efficiency for intermittent traffic). ATM transmits small cells (53 bytes) with minimal overhead (5 bytes of header and checksum, with 48 bytes for data payload), allowing for very quick switching times between the input and output interfaces on a switch.

ATM is a connection-oriented environment, in which each ATM endpoint (or node) must establish a separate connection to the specific endpoints in the ATM network with which it wants to exchange traffic. This connection (or channel) between the two endpoints is called a virtual circuit (VC).

Each VC is uniquely identified by the combination of a virtual path identifier (VPI) and virtual channel identifier (VCI). The VC is treated as a point-to-point mechanism to another switch or host and is capable of supporting bidirectional traffic.

In an ATM network, a VC can be either a permanent virtual circuit (PVC) or a switched virtual circuit (SVC). A network operator must manually configure a PVC, which remains active until it is manually torn down. An SVC is set up and torn down using an ATM signaling mechanism. On the ATM SPAs, this signaling is based on the ATM Forum User-Network Interface (UNI) specification V3.x and V4.0.

PVC and SVC Encapsulations

PVCs and SVCs are configured with an ATM encapsulation type that is based upon the ATM Adaptation Layer (AAL). The following types are supported:

- AAL5CISCOPPP—AAL5 Cisco PPP encapsulation, which is Cisco's proprietary PPP over ATM encapsulation.
- AAL5MUX—ATM Adaptation Layer 5 MUX encapsulation, also known as null encapsulation, that supports a single protocol (IP or IPX).

- AAL5NLPID—(Supported on ATM SPAs in a Cisco 7600 SIP-200 only) AAL5 Network Layer Protocol Identification (NLPID) encapsulation, which allows ATM interfaces to interoperate with High-Speed Serial Interfaces (HSSIs) that are using an ATM data service unit (ADSU) and running ATM-Data Exchange Interface (DXI).
- AAL5SNAP—AAL5 Logical Link Control/Subnetwork Access Protocol (LLC/SNAP) encapsulation, which supports Inverse ARP and incorporates the LLC/SNAP that precedes the protocol datagram. This allows the use of multiple protocols over the same VC, and is particularly well–suited for encapsulating IP packets.

Note The 1-Port OC-48c/STM-16 ATM SPA supports only AAL5MUX and AAL5SNAP encapsulations.

PVC and SVC Service Classes

ATM was designed with built-in quality of service capabilities to allow it to efficiently multiplex different types of traffic over the same links. To accomplish this, each PVC or SVC is configured with a service class that defines the traffic parameters, such as maximum cell rate or burst rate, for the circuit. The following service classes are available in ATM networks:

- Constant Bit Rate (CBR)—The ATM switch transmits ATM cells in a continuous bit-stream that is suitable for real-time traffic, such as voice and video. CBR is typically used for VCs that need a static amount of bandwidth (constant bit rate or average cell rate) that is continuously available for the duration of the active connection. The ATM switch guarantees that a VC with a CBR service class can send cells at the PCR at any time, but the VC is also free to use only part of the allocated bandwidth, or none of the bandwidth, as well.
- Unspecified Bit Rate (UBR)—The ATM switch does not make any quality of service (QoS) commitment at all to the PVC or SVC, but instead uses a best-effort attempt to send the traffic transmitted by the PVC or SVC. UBR typically is the default configuration and is used for non-critical Internet connectivity, including e–mail, file transfers, web browsing, and so forth. The ATM switch enforces a maximum peak cell rate (PCR) for the VC, to prevent the VC from using all bandwidth that is available on the line.
- Unspecified Bit Rate Plus (UBR+)—UBR+ is a special ATM service class developed by Cisco Systems. UBR+ uses MCR (minimum cell rate) along with PCR (peak cell rate). In UBR+, the MCR is a “soft guarantee” of minimum bandwidth. A switch signals the MCR value at call setup time when a switched VC is created. The ATM switch is then responsible for the guarantee of the bandwidth specified in the MCR parameter. A UBR+ VC is a UBR VC for which the MCR is signaled by the switch and guaranteed by the ATM switch. Therefore, UBR+ affects connection admission control and resource allocation on ATM switches. The UBR+ service class is supported only on SVCs for an ATM SPA. It is not supported on PVCs for an ATM SPA.

Note UBR+ is not supported on the 1-Port OC-48c/STM-16 ATM SPA.

- Variable Bit Rate–Non-Real Time (VBR–nrt)—The ATM switch attempts to guarantee a minimum burst size (MBS) and sustained cell rate (SCR) for non-real-time traffic that is bursty in nature, such as database queries or aggregating large volumes of traffic from many different sources. The ATM switch also enforces a maximum peak cell rate (PCR) for the VC, to prevent the VC from using all bandwidth that is available on the line.

- Variable Bit Rate–Real Time (VBR–rt)—The ATM switch guarantees a minimum burst size (MBS) and sustainable cell rate (SCR) for real-time traffic that is bursty in nature, such as voice, video conferencing, and multiplayer gaming. VBR-rt traffic has a higher priority than VBR-nrt traffic, allowing the real-time traffic to preempt the non-real-time traffic, if necessary. The ATM switch also enforces a maximum peak cell rate (PCR) for the VC, to prevent the VC from using all the bandwidth that is available on the line.

Note The ATM SPAs do not support the available bit rate (ABR) service class, which uses a minimum cell rate (MCR).

Advanced Quality of Service

In addition to the integrated QoS capabilities that are provided by the standard ATM service classes, the ATM SPA cards support a number of advanced QoS features. These features include the following:

- Per-VC and Per-VP Traffic Shaping—Enables service providers to control the bandwidth provided at the VC or VP level. (You cannot shape a VC that is part of a shaped VP. We can however enable both VC and VP shaping simultaneously (as long as shaped VCs use a different VPI value than the shaped VP.)
- Layer 3 (IP) QoS at the Per-VC Level—Allows marking and classifying traffic at the IP layer, for each VC, enabling service providers to control the individual traffic flows for a customer, so as to meet the customer’s particular QoS needs. The IP QoS can use the IP type of service (ToS) bits, the RFC 2475 Differentiated Services Code Point (DSCP) bits, and the MPLS EXP bits. WRED, LLQ, CBWFQ, policing, classification, and marking are supported.
- Multiprotocol Label Switching (MPLS)—Allows service providers to provide cost-effective virtual private networks (VPNs) to their customers, while simplifying load balancing and QoS management, without incurring the overhead of extensive Layer 3 routing.
- IP to ATM Mapping—Creates a mapping between the Cell Loss Priority (CLP) bit in ATM cell headers and the IP precedence or IP Differentiated Services Code Point (DSCP) bits.
- VC Bundling—Selects the output VC on the basis of the IP class of service (CoS) bits. (Supported only when using the Cisco 7600 SIP-200 and not the Cisco 7600 SIP-400.)

Note Additional QoS features are expected to be added with each Cisco IOS software release. See the release notes for each release for additional features that might be supported and for the restrictions that might affect existing features.

Supported Features

This section provides a list of some of the primary features supported by the ATM hardware and software:

- SIP-Dependent Features, page 6-6
- Basic Features, page 6-6
- SONET/SDH Error, Alarm, and Performance Monitoring, page 6-7
- Layer 2 Features, page 6-8

- Layer 3 Features, page 6-9
- High Availability Features, page 6-10
- Enhancements to RFC 1483 Spanning Tree Interoperability, page 6-10
- Supported Supervisor Engines and Line Cards, page 6-11
- Interoperability Problems, page 6-11
- BPDU Packet Formats, page 6-12

SIP-Dependent Features

Most features for the ATM SPAs are supported on both the Cisco 7600 SIP-200 and Cisco 7600 SIP-400, but some features are supported only on a particular model of SIP. Table 6-1 lists the features that are supported on only one model of SIP. Any supported features for the ATM SPAs that are not listed in this table are supported on both SIPs.

Table 6-1 SIP-Dependent Feature Support

Feature	Supported on Cisco 7600 SIP-200	Supported on Cisco 7600 SIP-400
AAL5NLPID encapsulation and Routed-NLPID-PDUs	Yes	No
ATM VC Access Trunk Emulation (multi-VLAN to VC)	Yes	No
Bridging of Routed Encapsulations (BRE)	Yes	No
Frame Relay to ATM (FR-ATM) internetworking	No	No
Network-Based Application Recognition (NBAR)	Yes	No
RFC-1483 ATM Half-Bridging and Routed Bridged Encapsulation (RBE)	Yes	No
VC Bundling (Selects the output VC on the basis of the IP CoS bits)	Yes	No
RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, Multipoint Bridging (MPB) (also known as multi-VC to VLAN)on the 2-Port and 4-Port OC-3c/STM-1c ATM SPA	Yes	No
Aggregate WRED	Yes	Yes

Basic Features

- Bellcore GR-253-CORE SONET/SDH compliance (ITU-T G.707, G.783, G.957, G.958)
- Interface-compatible with other Cisco ATM adapters

Note The ATM SPA is functionally similar to other ATM port adapters on the Catalyst 6500 Series switch, but because it is a different card type, the configuration for the slot is lost when you replace an existing ATM port adapter with an ATM SPA in a SIP.

- Supports both permanent virtual circuits (PVCs) and switched virtual circuits (SVCs)

- An absolute maximum of 16,384 (16 K) configured VCs per ATM SPA (4,096 [4 K] per interface) with the following recommended limitations:
  - On a Cisco 7600 SIP-400, 8000 PVCs are supported on multipoint subinterfaces. The limit of 16,384 PVCs only applies to the Cisco 7600 SIP-200.
  - A recommended maximum number of 2,048 PVCs on all point-to-point subinterfaces for all ATM SPAs in a SIP.
  - A recommended maximum number of 16,380 PVCs on all multipoint subinterfaces for all ATM SPAs in a SIP, and a recommended maximum number of 200 PVCs per each individual multipoint subinterface.
  - A recommended maximum number of 400 SVCs for all ATM SPAs in a SIP.
  - A recommended maximum number of 1,024 PVCs or 400 SVCs using service policies for all ATM SPAs in a SIP.
- Up to 4,096 simultaneous segmentations and reassemblies (SARs) per interface
- Supports a maximum number of 200 PVCs or SVCs using Link Fragmentation and Interleaving (LFI) for all ATM SPAs (or other ATM modules) in a Catalyst 6500 Series switch.
- A maximum number of 1000 PVCs or 400 SVCs configured with MQC policy maps.
- Up to 1,000 maximum virtual templates per switch
- ATM adaptation layer 5 (AAL5) for data traffic
- Hardware switching of multicast packets for point-to-point subinterfaces
- SONET/SDH (software selectable) optical fiber (2-Port and 4-Port OC-3c/STM-1 ATM SPA, 1-Port OC-48c/STM-16 ATM SPA, or 1-Port OC-12c/STM-4 ATM SPA), depending on the model of ATM SPA.
- Uses small form-factor pluggable (SFP) optical transceivers, allowing the same ATM SPA hardware to support multimode (MM), single-mode intermediate (SMI), or single-mode long (SML) reach, depending on the capabilities of the SPA.
- ATM section, line, and path alarm indication signal (AIS) cells, including support for F4 and F5 flows, loopback, and remote defect indication (RDI)
- Operation, Administration, and Maintenance (OAM) cells
- Online insertion and removal (OIR) of individual ATM SPAs from the SIP, as well as OIR of the SIPs with ATM SPAs installed

SONET/SDH Error, Alarm, and Performance Monitoring

- Fiber removed and reinserted
- Signal failure bit error rate (SF-BER)
- Signal degrade bit error rate (SD-BER)
- Signal label payload construction (C2)
- Path trace byte (J1)
- Section Diagnostics:
  - Loss of signal (SLOS)
  - Loss of frame (SLOF)
  - Error counts for B1

  - Threshold crossing alarms (TCA) for B1 (B1-TCA)
- Line Diagnostics:
  - Line alarm indication signal (LAIS)
  - Line remote defect indication (LRDI)
  - Line remote error indication (LREI)
  - Error counts for B2
  - Threshold crossing alarms for B2 (B2-TCA)
- Path Diagnostics:
  - Path alarm indication signal (PAIS)
  - Path remote defect indication (PRDI)
  - Path remote error indication (PREI)
  - Error counts for B3
  - Threshold crossing alarms for B3 (B3-TCA)
  - Loss of pointer (PLOP)
  - New pointer events (NEWPTR)
  - Positive stuffing event (PSE)
  - Negative stuffing event (NSE)
- The following loopback tests are supported:
  - Network (line) loopback
  - Internal (diagnostic) loopback
- Supported SONET/SDH synchronization:
  - Local (internal) timing (for inter-switch connections over dark fiber or WDM equipment)
  - Loop (line) timing (for connecting to SONET/SDH equipment)
  - +/– 4.6 ppm clock accuracy over full operating temperature

Layer 2 Features

- Supports the following encapsulation types:
  - AAL5SNAP (LLC/SNAP)
  - LLC encapsulated Bridged protocol
  - AAL5MUX (VC multiplexing)
  - AAL5NLPID and Routed-NLPID-PDUs (ATM SPAs in a Cisco 7600 SIP-200 only)
  - AAL5CISCOPPP
- Supports the following ATM traffic classes and per-VC traffic shaping modes:
  - Constant bit rate (CBR) with peak rate
  - Unspecified bit rate (UBR) with peak cell rate (PCR)
  - Non-real-time variable bit rate (VBR-nrt)
  - Variable bit rate real-time (VBR-rt)

  - Unspecified bit rate plus (UBR+) on SVCs

> **Note** ATM shaping is supported, but class queue-based shaping is not.

- ATM point-to-point and multipoint connections
- Explicit Forward Congestion Indication (EFCI) bit in the ATM cell header
- Frame Relay to ATM (FR-ATM) internetworking (ATM SPAs in a Cisco 7600 SIP-200 only)
- Integrated Local Management Interface (ILMI) operation, including keepalive, PVC discovery, and address registration and deregistration
- Link Fragmentation and Interleaving (LFI) performed in hardware
- VC–to–VC local switching and cell relay
- RFC 1755, *ATM Signaling Support for IP over ATM*
- ATM User-Network Interface (UNI) signalling 3.0, 3.1, and 4.0 only
- RFC 2225, *Classical IP and ARP over ATM* (obsoletes RFC 1577)
- Unspecified bit rate plus (UBR+) traffic service class on SVCs

Layer 3 Features

- ATM VC Access Trunk Emulation (multi-VLAN to VC) (ATM SPAs in a Cisco 7600 SIP-200 only)
- ATM over MPLS (AToM) in AAL5 mode (except for AToM cell packing)
- ATM over MPLS (AToM) in AAL5/AAL0 VC mode
- Bridging of Routed Encapsulations (BRE) (ATM SPAs in a Cisco 7600 SIP-200 only)
- Distributed Link Fragmentation and Interleaving (dLFI) for ATM (dLFI packet counters are supported, but dLFI byte counters are not supported)
- LFI+DCRTP
- Network-Based Application Recognition (NBAR) (ATM SPAs in a Cisco 7600 SIP-200 only)
- No limitation on the maximum number of VCs per VPI, up to the maximum number of 4,096 total VCs per interface (so there is no need to configure this limit using the **atm vc-per-vp** command, which is required on other ATM port adapters)
- OAM flow connectivity using OAM ping for segment or end-to-end loopback
- PVC multicast (PIM dense and sparse modes)
- Quality of service (QoS):
  - Policing
  - IP-to-ATM class of service (IP precedence and DSCP)
  - Per-VC class-based weighted fair queueing (CBWFQ)
  - Per-VC Layer 3 queuing
  - VC Bundling (Cisco 7600 SIP-200 only)
  - Weighted Random Early Detection (WRED)
  - Aggregate WRED

- RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*:
  - Routed Bridge Encapsulation (RBE) (ATM SPAs in a Cisco 7600 SIP-200 only)
  - Half-bridging (ATM SPAs in a Cisco 7600 SIP-200 only)
  - PVC bridging (full-bridging) is supported on Cisco 7600 SIP-200 and Cisco 7600 SIP-400
- Supports oversubscription by default
- Routing protocols:
  - Border Gateway Protocol (BGP)
  - Enhanced Interior Gateway Routing Protocol (EIGRP)
  - Interior Gateway Routing Protocol (IGRP)
  - Integrated Intermediate System-to-Intermediate System (IS-IS)
  - Open Shortest Path First (OSPF)
  - Routing Information Protocol version 1 and version 2 (RIPv1 and RIPv2)

High Availability Features

- 1+1 Automatic Protection Switching (APS) redundancy (PVC circuits only)
- Route Processor Redundancy (RPR)
- RPR Plus (RPR+)
- OSPF Nonstop Forwarding (NSF)
- Stateful Switchover (SSO)

Enhancements to RFC 1483 Spanning Tree Interoperability

This section describes an interoperability feature for the various spanning tree implementations across 1483 Bridge Mode ATM PVCs. Historically, vendors have not implemented spanning tree across RFC 1483 encapsulation consistently. Some Cisco IOS releases also may not support the full range of spanning-tree options. This feature addresses some of the practical challenges of interworking common variations of spanning tree over RFC 1483 Bridge Mode encapsulation.

Note This feature set is only supported on RFC 1483 Bridge Mode ATM permanent virtual circuits (PVCs).

The following are basic spanning tree terms:

- *IEEE 802.1D* is a standard for interconnecting LANs through media access control (MAC) bridges. IEEE 802.1D uses the Spanning-Tree Protocol to eliminate loops in the bridge topology, which cause broadcast storms.
- *Spanning Tree Protocol (STP)* as defined in IEEE 802.1D is a link-management protocol that provides path redundancy while preventing undesirable loops in the network. An IEEE 802.1D spanning tree makes it possible to have one spanning tree instance for the whole switch, regardless of the number of VLANs configured on the switch.

- *Bridge Protocol Data Unit (BPDU)* is the generic name for the frame used by the various spanning-tree implementations. The Spanning Tree Protocol uses the BPDU information 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.
- *Per VLAN Spanning Tree (PVST)* is a Cisco proprietary protocol that allows a Cisco device to support multiple spanning tree topologies on a per-VLAN basis. PVST uses the BPDUs defined in IEEE 802.1D (see Figure 6-2 on page 6-12), but instead of one STP instance per switch, there is one STP instance per VLAN.
- *PVST+* is a Cisco proprietary protocol that creates one STP instance per VLAN (as in PVST). However, PVST+ enhances PVST and uses Cisco proprietary BPDUs with a special 802.2 Subnetwork Access Protocol (SNAP) Organizational Unique Identifier (OUI)[1] (see Figure 6-2 on page 6-12) instead of the standard IEEE 802.1D frame format used by PVST. PVST+ BPDUs are also known as Simple Symmetric Transmission Protocol (SSTP) BPDUs.

Note RFC 1483 is referenced throughout this section, although it has been superseded by RFC 2684.

Supported Supervisor Engines and Line Cards

The Cisco 7600 series router supports PVST to PVST+ BPDU interoperability with the following line card:

- Cisco 7600 SIP-200

Interoperability Problems

The current interoperability problems can be summarized as follows:

- When transmitting STP BPDUs, many vendors' implementations of ATM-to-Ethernet bridging are not fully compliant with the specifications of RFC 1483, Appendix B. The most common variation of the standard is to use an ATM Common Part Convergence Sublayer (CPCS) SNAP protocol data unit (PDU) with OUI: 00-80-C2 and PID: 00-07. Appendix B reserved this OUI/PID combination for generic Ethernet frames without BPDUs. Appendix B specifies OUI: 00-80-C2 and protocol identifier (PID): 00-0E for frames with BPDU contents.
- There are several varieties of the Spanning-TreeProtocol used by Cisco products on ATM interfaces. The Catalyst 5000 series supports only PVST on ATM interfaces. The Cisco 7600 router and Catalyst 6500 series switches support only PVST+ on ATM interfaces. Most other Cisco routers implement classic IEEE 802.1D on ATM interfaces.

  When the Cisco 7600 series router and the Catalyst 6500 series switch first implemented 1483 Bridging (on Cisco IOS Release 12.1E) on the Cisco 7600 FlexWAN module, the platform used OUI: 00-80-C2 and PID: 00-0E to maximize interoperability with all other Cisco IOS products.

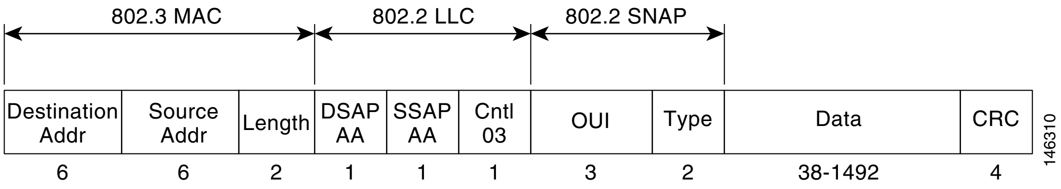
  However, there are so many implementations that do not send PVST or IEEE 802.1D BPDUs with PID: 00-0E that the Cisco 7600 series and the Catalyst 6500 series reverted to the more common implementation of RFC 1483 (with PID: 00-07) in Cisco IOS 12.2SX. This spanning tree interoperability feature provides the option of encapsulating BPDUs across RFC 1483 with either PID: 00-07 or PID: 00-0E.

[1] The Organizational Unique Identifier (OUI) portion of the MAC address often identifies the vendor of the upper layer protocol or the manufacturer of the Ethernet adapter. The OUI value of 00-00-0C identifies Cisco Systems as the manufacturer of the Ethernet adapter.

BPDU Packet Formats

This section describes the various BPDU packet formats. Figure 6-1 shows the generic IEEE 802.2/802.3 frame format, which is used by PVST+,—but is not used by PVST.

Figure 6-1 IEEE 802.2/802.3 SNAP Encapsulation Frame Format



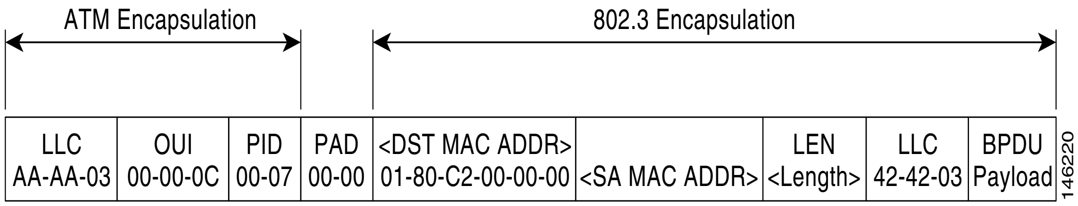
In an Ethernet SNAP frame, the SSAP and DSAP fields are always set to AA. These codes identify it as a SNAP frame. The Control field always has a value of 03, which specifies connectionless logical link control (LLC) services.

The Type field identifies the upper layer protocol to which data should be passed. For example, a Type field of hex 0800 represents IP, while a value of 8137 indicates that data is meant for IPX.

Catalyst 5000 PVST BPDU Packet Format

The Catalyst 5000 series switches send and receive BPDUs in PVST format on ATM interfaces (see Figure 6-2).

Figure 6-2 BPDU PVST Frame Format Used by the Catalyst 5000 Switch



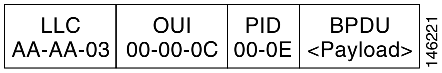
- BPDUs sent by the Catalyst 5000 switch use a PID of 0x00-07, which does not comply with RFC 1483. The Cisco 7600 series router also has the ability to send BPDUs in this data format.
- The PAD portion of the ATM encapsulation varies from 0 to 47 bytes in length to ensure complete ATM cell payloads.
- By using the **bridge-domain** command's **ignore-bpdu-pid** optional keyword, the Catalyst 5000 switch sends this frame by default.
- The Catalyst 5000 switch cannot accept the PVST+ BPDUs and blocks the ATM port, giving the following error message:

```
%SPANTREE-2-RX_1QNON1QTRUNK: Rcved 1Q-BPDU on non-1Q-trun port 6/1 vlan 10
%SPANTREE-2-RX_BLKPORTPVID: Block 6/1 on rcving vlan 10 for inc peer vlan 0
```

Cisco 7200 and Cisco 7500 Routers IEEE 802.1D BPDU Frame Format

Figure 6-3 shows the Cisco 7200 and Cisco 7500 series routers IEEE 802.1D BPDU frame format:

Figure 6-3 Frame Format for the Cisco 7200 and Cisco 7500 Routers IEEE 802.1D BPDU

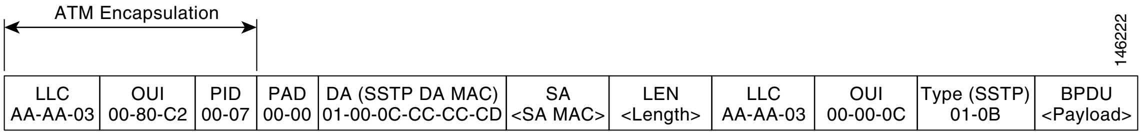


LLC	OUI	PID	BPDU
AA-AA-03	00-00-0C	00-0E	<Payload>

Cisco 7600 Router PVST+ BPDU Frame Format

The Cisco 7600 series router PVST+ BPDU packet format is shown in Figure 6-4. These BPDUs are not IEEE 802.1D BPDUs, but Cisco proprietary SSTP BPDUs.

Figure 6-4 Cisco 7600 Router PVST+ BPDU Frame Format (1483 Bridge Mode)

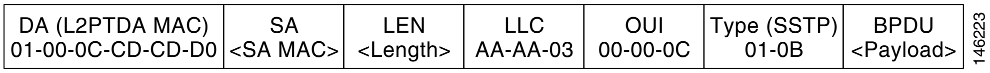


LLC	OUI	PID	PAD	DA (SSTP DA MAC)	SA	LEN	LLC	OUI	Type (SSTP)	BPDU
AA-AA-03	00-80-C2	00-07	00-00	01-00-0C-CC-CC-CD	<SA MAC>	<Length>	AA-AA-03	00-00-0C	01-0B	<Payload>

Cisco L2PT BPDU Frame Format

Figure 6-5 shows the Cisco Layer 2 Protocol Tunneling (L2PT) BPDU SNAP frame format.

Figure 6-5 L2PT BPDU SNAP Frame Format



DA (L2PTDA MAC)	SA	LEN	LLC	OUI	Type (SSTP)	BPDU
01-00-0C-CD-CD-D0	<SA MAC>	<Length>	AA-AA-03	00-00-0C	01-0B	<Payload>

Unsupported Features

- The following high availability features are not supported:
  - APS N+1 redundancy is not supported
  - APS redundancy is not supported on SVCs
  - APS reflector mode (**aps reflector** interface configuration command) is not supported
- The **atm bridge-enable** command, which was used in previous releases on other ATM interfaces to enable multipoint bridging on PVCs, is not supported on ATM SPA interfaces. Instead, use the **bridge** option with the **encapsulation** command to enable RFC 1483 half-bridging on PVCs. See the "Configuring ATM Routed Bridge Encapsulation" section on page 7-20.
- PVC autoprovisioning (**create on-demand** VC class configuration command) is not supported.
- Creating SVCs with UNI signalling 4.1 is not supported (UNI signalling 3.0, 3.1, and 4.0 are supported).
- Enhanced Remote Defect Indication–Path (ERDI-P) is not supported.

- Fast Re-Route (FRR) over ATM is not supported.
- LAN Emulation (LANE) is not supported.
- Multicast SVCs are not supported.
- Available Bit Rate (ABR) traffic service class is not supported.
- Unspecified bit rate plus (UBR+) traffic service class is not supported on PVCs.
- VP–to–VP local switching and cell relay are not supported.

Prerequisites

- The 2-Port and 4-Port OC-3c/STM-1 ATM SPAs must use either the Cisco 7600 SIP-200 or Cisco 7600 SIP-400.
- The 1-Port OC-12c/STM-4 ATM SPA must use the Cisco 7600 SIP-400.
- The 1-Port OC-48c/STM-16 ATM SPA must use the Cisco 7600 SIP-400.
- The Cisco 7600 SIP-200 requires a Catalyst 6500 Series switch using a SUP-720 3B and above processor that is running Cisco IOS Release 12.2(18)SXE or later release.
- The Cisco 7600 SIP-400 requires a Catalyst 6500 Series switch using a SUP-720 processor that is running Cisco IOS Release 12.2(18)SXE or later release.
- Before beginning to configure the ATM SPA, have the following information available:
  - Protocols you plan to route on the new interfaces.
  - IP addresses for all ports on the new interfaces, including subinterfaces.
  - Bridging encapsulations you plan to use.

Restrictions

Note For other SIP-specific restrictions, see the "Restrictions" section on page 3-15.

- The 1-Port OC-48c/STM-16 ATM SPA does not support the following features: AToM, BRE, LFI, RBE, SVCs, UBR+, RFC 2225 (formerly RFC 1577), or bridging.
- The ATM SPAs in the Catalyst 6500 Series switch do not support APS reflector and reflector channel modes. (These modes require a facing PTE, which is typically a Cisco ATM switch.)
- The ATM SPA is functionally similar to other ATM port adapters on the Catalyst 6500 Series switch, such as the PA-A3, but it is a different card type, so the slot's previous configuration is lost when you replace an existing ATM port adapter with an ATM SPA.
- The following restrictions apply to the operation of QoS on the ATM SPAs:
  - The ATM SPAs do not support bandwidth-limited priority queueing, but support only strict priority policy maps (that is, the **priority** command without any parameters).
  - A maximum of one **priority** command is supported in a policy map.
  - You cannot use the **match input interface** command in policy maps and class maps that are being used for ATM SPAs.

- Hierarchical traffic shaping (traffic shaping on both the VC and VP for a circuit) is not supported. Traffic shaping can be configured only on the VC or on the VP, but not both.
- ATM (Layer 2) output shaping is supported, but IP (Layer 3) shaping on an output (egress) interface is not supported. In particular, this means that you cannot use any **shape** class-map configuration commands in policy maps that are being used in the output direction. This includes the **shape adaptive**, **shape average**, **shape fecn-adapt**, and **shape peak** commands.
- The ATM SPA interfaces support a maximum of six configured precedences (using the **random-detect aggregate** command) in each class map in a policy map. The maximum number of configurable subclass groups is 7.

- For best performance, we recommend the following maximums:
  - A maximum number of 2,048 PVCs on all point-to-point subinterfaces for all ATM SPAs in a SIP.
  - A maximum number of 16,380 PVCs on all multipoint subinterfaces for all ATM SPAs in a SIP.
  - A maximum number of 400 SVCs for all ATM SPAs in a SIP carrier card.
  - A maximum number of 1024 PVCs or SVCs s using service policies for all ATM SPAs in a switch.
  - A maximum number of 200 PVCs or SVCs using Link Fragmentation and Interleaving (LFI) for all ATM SPAs in a switch.
  - A maximum number of 200 PVCs on each multipoint subinterface being used on an ATM SPA.

Note These limits are flexible and depend on all factors that affect performance in the switch, such as processor card, type of traffic, and so on.

- In the default configuration of the transmit path trace buffer, the ATM SPA does not support automatic updates of remote host name and IP address (as displayed by the **show controllers atm** command). This information is updated only when the interface is shut down and reactivated (using the **shutdown** and **no shutdown** commands). Information for the received path trace buffer, however, is automatically updated.
- The **show ppp multilink** command displays only the packet counters, and not byte counters, for a dLFI configuration on an ATM SPA interface.

Supported MIBs

The following MIBs are supported in Cisco IOS Release 12.2(18)SXE and later releases for the ATM SPAs on the Catalyst 6500 Series switch.

Common MIBs

- ENTITY-MIB
- IF-MIB
- MIB-II

Cisco-Specific Common MIBs

- CISCO-ENTITY-EXT-MIB
- OLD-CISCO-CHASSIS-MIB

- CISCO-CLASS-BASED-QOS-MIB
- CISCO-ENTITY-FRU-CONTROL-MIB
- CISCO-ENTITY-ASSET-MIB
- CISCO-ENTITY-SENSOR-MIB
- CISCO-MQC-MIB

ATM Industry MIBs

- ATM-MIB (RFC 2515)
- ATM-ACCOUNTING-INFORMATION-MIB (RFC 2512)
- SONET-MIB

Cisco-Specific ATM MIBs

- CISCO-ATM-EXT-MIB
- CISCO-ATM-PVC-MIB
- CISCO-AAL5-MIB
- CISCO-CLASS-BASED-QOS-MIB
- CISCO-IETF-ATM2-PVCTRAP-MIB
- CISCO-MQC-MIB
- CISCO-SONET-MIB

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

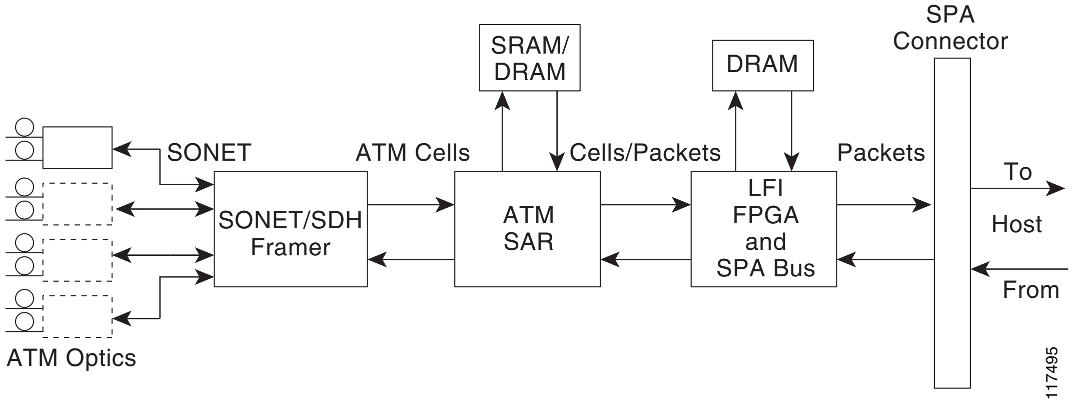
To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

SPA Architecture

This section provides an overview of the data path for the ATM SPAs, for use in troubleshooting and monitoring. Figure 6-6 shows the data path for ATM traffic as it travels between the ATM optical connectors on the front panel of the ATM SPA to the backplane connector that connects the SPA to the SIP.

Figure 6-6 ATM SPA Data Architecture



Path of Cells in the Ingress Direction

The following steps describe the path of an ingress cell as it is received from the ATM network and converted to a data packet before transmission through the SIP to the switch's processors for switching, routing, or further processing:

1. The SONET/SDH framer device receives incoming cells on a per-port basis from the SPA's optical circuitry. (The ATM SPA supports 1, 2, or 4 optical ports, depending on the model of SPA.)
2. The SONET/SDH framer removes the SONET overhead information, performs any necessary clock and data recovery, and processes any SONET/SDH alarms that might be present. The framer then extracts the 53-byte ATM cells from the data stream and forwards each cell to the ATM segmentation and re-assembly (SAR) engine.
3. The SAR engine receives the cells from the framer and reassembles them into the original packets, temporarily storing them in a per-port receive buffer until they can be forwarded to the LFI FPGA. The SAR engine discards any packets that have been corrupted in transit.
4. The LFI FPGA receives the packets from the SAR engine and forwards them to the host processor for further routing, switching, or additional processing. The FPGA also performs LFI reassembly as needed, and collects the traffic statistics for the packets that it passes.

Path of Packets in the Egress Direction

The following steps describe the path of an egress packet as the SPA receives it from the switch through the SIP and converts it to ATM cells for transmission on the ATM network:

1. The LFI FPGA receives the packets from the host processor and stores them in its packet buffers until the SAR engine is ready to receive them. The FPGA also performs any necessary LFI processing on the packets before forwarding them to the SAR engine. The FPGA also collects the traffic statistics for the packets that it passes.
2. The SAR engine receives the packets from the FPGA and supports multiple CBWFQ queues to store the packets until they can be fully segmented. The SAR engine performs the necessary WRED queue admission and CBWFQ QoS traffic scheduling on its queues before segmenting the packets into ATM cells and shaping the cells into the SONET/SDH framer.

3. The SONET/SDH framer receives the packets from the SAR engine and inserts each cell into the SONET data stream, adding the necessary clocking, SONET overhead, and alarm information. The framer then outputs the data stream out the appropriate optical port.
4. The optical port conveys the optical data onto the physical layer of the ATM network.

Displaying the SPA Hardware Type

To verify the SPA hardware type that is installed in your Catalyst 6500 Series switch, use the **show interfaces** or **show diagbus** commands. A number of other **show** commands also provide information about the SPA hardware.

Table 6-2 shows the hardware description that appears in the **show** command output for each type of ATM SPA that is supported on the Catalyst 6500 Series switch.

Table 6-2 ATM SPA Hardware Descriptions in show Commands

SPA	Description in show interfaces Command	Description in show diagbus Command
SPA-2XOC3-ATM	Hardware is SPA-2XOC3-ATM	SPA-2XOC3-ATM (0x046E)
SPA-4XOC3-ATM	Hardware is SPA-4XOC3-ATM	SPA-4XOC3-ATM (0x3E1)
SPA-1XOC12-ATM	Hardware is SPA-1XOC12-ATM	SPA-1XOC12-ATM (0x03E5)
SPA-1XOC48-ATM	Hardware is SPA-1XOC48-ATM	SPA-1XOC48-ATM (0x3E6)

Example of the show interfaces Command

The following example shows output from the **show interfaces atm** command on a Catalyst 6500 Series switch with an ATM SPA installed in the first subslot of a SIP that is installed in slot 5:

```
Router# show interfaces atm 5/0/0

ATM5/0/0 is up, line protocol is up
  Hardware is SPA-4XOC3-ATM, address is 000d.2959.d780 (bia 000d.2959.d78a)
  MTU 4470 bytes, sub MTU 4470, BW 149760 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM, loopback not set
  Encapsulation(s): AAL5
  4095 maximum active VCs, 1 current VCCs
  VC idle disconnect time: 300 seconds
  0 carrier transitions
  Last input 00:00:09, output 00:00:09, output hang never
  Last clearing of "show interface" counters 00:01:26
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     5 packets input, 540 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     5 packets output, 720 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Note The value for "packets output" in the default version of the **show interfaces atm** command includes the bytes used for ATM AAL5 padding, trailer and ATM cell header. To see the packet count without the padding, header, and trailer information, use the **show interfaces atm statistics** or **show atm pvc** commands.

Example of the show diagbus Command

The following example shows output from the **show diagbus** command on a Catalyst 6500 Series switch with two ATM SPAs installed in a Cisco 7600 SIP-400 that is installed in slot 4:

```
Router# show diagbus 4

Slot 4: Logical_index 8
        4-adapter SIP-400 controller
        Board is analyzed ipc ready
        HW rev 0.300, board revision 08
        Serial Number:  Part number: 73-8272-03

        Slot database information:
        Flags: 0x2004   Insertion time: 0x1961C (01:16:54 ago)

        Controller Memory Size:
                384 MBytes CPU Memory
                128 MBytes Packet Memory
                512 MBytes Total on Board SDRAM
        IOS (tm) cwlc Software (sip1-DW-M), Released Version 12.2(17)SX [BLD-sipedon2 107]

        SPA Information:
        subslot 4/0: SPA-4XOC3-ATM (0x3E1), status: ok
        subslot 4/1: SPA-1XOC12-ATM (0x3E5), status: ok
```

Example of the show controllers Command

The following example shows output from the **show controllers atm** command on a Catalyst 6500 Series switch with an ATM SPAs installed in the second subslot of a SIP that is installed in slot 5:

```
Router# show controllers atm 5/1/0

Interface ATM5/1/0 (SPA-4XOC3-ATM[4/0]) is up
 Framing mode: SONET OC3 STS-3c

SONET Subblock:
SECTION
  LOF = 0          LOS     = 0                            BIP(B1) = 603
LINE
  AIS = 0          RDI     = 2          FEBE = 2332       BIP(B2) = 1018
PATH
  AIS = 0          RDI     = 1          FEBE = 28         BIP(B3) = 228
  LOP = 0          NEWPTR = 0           PSE  = 1          NSE     = 2

Active Defects: None
Active Alarms:  None
Alarm reporting enabled for: SF SLOS SLOF B1-TCA B2-TCA PLOP B3-TCA

ATM framing errors:
  HCS (correctable):   0
  HCS (uncorrectable): 0
```

```
APS
 not configured

PATH TRACE BUFFER : STABLE

BER thresholds:  SF = 10e-3  SD = 10e-6
TCA thresholds:  B1 = 10e-6  B2 = 10e-6  B3 = 10e-6

  Clock source:  line
```

CHAPTER 7

Configuring the ATM SPAs

This chapter provides information about configuring the ATM SPAs on the Catalyst 6500 Series switch. It includes the following sections:

- Configuration Tasks, page 7-1
- Verifying the Interface Configuration, page 7-61
- Configuration Examples, page 7-63

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Configuration Tasks

This section describes the most common configurations for the ATM SPAs on a Catalyst 6500 Series switch. It contains procedures for the following configurations:

- Required Configuration Tasks, page 7-2
- Specifying the Interface Address on a SPA, page 7-3
- Modifying the Interface MTU Size, page 7-3
- Creating a Permanent Virtual Circuit, page 7-6
- Creating a PVC on a Point-to-Point Subinterface, page 7-8
- Configuring a PVC on a Multipoint Subinterface, page 7-10
- Configuring RFC 1483 Bridging for PVCs, page 7-12
- Configuring RFC 1483 Bridging for PVCs with IEEE 802.1Q Tunneling, page 7-15
- Configuring ATM RFC 1483 Half-Bridging, page 7-17
- Configuring ATM Routed Bridge Encapsulation, page 7-20
- Configuring RFC 1483 Bridging of Routed Encapsulations, page 7-22
- Configuring MPLS over RBE, page 7-25
- Configuring Aggregate WRED for PVCs, page 7-26

Required Configuration Tasks

The ATM SPA interface must be initially configured with an IP address to allow further configuration. Some of the required configuration commands implement default values that might or might not be appropriate for your network. If the default value is correct for your network, then you do not need to configure the command. To perform the basic configuration of each interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/subslot/port`	Enters interface configuration mode for the indicated port on the specified ATM SPA.
Step 2	`Router(config-if)# ip address address mask [secondary]`	(Optional in some configurations) Assigns the specified IP address and subnet mask to the interface. Repeat the command with the optional **secondary** keyword to assign additional, secondary IP addresses to the port.
Step 3	`Router(config-if)# description string`	(Optional) Assigns an arbitrary string, up to 80 characters long, to the interface. This string can identify the purpose or owner of the interface, or any other information that might be useful for monitoring and troubleshooting.
Step 4	`Router(config-if)# no shutdown`	Enables the interface.
	Repeat Step 1 through Step 4 for each port on the ATM SPA to be configured.	
Step 5	`Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Specifying the Interface Address on a SPA

Two ATM SPAs can be installed in a SIP. SPA interface ports begin numbering with "0" from left to right. Single-port SPAs use only the port number 0. To configure or monitor SPA interfaces, you need to specify the physical location of the SIP, SPA, and interface in the CLI. The interface address format is *slot*/*subslot*/*port*, where:

- *slot*—Specifies the chassis slot number in the Catalyst 6500 Series switch where the SIP is installed.
- *subslot*—Specifies the secondary slot of the SIP where the SPA is installed.
- *port*—Specifies the number of the individual interface port on a SPA.

The following example shows how to specify the first interface (0) on a SPA installed in the first subslot of a SIP (0) installed in chassis slot 3:

```
Router(config)# interface serial 3/0/0
```

This command shows a serial SPA as a representative example, however the same *slot*/*subslot*/*port* format is similarly used for other SPAs (such as ATM and POS) and other non-channelized SPAs.

For more information about identifying slots and subslots, see the "Identifying Slots and Subslots for SIPs, SSCs, and SPAs" section on page 4-2.

Modifying the Interface MTU Size

The maximum transmission unit (MTU) values might need to be reconfigured from their defaults on the ATM SPAs to match the values used in your network.

Interface MTU Configuration Guidelines

When configuring the interface MTU size on an ATM SPA, consider the following guidelines.

The Cisco IOS software supports several types of configurable MTU options at different levels of the protocol stack. You should ensure that all MTU values are consistent to avoid unnecessary fragmentation of packets. These MTU values are the following:

- Interface MTU—Configured on a per-interface basis and defines the maximum packet size (in bytes) that is allowed for traffic received on the network. The ATM SPA checks traffic coming in from the network and drops packets that are larger than this maximum value. Because different types of Layer 2 interfaces support different MTU values, choose a value that supports the maximum possible packet size that is possible in your particular network topology.
- IP MTU—Configured on a per-interface or per-subinterface basis and determines the largest maximum IP packet size (in bytes) that is allowed on the IP network without being fragmented. If an IP packet is larger than the IP MTU value, the ATM SPA fragments it into smaller IP packets before forwarding it on to the next hop.
- Multiprotocol Label Switching (MPLS) MTU—Configured on a per-interface or per-subinterface basis and defines the MTU value for packets that are tagged with MPLS labels or tag headers. When an IP packet that contains MPLS labels is larger than the MPLS MTU value, the ATM SPA fragments it into smaller IP packets. When a non-IP packet that contains MPLS labels is larger than the MPLS MTU value, the ATM SPA drops it.

All devices on a particular physical medium must have the same MPLS MTU value to allow proper MPLS operation. Because MPLS labels are added on to the existing packet and increase the packet's size, choose appropriate MTU values so as to avoid unnecessarily fragmenting MPLS-labeled packets.

If the IP MTU or MPLS MTU values are currently the same size as the interface MTU, changing the interface MTU size also automatically sets the IP MTU or MPLS MTU values to the new value. Changing the interface MTU value does not affect the IP MTU or MPLS MTU values if they are not currently set to the same size as the interface MTU.

Different encapsulation methods and the number of MPLS MTU labels add additional overhead to a packet. For example, SNAP encapsulation adds an 8-byte header, IEEE 802.1Q encapsulation adds a 2-byte header, and each MPLS label adds a 4-byte header. Consider the maximum possible encapsulations and labels that are to be used in your network when choosing the MTU values.

Tip The MTU values on the local ATM SPA interfaces must match the values being used in the ATM network and remote ATM interface. Changing the MTU values on an ATM SPA does not reset the local interface, but be aware that other platforms and ATM SPAs do reset the link when the MTU value changes. This could cause a momentary interruption in service, so we recommend changing the MTU value only when the interface is not being used.

Note The interface MTU value on the ATM SPA also determines which packets are recorded as “giants” in the **show interfaces atm** command. The interface considers a packet to be a giant packet when it is more than 24 bytes larger than the interface MTU size. For example, if using an MTU size of 1500 bytes, the interface increments the giants counter when it receives a packet larger than 1524 bytes.

Interface MTU Configuration Task

To change the MTU values on the ATM SPA interfaces, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/subslot/port`	Enters interface configuration mode for the indicated port on the specified ATM SPA.
Step 2	`Router(config-if)# mtu bytes`	(Optional) Configure the maximum transmission unit (MTU) size for the interface. The valid range for *bytes* is from 64 to 9216 bytes, with a default of 4470 bytes. As a general rule, do not change the MTU value unless you have a specific application need to do so. **Note** If the IP MTU or MPLS MTU values are currently the same size as the interface MTU, changing the interface MTU size also automatically sets the IP MTU or MPLS MTU values to the same value.
Step 3	`Router(config-if)# ip mtu bytes`	(Optional) Configures the MTU value, in bytes, for IP packets on this interface. The valid range for an ATM SPA is 64 to 9288, with a default value equal to the MTU value configured in Step 2.

	Command	Purpose
Step 4	`Router(config-if)# mpls mtu bytes`	(Optional) Configures the MTU value, in bytes, for MPLS-labeled packets on this interface. The valid range for an ATM SPA is 64 to 9216 bytes, with a default value equal to the MTU value configured in Step 2.
	Note Repeat Step 1 through Step 4 for each interface port on the ATM SPA to be configured.	
Step 5	`Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the MTU Size

To verify the MTU sizes for an interface, use the **show interface**, **show ip interface**, and **show mpls interface** commands, as in the following example:

```
Router# show interface atm 4/1/0

ATM4/1/0 is up, line protocol is up
  Hardware is SPA-4XOC3-ATM, address is 000d.2959.d5ca (bia 000d.2959.d5ca)
  MTU 4470 bytes, sub MTU 4470, BW 149760 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM, loopback not set
  Encapsulation(s): AAL5
  4095 maximum active VCs, 0 current VCCs
  VC idle disconnect time: 300 seconds
  0 carrier transitions
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out

Router# show ip interface atm 4/1/0

ATM4/1/0 is up, line protocol is up
  Internet address is 200.1.0.2/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 4470 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.9
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP Feature Fast switching turbo vector
  IP Null turbo vector
  VPN Routing/Forwarding "vpn2600-2"
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled

Router# show mpls interface atm 4/1/0 detail

Interface ATM3/0:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        MPLS operational
        MPLS turbo vector
        MTU = 4470
        ATM labels: Label VPI = 1
                Label VCI range = 33 - 65535
                Control VC = 0/32
```

To view the maximum possible size for datagrams passing out the interface using the configured MTU value, use the **show atm interface atm** command:

```
Router# show atm interface atm 4/1/0
   Interface ATM4/1/0:
   AAL enabled: AAL5, Maximum VCs: 4096, Current VCCs: 2
   Maximum Transmit Channels: 0
   Max. Datagram Size: 4528
   PLIM Type: SONET - 155000Kbps, TX clocking: LINE
   Cell-payload scrambling: ON
   sts-stream scrambling: ON
   8359 input, 8495 output, 0 IN fast, 0 OUT fast, 0 out drop
   Avail bw = 155000
   Config. is ACTIVE
```

Creating a Permanent Virtual Circuit

To use a permanent virtual circuit (PVC), configure the PVC in both the switch and the ATM switch. PVCs remain active until the circuit is removed from either configuration. To create a PVC on the ATM interface and enter interface ATM VC configuration mode, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface atm** *slot/subslot/port* or Router(config)# **interface atm** *slot/subslot/port.subinterface*	Enters interface or subinterface configuration mode for the indicated port on the specified ATM SPA.
Step 2	Router(config-if)# **ip address** *address mask*	Assigns the specified IP address and subnet mask to the interface or subinterface.
Step 3	Router(config-if)# **atm tx-latency** *milliseconds*	(Optional) Configures the default transmit latency for VCs on this ATM SPA interface. The valid range for *milliseconds* is from 1 to 200, with a default of 100 milliseconds.
Step 4	Router(config-if)# **pvc** [*name*] *vpi/vci* [**ilmi** | **qsaal**]	Configures a new ATM PVC by assigning its VPI/VCI numbers and enters ATM VC configuration mode. The valid values for *vpi/vci* are: *vpi*—Specifies the VPI ID. The valid range is 0 to 255. *vci*—Specifies the VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except for 5 for the QSAAL PVC and 16 for the ILMI PVC. You can also configure the following options: *name*—(Optional) An arbitrary string that identifies this PVC. **ilmi**—(Optional) Configures the VC to exclusively carry ILMI protocol traffic (default). **qsaal**—(Optional) Configures the VC to exclusively carry qsaal protocol traffic.

Note When using the **pvc** command, remember that the *vpi/vci* combination forms a unique identifier for the interface and all of its subinterfaces. If you specify a *vpi/vci* combination that has been used on another subinterface, the Cisco IOS software assumes that you want to modify that PVC's configuration and automatically switches to its parent subinterface.

	Command	Purpose
Step 5	Router(config-if-atm-vc)# **protocol** *protocol* {*protocol-address* | **inarp**} [[**no**] **broadcast**]	Configures the PVC for a particular protocol and maps it to a specific *protocol-address*. *protocol*—Typically set to either **ip** or **ppp**, but other values are possible. *protocol-address*—Destination address or virtual interface template for this PVC (if appropriate for the *protocol*). **inarp**—Specifies that the PVC uses Inverse ARP to determine its address. [**no**] **broadcast**—(Optional) Specifies that this mapping should (or should not) be used for broadcast packets.
Step 6	Router(config-if-atm-vc)# **inarp** *minutes*	(Optional) If using Inverse ARP, configures how often the PVC transmits Inverse ARP requests to confirm its address mapping. The valid range is 1 to 60 minutes, with a default of 15 minutes.
Step 7	Router(config-if-atm-vc)# **encapsulation aal5snap**	(Optional) Configures the ATM adaptation layer (AAL) and encapsulation type. The default and only supported type is **aal5snap**.

	Command	Purpose
Step 8	`Router(config-if-atm-vc)# tx-limit buffers`	(Optional) Specifies the number of transmit buffers for this VC. The valid range is from 1 to 57343, with a default value that is based on the current VC line rate and on the latency value that is configured with the **atm tx-latency** command.
	Note Repeat Step 4 through Step 8 for each PVC to be configured on this interface.	
Step 9	`Router(config-if-atm-vc)# end`	Exits ATM VC configuration mode and returns to privileged EXEC mode.

Verifying a PVC Configuration

To verify the configuration of a particular PVC, use the **show atm pvc** command:

```
Router# show atm pvc 1/100

ATM3/0/0: VCD: 1, VPI: 1, VCI: 100
UBR, PeakRate: 149760
AAL5-LLC/SNAP, etype:0x0, Flags: 0xC20, VCmode: 0x0
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC status: Not Managed
ILMI VC status: Not Managed
InARP frequency: 15 minutes(s)
Transmit priority 6
InPkts: 94964567, OutPkts: 95069747, InBytes: 833119350, OutBytes: 838799016
InPRoc: 1, OutPRoc: 1, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 94964566, OutAS: 95069746
InPktDrops: 0,  OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP

VC 1/100 doesn't exist on 7 of 8 ATM interface(s)
```

Tip To verify the configuration and current status of all PVCs on a particular interface, you can also use the **show atm vc interface atm** command.

Creating a PVC on a Point-to-Point Subinterface

Use point-to-point subinterfaces to provide each pair of switches with its own subnet. When you create a PVC on a point-to-point subinterface, the switch assumes it is the only point-to-point PVC that is configured on the subinterface, and it forwards all IP packets with a destination IP address in the same subnet to this VC. To configure a point-to-point PVC, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface atm** *slot/subslot/port.subinterface* **point-to-point**	Creates the specified point-to-point subinterface on the given port on the specified ATM SPA, and enters subinterface configuration mode.
Step 2	Router(config-subif)# **ip address** *address mask*	Assigns the specified IP address and subnet mask to this subinterface.
Step 3	Router(config-subif)# **pvc** [*name*] *vpi*/*vci* [**ilmi** | **qsaal**]	Configures a new ATM PVC by assigning its VPI/VCI numbers and enters ATM VC configuration mode. The valid values for *vpi/vci* are: *vpi*—Specifies the VPI ID. The valid range is 0 to 255. *vci*—Specifies the VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except for 5 for the QSAAL PVC and 16 for the ILMI PVC. You can also configure the following options: *name*—(Optional) An arbitrary string that identifies this PVC. **ilmi**—(Optional) Configures the PVC to use ILMI encapsulation (default). **qsaal**—(Optional) Configures the PVC to use QSAAL encapsulation.
	Note When using the **pvc** command, remember that the *vpi/vci* combination forms a unique identifier for the interface and all of its subinterfaces. If you specify a *vpi/vci* combination that has been used on another subinterface, the Cisco IOS software assumes that you want to modify that PVC's configuration and automatically switches to its parent subinterface.	
Step 4	Router(config-if-atm-vc)# **protocol** *protocol protocol-address* [[**no**] **broadcast**]	Configures the PVC for a particular protocol and maps it to a specific *protocol-address*. *protocol*—Typically set to **ppp** for point-to-point subinterfaces, but other values are possible. *protocol-address*—Destination address or virtual template interface for this PVC (as appropriate for the specified *protocol*). [**no**] **broadcast**—(Optional) Specifies that this mapping should (or should not) be used for broadcast packets. The protocol command also has an **inarp** option, but this option is not meaningful on point-to-point PVCs that use a manually configured address.
Step 5	Router(config-if-atm-vc)# **encapsulation aal5snap**	(Optional) Configures the ATM adaptation layer (AAL) and encapsulation type. The default and only supported type is **aal5snap**.
	Repeat Step 1 through Step 5 for each point-to-point subinterface to be configured on this ATM SPA.	
Step 6	Router(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying a Point-to-Point PVC Configuration

To verify the configuration of a particular PVC, use the **show atm pvc** command:

```
Router# show atm pvc 3/12

ATM3/1/0.12: VCD: 3, VPI: 3, VCI: 12
UBR, PeakRate: 149760
AAL5-LLC/SNAP, etype:0x0, Flags: 0xC20, VCmode: 0x0
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC status: Not Managed
ILMI VC status: Not Managed
InARP frequency: 15 minutes(s)
Transmit priority 6
InPkts: 3949645, OutPkts: 3950697, InBytes: 28331193, OutBytes: 28387990
InPRoc: 1, OutPRoc: 1, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 3949645, OutAS: 3950697
InPktDrops: 0,  OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP
```

Tip To verify the configuration and current status of all PVCs on a particular interface, you can also use the **show atm vc interface atm** command.

Configuring a PVC on a Multipoint Subinterface

Creating a multipoint subinterface allows you to create a point-to-multipoint PVC that can be used as a broadcast PVC for all multicast requests. To create a PVC on a multipoint subinterface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm` *slot/subslot/port.subinterface* **multipoint**	Creates the specified point-to-multipoint subinterface on the given port on the specified ATM SPA, and enters subinterface configuration mode.
Step 2	`Router(config-subif)#` **ip address** *address mask*	Assigns the specified IP address and subnet mask to this subinterface.
Step 3	`Router(config-subif)#` **no ip directed-broadcast**	(Optional) Disables the forwarding of IP directed broadcasts, which are sometimes used in denial of service (DOS) attacks.

	Command	Purpose
Step 4	Router(config-subif)# **pvc** [*name*] *vpi*/*vci* [**ilmi** | **qsaal**]	Configures a new ATM PVC by assigning its VPI/VCI numbers and enters ATM VC configuration mode. The valid values for *vpi/vci* are: • *vpi*—Specifies the VPI ID. The valid range is 0 to 255. • *vci*—Specifies the VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except for 5 for the QSAAL PVC and 16 for the ILMI PVC. You can also configure the following options: • *name*—(Optional) An arbitrary string that identifies this PVC. • **ilmi**—(Optional) Configures the PVC to use ILMI encapsulation (default). • **qsaal**—(Optional) Configures the PVC to use QSAAL encapsulation.
	Note When using the **pvc** command, remember that the *vpi/vci* combination forms a unique identifier for the interface and all of its subinterfaces. If you specify a *vpi/vci* combination that has been used on another subinterface, the Cisco IOS software assumes that you want to modify that PVC's configuration and automatically switches to its parent subinterface.	
Step 5	Router(config-if-atm-vc)# **protocol** *protocol* {*protocol-address* | **inarp**} **broadcast**	Configures the PVC for a particular protocol and maps it to a specific *protocol-address*. • *protocol*—Typically set to **ip** for multipoint subinterfaces, but other values are possible. • *protocol-address*—Destination address or virtual template interface for this PVC (if appropriate for the *protocol*). • **inarp**—Specifies that the PVC uses Inverse ARP to determine its address. • **broadcast**— Specifies that this mapping should be used for multicast packets.
Step 6	Router(config-if-atm-vc)# **inarp** *minutes*	(Optional) If using Inverse ARP, configures how often the PVC transmits Inverse ARP requests to confirm its address mapping. The valid range is 1 to 60 minutes, with a default of 15 minutes.
Step 7	Router(config-if-atm-vc)# **encapsulation aal5snap**	(Optional) Configures the ATM adaptation layer (AAL) and encapsulation type. The default and only supported type is **aal5snap**.
	Note Repeat Step 1 through Step 7 for each multipoint subinterface to be configured on this ATM SPA.	
Step 8	Router(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying a Multipoint PVC Configuration

To verify the configuration of a particular PVC, use the **show atm pvc** command:

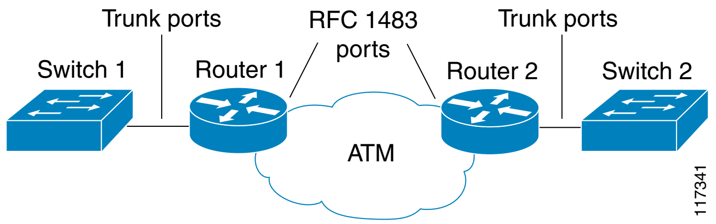
```
Router# show atm pvc 1/120

ATM3/1/0.120: VCD: 1, VPI: 1, VCI: 120
UBR, PeakRate: 149760
AAL5-LLC/SNAP, etype:0x0, Flags: 0xC20, VCmode: 0x0
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC status: Not Managed
ILMI VC status: Not Managed
InARP frequency: 15 minutes(s)
Transmit priority 6
InPkts: 1394964, OutPkts: 1395069, InBytes: 1833119, OutBytes: 1838799
InPRoc: 1, OutPRoc: 1, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 94964, OutAS: 95062
InPktDrops: 0,  OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP
```

Note To verify the configuration and current status of all PVCs on a particular interface, you can also use the **show atm vc interface atm** command.

Configuring RFC 1483 Bridging for PVCs

RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer* 5, specifies the implementation of point-to-point bridging of Layer 2 PDUs from an ATM interface. Figure 7-1 shows an example in which the two routers receive VLANs over their respective trunk links and then forward that traffic out through the ATM interfaces into the ATM cloud. In this example, the device with the ATM SPA is shown as a router, but it can also be a Catalyst 6500 series switch.

Figure 7-1 Example of RFC 1483 Bridging Topology



Tip RFC 1483 has been updated and superseded by RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*.

RFC 1483 Bridging for PVCs Configuration Guidelines

When configuring RFC 1483 bridging for PVCs, consider the following guidelines:

- PVCs must use AAL5 Subnetwork Access Protocol (SNAP) encapsulation.
- To use the Virtual Trunking Protocol (VTP), ensure that each main interface has a subinterface that has been configured for the management VLANs (VLANs 1 and 1002-1005). VTP is not supported on bridged VCs on a Cisco 7600 SIP-200.
- RFC 1483 bridging in a switched virtual circuit (SVC) environment is not supported.
- The 1-Port OC-48c/STM-16 ATM SPA does not support RFC 1483 bridging.

RFC 1483 Bridging for PVCs Configuration Task

To configure RFC 1483 bridging for PVCs, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/subslot/port.subinterface point-to-point`	(Optional) Creates the specified point-to-point subinterface on the given port on the specified ATM SPA card, and enters subinterface configuration mode. **Note** Although it is most common to create the PVCs on subinterfaces, you can also omit this step to create the PVCs for RFC 1483 bridging on the main interface.
Step 2	`Router(config-subif)# pvc [name] vpi/vci [ilmi | qsaal]`	Configures a new ATM PVC by assigning its VPI/VCI numbers and enters ATM VC configuration mode. The valid values for *vpi/vci* are: • *vpi*—Specifies the VPI ID. The valid range is 0 to 255. • *vci*—Specifies the VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except for 5 for the QSAAL PVC and 16 for the ILMI PVC. You can also configure the following options: • *name*—(Optional) An arbitrary string that identifies this PVC. • **ilmi**—(Optional) Configures the PVC to use ILMI encapsulation (default). • **qsaal**—(Optional) Configures the PVC to use QSAAL encapsulation.

	Command	Purpose
Step 3	`Router(config-if-atm-vc)#`**`bridge-domain`** *`vlan-id`* `[`**`access`** `|` **`dot1q`** `|` **`dot1q-tunnel`**`]` `[`**`ignore-bpdu-pid`**`]` `|` `{`**`pvst-tlv`** *`CE-vlan`*`}` `[`**`increment`**`]` `[`**`split-horizon`**`]`	Binds the PVC to the specified *vlan-id*. You can optionally specify the following keywords: • **dot1q**—(Optional) Includes the IEEE 802.1Q tag, which preserves the VLAN ID and class of service (CoS) information across the ATM cloud. • **dot1q-tunnel**—(Optional) Enables tunneling of IEEE 802.1Q VLANs over the same link. See the "Configuring RFC 1483 Bridging for PVCs with IEEE 802.1Q Tunneling" section on page 7-15. • **ignore-bpdu-pid**—(Optional) Ignores bridge protocol data unit (BPDU) packets, to allow interoperation with ATM customer premises equipment (CPE) devices that do not distinguish BPDU packets from data packets. Without this keyword, IEEE BPDUs are sent out using a PID of 0x00-0E, which complies with RFC 1483. With this keyword, IEEE BPDUs are sent out using a PID of 0x00-07, which is normally reserved for RFC 1483 data. • **pvst-tlv—**When transmitting, the **pvst-tlv** keyword translates PVST+ BPDUs into IEEE BPDUs. When receiving, the **pvst-tlv** keyword translates IEEE BPDUs into PVST+ BPDUs. • **split-horizon**—(Optional) Enables RFC 1483 split horizon mode to globally prevent bridging between PVCs in the same VLAN.
Step 4	`Router(config-if-atm-vc)#` **`encapsulation aal5snap`**	(Optional) Configures the ATM adaptation layer (AAL) and encapsulation type. The default and only supported type is **aal5snap**.
	Note Repeat Step 1 through Step 4 for each interface on the ATM SPA to be configured.	
Step 5	`Router(config-if-atm-vc)#` **`end`**	Exits ATM VC configuration mode and returns to privileged EXEC mode.

Verifying the RFC 1483 Bridging Configuration

To verify the RFC 1483 bridging configuration and status, use the **show interface atm** command:

```
Router# show interface atm 6/1/0.3

ATM6/1/0.3 is up, line protocol is up
  Hardware is SPA-4XOC3-ATM
  Internet address is 10.10.10.13/24
  MTU 4470 bytes, BW 149760 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM
  5 packets input, 566 bytes
  5 packets output, 566 bytes
  1445 OAM cells input, 1446 OAM cells output
```

Layer 2 Protocol Tunneling Topology CLI Configuration Task

To enable BPDU translation for the Layer 2 Protocol Tunneling (L2PT) topologies, use the following command:

```
bridge-domain PE-vlan dot1q-tunnel ignore-bpdu-pid pvst-tlv CE-vlan
```

Configuring RFC 1483 Bridging for PVCs with IEEE 802.1Q Tunneling

RFC 1483 bridging (see the "Configuring RFC 1483 Bridging for PVCs" section on page 7-12) can also include IEEE 802.1Q tunneling, which allows service providers to aggregate multiple VLANs over a single VLAN, while still keeping the individual VLANs segregated and preserving the VLAN IDs for each customer. This tunneling simplifies traffic management for the service provider, while securing the customer networks.

Also, the IEEE 802.1Q tunneling is configured only on the service provider switches, so it does not require any additional configuration on the customer-side switches. The customer side is not aware of the configuration.

Note For complete information on IEEE 802.1Q tunneling on the Catalyst 6500 series switch, see the *Catalyst 6500 Series Cisco IOS Software Configuration Guide*, *12.2SX* at the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/book.html

Note RFC 1483 has been updated and superseded by RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*.

RFC 1483 Bridging for PVCs with IEEE 802.1Q Tunneling Configuration Guidelines

When configuring RFC 1483 bridging for PVCs with IEEE 802.1Q tunneling, consider the following guidelines:

- Customer equipment must be configured for RFC 1483 bridging with IEEE 802.1Q tunneling, using the **bridge-domain dot1q** ATM VC configuration command. See the "Configuring RFC 1483 Bridging for PVCs" section on page 7-12 for more information.
- PVCs must use AAL5 encapsulation.
- RFC 1483 bridged PVCs must terminate on the ATM SPA, and the traffic forwarded over this bridged connection to the edge must be forwarded through an Ethernet port.
- To use the Virtual Trunking Protocol (VTP), each main interface should have a subinterface that has been configured for the management VLANs (VLANs 1 and 1002–1005).
- RFC 1483 bridging in a switched virtual circuit (SVC) environment is not supported.

RFC 1483 Bridging for PVCs with IEEE 802.1Q Tunneling Configuration Task

To configure RFC 1483 bridging for PVCs with IEEE 802.1Q tunneling, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface atm** *slot/subslot.port.subinterface* **point-to-point**	(Optional) Creates the specified point-to-point subinterface on the given port on the specified ATM SPA, and enters subinterface configuration mode. **Note** Although it is most common to create the PVCs on subinterfaces, you can also omit this step to create the PVCs for RFC 1483 bridging on the main interface.
Step 2	Router(config-subif)# **pvc** [*name*] *vpi/vci* [**ilmi** | **qsaal**]	Configures a new ATM PVC by assigning its VPI/VCI numbers and enters ATM VC configuration mode. The valid values for *vpi/vci* are: • *vpi*—Specifies the VPI ID. The valid range is 0 to 255. • *vci*—Specifies the VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except for 5 for the QSAAL PVC and 16 for the ILMI PVC. You can also configure the following options: • *name*—(Optional) An arbitrary string that identifies this PVC. • **ilmi**—(Optional) Configures the PVC to use ILMI encapsulation (default). • **qsaal**—(Optional) Configures the PVC to use QSAAL encapsulation.
	Note When using the **pvc** command, remember that the *vpi/vci* combination forms a unique identifier for the interface and all of its subinterfaces. If you specify a *vpi/vci* combination that has been used on another subinterface, the Cisco IOS software assumes that you want to modify that PVC's configuration and automatically switches to its parent subinterface.	
Step 3	Router(config-if-atm-vc)# **bridge-domain** *vlan-id* **dot1q-tunnel**	Binds the PVC to the specified *vlan-id* and enables the use of IEEE 802.1Q tunneling on the PVC. This preserves the VLAN ID information across the ATM cloud.
Step 4	Router(config-if-atm-vc)# **encapsulation aal5snap**	(Optional) Configures the ATM adaptation layer (AAL) and encapsulation type. The default and only supported type is **aal5snap**.
	Note Repeat Step 1 through Step 4 for each interface on the ATM SPA to be configured.	
Step 5	Router(config-if-atm-vc)# **end**	Exits ATM VC configuration mode and returns to privileged EXEC mode.

Verifying the RFC 1483 for PVCs Bridging with IEEE 802.1Q Tunneling Configuration

To verify the IEEE 802.1Q tunneling on an ATM SPA, use the **show 12-protocol-tunnel** command:

```
Router# show l2protocol-tunnel

COS for Encapsulated Packets: 5

Port     Protocol  Shutdown  Drop      Encapsulation Decapsulation Drop
                   Threshold Threshold Counter       Counter       Counter
------- --------  --------- --------- ------------- ------------- -------------
Gi4/2     cdp          ----      ----             0             0             0
          stp          ----      ----             0             0             0
          vtp          ----      ----             0             0             0
ATM6/2/1 cdp          ----      ----           n/a           n/a           n/a
          stp          ----      ----           n/a           n/a           n/a
          vtp          ----      ----           n/a           n/a           n/a
```

Note The counters in the output of the **show l2protocol-tunnel** command are not applicable for ATM interfaces when IEEE 802.1Q tunneling is enabled.

Use the following command to display the interfaces that are configured with an IEEE 802.1Q tunnel:

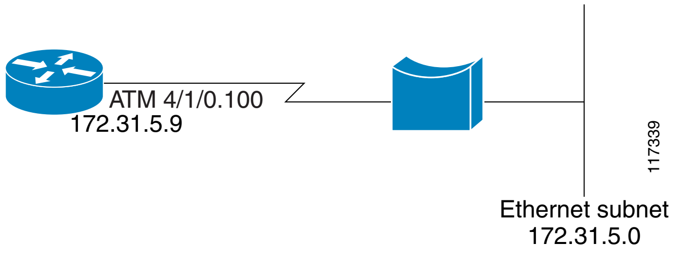
```
Router# show dot1q-tunnel
LAN Port(s)
-----------
Gi4/2

ATM Port(s)
-----------
ATM6/2/1
```

Configuring ATM RFC 1483 Half-Bridging

The ATM SPA supports ATM RFC 1483 half-bridging, which routes IP traffic from a stub-bridged ATM LAN over bridged RFC 1483 Ethernet traffic, without using integrated routing and bridging (IRB). This allows bridged traffic that terminates on an ATM PVC to be routed on the basis of the destination IP address.

For example, Figure 7-2 shows a remote bridged Ethernet network connecting to a routed network over a device that bridges the Ethernet LAN to the ATM interface.

Figure 7-2 ATM RFC 1483 Half-Bridging



When half-bridging is configured, the ATM interface receives the bridged IP packets and routes them according to each packet's IP destination address. Similarly, when packets are routed to this ATM PVC, it then forwards them out as bridged packets on its bridge connection.

This use of a stub network topology offers better performance and flexibility over integrated routing and bridging (IRB). This also helps to avoid a number of issues such as broadcast storms and security risks.

In particular, half-bridging reduces the potential security risks that are associated with normal bridging configurations. Because the ATM interface allocates a single virtual circuit (VC) to a subnet (which could be as small as a single IP address), half-bridging limits the size of the nonsecured network that can be allowed access to the larger routed network. This makes half-bridging configurations ideally suited for customer access points, such digital subscriber lines (DSL).

Note RFC 1483 has been updated and superseded by RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. However, to avoid confusion, this document continues to use the previously-used terminology of "RFC 1483 ATM half-bridging."

To configure a point-to-multipoint ATM PVC for ATM half-bridging, use the configuration task in the following section.

Note Use the following configuration task when you want to configure point-to-multipoint PVCs for half-bridging operation. Use the configuration task in the next section, "Configuring ATM Routed Bridge Encapsulation," to configure a point-to-point PVC for similar functionality.

ATM RFC 1483 Half-Bridging Configuration Guidelines

When configuring ATM RFC 1483 half-bridging, consider the following guidelines:

- Supports only IP traffic and access lists.
- Supports only fast switching and process switching.
- Supports only PVCs that are configured on multipoint subinterfaces. SVCs are not supported for half-bridging.
- A maximum of one PVC can be configured for half-bridging on each subinterface. Other PVCs can be configured on the same subinterface, as long as they are not configured for half-bridging as well.
- The same PVC cannot be configured for both half-bridging and full bridging.

ATM RFC 1483 Half-Bridging Configuration Task

To configure ATM RFC 1483 half-bridging, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface atm`** *`slot/subslot/port.subinterface`* **`multipoint`**	Creates the specified point-to-point subinterface on the given port on the specified ATM SPA, and enters subinterface configuration mode.
Step 2	`Router(config-subif)#` **`ip address`** *`address mask`* `[`**`secondary`**`]`	Assigns the specified IP address and subnet mask to this subinterface. This IP address should be on the same subnet as the remote bridged network (the Ethernet network).

	Command	Purpose
Step 3	`Router(config-subif)# pvc [name] vpi/vci [ilmi | qsaal]`	Configures a new ATM PVC by assigning its VPI/VCI numbers and enters ATM VC configuration mode. The valid values for *vpi/vci* are: • *vpi*—Specifies the VPI ID. The valid range is 0 to 255. • *vci*—Specifies the VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except for 5 for the QSAAL PVC and 16 for the ILMI PVC. You can also configure the following options: • *name*—(Optional) An arbitrary string that identifies this PVC. • **ilmi**—(Optional) Configures the PVC to use ILMI encapsulation (default). • **qsaal**—(Optional) Configures the PVC to use QSAAL encapsulation.
	Note When using the **pvc** command, remember that the *vpi/vci* combination forms a unique identifier for the interface and all of its subinterfaces. If you specify a *vpi/vci* combination that has been used on another subinterface, the Cisco IOS software assumes that you want to modify that PVC's configuration and automatically switches to its parent subinterface.	
Step 4	`Router(config-if-atm-vc)# encapsulation aal5snap bridge`	(Optional) Configures the ATM adaptation layer (AAL) and encapsulation type, and specifies that half-bridging should be used.
Step 5	`Router(config-if-atm-vc)# end`	Exits ATM VC configuration mode and returns to privileged EXEC mode.

Verifying the ATM RFC 1483 Half-Bridging Configuration

To verify the ATM RFC 1483 half-bridging configuration, use the **show atm vc** command:

```
Router# show atm vc 20

ATM4/0/0.20: VCD: 20, VPI: 1, VCI: 20
UBR, PeakRate: 149760
AAL5-LLC/SNAP, etype:0x0, Flags: 0xC20, VCmode: 0x0
OAM frequency: 0 second(s)
InARP frequency: 15 minutes(s), 1483-half-bridged-encap
Transmit priority 6
InPkts: 2411, OutPkts: 2347, InBytes: 2242808, OutBytes: 1215746
InPRoc: 226, OutPRoc: 0
InFast: 0, OutFast: 0, InAS: 2185, OutAS: 2347
InPktDrops: 1,  OutPktDrops: 0
InByteDrops: 0, OutByteDrops: 0
CrcErrors: 139, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 0
OAM cells sent: 0
Status: UP
```

Configuring ATM Routed Bridge Encapsulation

The ATM SPAs support ATM Routed Bridge Encapsulation (RBE), which is similar in functionality to RFC 1483 ATM half-bridging, except that ATM half-bridging is configured on a point-to-multipoint PVC, while RBE is configured on a point-to-point PVC (see the "Configuring ATM RFC 1483 Half-Bridging" section on page 7-17).

Note The 1-Port OC-48c/STM-16 ATM SPA does not support RBE.

Use the following configuration task to configure a point-to-point subinterface and PVC for RBE bridging.

Note RFC 1483 has been updated and superseded by RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*.

ATM Routed Bridge Encapsulation Configuration Guidelines

When configuring ATM RBE, consider the following guidelines:

- Supported only on ATM SPAs in a Cisco 7600 SIP-200. RBE is not supported when using a Cisco 7600 SIP-400.
- Supports only AAL5SNAP encapsulation.
- Supports only IP access lists, not MAC-layer access lists.
- Supports only fast switching and process switching.
- Supports distributed Cisco Express Forwarding (dCEF).
- Supports only PVCs on point-to-point subinterfaces. SVCs are not supported for half-bridging.
- The **bridge-domain** command cannot be used on any PVC that is configured for RBE, because an RBE PVC acts as the termination point for bridged packets.
- The **atm bridge-enable** command, which was used in previous releases on other ATM interfaces, is not supported on ATM SPA interfaces.
- The IS-IS protocol is not supported with point-to-point PVCs that are configured for RBE bridging.

RBE Configuration Limitation Supports Only One Remote MAC Address

On the Catalyst 6500 Series switch with the Supervisor Engine 720 and the following port adapters, an ATM PVC with an RBE configuration can send packets to only a single MAC address:

- ATM SPA on the Cisco 7600 SIP-200 line card

This restriction occurs because the Catalyst 6500 Series switch keeps only one MAC address attached to an RBE PVC. The MAC address-to-PVC mapping is refreshed when a packet is received from the host. If there are multiple hosts connected to the PVC, the mapping will not be stable and traffic forwarding will be affected.

The solution to this problem is as follows:

1. Configure the ATM PVC for RFC 1483 bridging using the **bridge domain** *vlan x* command line interface.
2. Configure an **interface vlan** *vlan x* with the IP address of the RBE subinterface.

ATM Routed Bridge Encapsulation Configuration Task

To configure ATM routed bridge encapsulation, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface atm** *slot/subslot/port.subinterface* **point-to-point**	Creates the specified multipoint subinterface on the given port on the specified ATM SPA, and enters subinterface configuration mode.
Step 2	Router(config-subif)# **atm route-bridge ip**	Enables ATM RFC 1483 half-bridging (RBE bridging). **Note** The **atm route-bridge ip** command can be given either before or after you create the PVC.
Step 3	Router(config-subif)# **ip address** *address mask* [**secondary**]	Assigns the specified IP address and subnet mask to this subinterface. This IP address should be on the same subnet as the remote bridged network (the Ethernet network).
Step 4	Router(config-subif)# **pvc** [*name*] *vpi*/*vci* [**ilmi** | **qsaal**]	Configures a new ATM PVC by assigning its VPI/VCI numbers and enters ATM VC configuration mode. The valid values for *vpi/vci* are: • *vpi*—Specifies the VPI ID. The valid range is 0 to 255. • *vci*—Specifies the VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except for 5 for the QSAAL PVC and 16 for the ILMI PVC. You can also configure the following options: • *name*—(Optional) An arbitrary string that identifies this PVC. • **ilmi**—(Optional) Configures the PVC to use ILMI encapsulation (default). • **qsaal**—(Optional) Configures the PVC to use QSAAL encapsulation.
	Note When using the **pvc** command, remember that the *vpi/vci* combination forms a unique identifier for the interface and all of its subinterfaces. If you specify a *vpi/vci* combination that has been used on another subinterface, the Cisco IOS software assumes that you want to modify that PVC's configuration and automatically switches to its parent subinterface.	
Step 5	Router(config-if-atm-vc)# **encapsulation aal5snap**	(Optional) Configures the ATM adaptation layer (AAL) and encapsulation type. The only supported encapsulation for an RBE PVC is **aal5snap**.
Step 6	Router(config-if-atm-vc)# **end**	Exits ATM VC configuration mode and returns to privileged EXEC mode.

Note The **atm route-bridge ip** command, like other subinterface configuration commands, is not automatically removed when you delete a subinterface. If you want to remove a subinterface and recreate it without the half-bridging, be sure to manually remove the half-bridging configuration, using the **no atm route-bridge ip** command.

Verifying the ATM Routed Bridge Encapsulation Configuration

To verify the RBE bridging configuration, use the **show ip cache verbose** command:

```
Router# show ip cache verbose

IP routing cache 3 entries, 572 bytes
   9 adds, 6 invalidates, 0 refcounts
Minimum invalidation interval 2 seconds, maximum interval 5 seconds,
   quiet interval 3 seconds, threshold 0 requests
Invalidation rate 0 in last second, 0 in last 3 seconds
Last full cache invalidation occurred 00:30:34 ago

Prefix/Length       Age       Interface         Next Hop
10.1.0.51/32-24     00:30:10  Ethernet3/1/0     10.1.0.51    14
0001C9F2A81D00600939BB550800
10.8.100.50/32-24   00:00:04  ATM1/1/0.2        10.8.100.50  28
00010000AA030080C2000700000007144F5D201C0800
10.8.101.35/32-24   00:06:09  ATM1/1/0.4        10.8.101.35  28
00020000AA030080C20007000000E01E8D3F901C0800
```

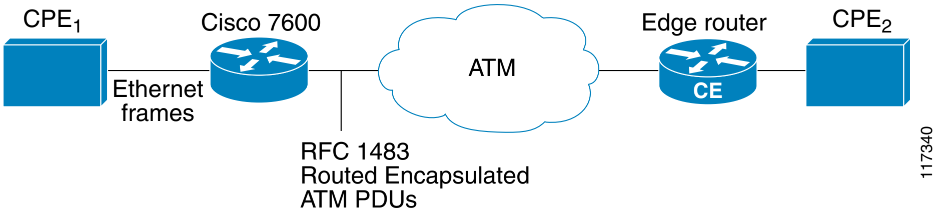
Configuring RFC 1483 Bridging of Routed Encapsulations

Bridging of routed encapsulations (BRE) enables the ATM SPA to receive RFC 1483 routed encapsulated packets and forward them as Layer 2 frames. In a BRE configuration, the PVC receives the routed PDUs, removes the RFC 1483 routed encapsulation header, and adds an Ethernet MAC header to the packet. The Layer 2 encapsulated packet is then switched by the supervisor engine to the Layer 2 interface determined by the VLAN number and destination MAC.

Note The 1-Port OC-48c/STM-16 ATM SPA does not support bridging.

Figure 7-3 shows a topology where an interface on an ATM SPA receives routed PDUs from the ATM cloud and encapsulates them as Layer 2 frames. It then forwards the frames to the Layer 2 customer device. In this example, the device with the ATM SPA is shown as a Cisco 7600 series router, but it can also be a Catalyst 6500 series switch.

Figure 7-3 Example BRE Topology



RFC 1483 Bridging of Routed Encapsulations Configuration Guidelines

When configuring RFC 1483 bridging of routed encapsulations, consider the following guidelines:

- BRE requires that the ATM SPAs are installed in a Cisco 7600 SIP-200.
- PVCs must use AAL5 encapsulation.
- RFC 1483 bridged PVCs must terminate on the ATM SPA, and the traffic forwarded over this bridged connection to the edge must be forwarded through an Ethernet port.
- To use the Virtual Trunking Protocol (VTP), each main interface should have a subinterface that has been configured for the management VLANs (VLANs 1 and 1002–1005).
- BRE is not supported when using a Cisco 7600 SIP-400.
- Concurrent configuration of RFC 1483 bridging and BRE on the same PVC and VLAN is not supported.
- Bridging between RFC 1483 bridged PVCs is not supported.
- RFC 1483 bridging in a switched virtual circuit (SVC) environment is not supported.

RFC 1483 Bridging of Routed Encapsulations Configuration Task

To configure RFC 1483 bridging of routed encapsulations, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface atm`** *`slot/subslot/port`*	Enters interface configuration mode for the indicated port on the specified ATM SPA.
Step 2	`Router(config-if)#` **`no ip address`**	Assigns no IP address to the interface.
Step 3	`Router(config-if)#` **`spanning-tree bpdufilter enable`**	(Optional) Blocks all Spanning Tree BPDUs on the ATM interface. This command should be used if this ATM interface is configured only for BRE VLANs. **Note** If this ATM interface is configured for both BRE and RFC 1483 bridged VLANs, do not enter this command unless you want to explicitly block BPDUs on the interface.
Step 4	`Router(config-if)#` **`no shutdown`**	Enables the interface.
Step 5	`Router(config-if)#` **`interface atm`** *`slot/subslot/port.subinterface`* **`point-to-point`**	Creates the specified point-to-point subinterface on the given port on the specified ATM SPA, and enters subinterface configuration mode.
Step 6	`Router(config-subif)#` **`no ip address`**	Assigns no IP address to the subinterface.

	Command	Purpose
Step 7	`Router(config-subif)#` **`pvc`** `[`*`name`*`]` *`vpi/vci`* `[`**`ilmi`** `|` **`qsaal`**`]`	Configures a new ATM PVC by assigning its VPI/VCI numbers and enters ATM VC configuration mode. The valid values for *vpi/vci* are: • *vpi*—Specifies the VPI ID. The valid range is 0 to 255. • *vci*—Specifies the VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except for 5 for the QSAAL PVC and 16 for the ILMI PVC. You can also configure the following options: • *name*—(Optional) An arbitrary string that identifies this PVC. • **ilmi**—(Optional) Configures the PVC to use ILMI encapsulation (default). • **qsaal**—(Optional) Configures the PVC to use QSAAL encapsulation.
	Note When using the **pvc** command, remember that the *vpi/vci* combination forms a unique identifier for the interface and all of its subinterfaces. If you specify a *vpi/vci* combination that has been used on another subinterface, the Cisco IOS software assumes that you want to modify that PVC's configuration and automatically switches to its parent subinterface.	
Step 8	`Router(config-if-atm-vc)#` **`bre-connect`** *`vlan-id`* `[`**`mac`** *`mac-address`*`]`	Enables BRE bridging on the PVC, where: • **mac** *mac-address*—(Optional) Specifies the hardware (MAC) address of the destination customer premises equipment (CPE) device at the remote end of the VLAN connection.
Step 9	`Router(config-if-atm-vc)#` **`interface gigabitethernet`** *`slot/port`*	Enters interface configuration mode for the specified Gigabit Ethernet interface.
Step 10	`Router(config-if)#` **`switchport`**	Configures the Gigabit Ethernet interface for Layer 2 switching.
Step 11	`Router(config-if)#` **`switchport access vlan`** *`vlan-id`*	(Optional) Specifies the default VLAN for the interface. This should be the same VLAN ID that was specified in the **bre-connect** command in Step 8.
Step 12	`Router(config-if)#` **`switchport mode access`**	Puts the interface into nontrunking mode.
Step 13	`Router(config-if)#` **`end`**	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the RFC 1483 Bridging of Routed Encapsulations Configuration

Use the following commands to verify the RFC 1483 bridging of routed encapsulations configuration:

```
Router# show running-config interface atm 5/0/2.1
!
interface ATM5/0/2.1 point-to-point
 pvc 0/100
  bre-connect 100 ip 10.1.1.2
 !

Router# show running-config interface gigabitethernet 1/2
```

```
interface GigabitEthernet1/2
 no ip address
 switchport
 switchport access vlan 100
 no cdp enable
!

Router# show vlan id 100

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
100  VLAN0100                         active    Gi1/2, AT5/0/2

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
100  enet  100100     1500  -      -      -        -    -        0      0

Router# show atm vlan

Interface       Bridge VCD          Vlan ID
ATM4/5/0/2.1    1                   100
```

Configuring MPLS over RBE

The ATM SPAs support MLPS over RBE on a Cisco 7600 SIP-200. For more information on RBE, see the "Configuring ATM Routed Bridge Encapsulation" section on page 7-20. To configure both RBE and MPLS on the ATM subinterface, perform this task:

	Command	Purpose
Step 1	`Router(config)# `**`interface atm`** *`slot/subslot/port`*	Enters interface configuration mode for the indicated port on the specified ATM SPA.
Step 2	`Router(config-if)# `**`ip address`**	Assigns an IP address to the interface.
Step 3	`Router(config-if)#`**`atm route-bridge ip`**	Configures RBE.
Step 4	`Router(config-if)# `**`mpls ip`**	Configures MPLS.

Verifying MPLS over RBE Configuration

Use the following commands to verify MPLS over RBE configuration:

```
Router#show running interfaces a4/1/0.200
interface ATM4/1/0.200 point-to-point
 ip address 3.0.0.2 255.255.0.0
 atm route-bridged ip
 tag-switching ip
 pvc 10/200
 !
Router#sh mpls interfaces
Interface              IP            Tunnel   Operational
ATM4/1/0.200           Yes (ldp)     No       Yes

Router#show mpls ldp bindings
  tib entry: 5.0.0.0/16, rev 2
        local binding:  tag: imp-null
  tib entry: 6.0.0.0/16, rev 4
```

```
        local binding:  tag: imp-null
        remote binding: tsr: 3.0.0.1:0, tag: imp-null

Router#show mpls ldp neighbor
    Peer LDP Ident: 3.0.0.1:0; Local LDP Ident 3.0.0.2:0
        TCP connection: 3.0.0.1.646 - 3.0.0.2.11001
        State: Oper; Msgs sent/rcvd: 134/131; Downstream
        Up time: 01:51:08
        LDP discovery sources:
          ATM4/1/0.200, Src IP addr: 6.0.0.1
        Addresses bound to peer LDP Ident:
          6.0.0.1

Router#show mpls forwarding
Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Pop tag     3.0.0.0/16        0          AT4/1/0.200 6.0.0.1
17     Pop tag     16.16.16.16/32    0          AT4/1/0.200 6.0.0.1

18     19          13.13.13.13/32    134        AT4/1/0.200 6.0.0.1       <<<<<

19     Pop tag     17.17.17.17/32    0          PO8/0/0.1  point2point
```

Configuring Aggregate WRED for PVCs

Weighted Random Early Detection (WRED) is the Cisco implementation of Random Early Detection (RED) for standard Cisco IOS platforms. RED is a congestion-avoidance technique that takes advantage of the congestion-control mechanism of TCP to anticipate and avoid congestion before it occurs. By dropping packets prior to periods of high congestion, RED tells the packet source (usually TCP) to decrease its transmission rate. When configured, WRED can selectively discard lower priority traffic and provide differentiated performance characteristics for different classes of service.

The Aggregate WRED feature provides a means to overcome limitations of WRED implementations that can only support a limited number of unique subclasses. When an interface enables support for aggregate WRED, subclasses that share the same minimum threshold, maximum threshold and mark probability values can be configured into one aggregate subclass based on their IP precedence value or differentiated services code point (DSCP) value. (The DSCP value is the first six bits of the IP type of service [ToS] byte.) You can also define a default aggregate subclass for all subclasses that have not been explicitly defined.

For more complete information on WRED, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Aggregate WRED Configuration Guidelines

When configuring aggregate WRED on an ATM SPA interface, consider the following guidelines:

- The Aggregate WRED feature requires that the ATM SPAs are installed in a Cisco 7600 SIP-200 or a Cisco 7600 SIP-400.
- With the Aggregate WRED feature, the previous configuration limitation of a maximum of 6 precedence values per class per WRED policy map is no longer in effect.
- When you configure a policy map class for aggregated WRED on an ATM interface, then you cannot also configure the standard *random-detect* commands in interface configuration or policy-map class configuration mode.

- Specifying the **precedence-based** keyword is optional, **precedence-based** is the default form of aggregate WRED.
- The set of subclass values (IP precedence or DSCP) defined on a **random-detect precedence (aggregate)** or **random-detect dscp (aggregate)** CLI will be aggregated into a single hardware WRED resource. The statistics for these subclasses will also be aggregated.
- Defining WRED parameter values for the default aggregate class is optional. If defined, WRED parameters applied to the default aggregate class will be used for all subclasses that have not been explicitly configured. If all possible IP precedence or DSCP values are defined as subclasses, a default specification is unnecessary. If the optional parameters for a default aggregate class are not defined and packets with an unconfigured IP precedence or DSCP value arrive at the interface, these undefined subclass values will be set based on interface (VC) bandwidth.
- After aggregate WRED has been configured in a service policy map, the service policy map must be applied at the ATM VC level (as shown in Step 5 through Step 8 of "Configuring Aggregate WRED Based on IP Precedence").
- The Aggregate WRED feature is not supported in a switched virtual circuit (SVC) environment.

Configuring Aggregate WRED Based on IP Precedence

To configure aggregate WRED to drop packets based on IP precedence values, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# **policy-map** *policy-map-name*`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. • *policy-map-name*—Name of a service policy map to be created. The name can be a maximum of 40 alphanumeric characters.
Step 2	`Router(config-pmap)# **class** {*class-name* | **class-default**}`	Specifies the class policy to be configured. • *class-name*—Name of class you want to configure. Note that WRED can be defined for a user-defined class only if the class has the bandwidth/shape feature enabled. • **class-default**—Default class.

	Command	Purpose
Step 3	`Router(config-pmap-c)#` **`random-detect`** `[`**`precedence-based`**`]` **`aggregate`** `[`**`minimum-thresh`** *`min-thresh`* **`maximum-thresh`** *`max-thresh`* **`mark-probability`** *`mark-prob`*`]`	Enables aggregate WRED based on IP precedence values. If optional parameters for a default aggregate class are not defined, these parameters will be set based on interface (VC) bandwidth. • **precedence-based**—(Optional) Specifies that aggregate WRED is to drop packets based on IP precedence values. This is the default. • *min-thresh*—(Optional) Minimum threshold in number of packets. The value range of this argument is from 1 to 12288. • *max-thresh*—(Optional) Maximum threshold in number of packets. The value range of this argument is from the value of the minimum threshold argument to 12288. • *mark-prob*—(Optional) Denominator for the fraction of packets dropped when the average queue depth is at the maximum threshold. The value range is from 1 to 255.
Step 4	`Router(config-pmap-c)#` **`random-detect precedence values`** *`sub-class-val1`* `[...[`*`sub-class-val8`*`]]` **`minimum-thresh`** *`min-thresh`* **`maximum-thresh`** *`max-thresh`* `[`**`mark-probability`** *`mark-prob`*`]`	Configures the WRED parameters for packets with one or more specific IP precedence values. • *sub-class-val1* [...[*sub-class-val8*]] —One or more specific IP precedence values to which the following WRED profile parameter specifications are to apply. A maximum of 8 subclasses (IP precedence values) can be specified per CLI entry. The IP precedence value can be a number from 0 to 7. • *min-thresh*—Minimum threshold in number of packets. The value range of this argument is from 1 to 12288. • *max-thresh*—Maximum threshold in number of packets. The value range of this argument is from the value of the minimum threshold argument to 12288. • *mark-prob*—Denominator for the fraction of packets dropped when the average queue depth is at the maximum threshold. The value range is from 1 to 255. Repeat this command for each set of IP precedence values that share WRED parameters.

	Command	Purpose
Step 5	`Router(config-pmap-c)# ` **`interface atm`** *`slot/subslot/port.subinterface`* **`point-to-point`**	Creates the specified point-to-point subinterface on the given port on the specified ATM SPA, and enters subinterface configuration mode. • *slot*—Chassis slot number where the SIP is installed. • *subslot*—Secondary slot of the SIP where the SPA is installed. • *port* —Number of the individual interface port on the SPA. • *.subinterface*—Subinterface number. The number that precedes the period must match the number to which this subinterface belongs. The range is 1 to 4,294,967,293.
Step 6	`Router(config-subif)# ` **`ip address`** *`address mask`*	Assigns the specified IP address and subnet mask to the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 7	`Router(config-subif)# ` **`pvc`** `[`*`name`*`]` *`vpi/vci`* `[`**`ilmi`** `|` **`qsaal`**`]`	Configures a new ATM PVC by assigning an optional name and its VPI/VCI numbers. • *name*—(Optional) An arbitrary string that identifies this PVC. • *vpi*—VPI ID. The range is 0 to 255. • *vci*—VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except 5 for the QSAAL PVC and 16 for the ILMI PVC.
Step 8	`Router(config-subif)# ` **`service-policy output`** *`policy-map-name`*	Attaches the specified policy map to the subinterface. • *policy-map-name*—Name of a service policy map to be attached. The name can be a maximum of 40 alphanumeric characters.

Verifying the Precedence-Based Aggregate WRED Configuration

To verify a precedence-based aggregate WRED configuration, use the **show policy-map interface** command. Note that the statistics for IP precedence values 0 through 3 and 4 and 5 have been aggregated into one line each.

```
Router# show policy-map interface a4/1/0.10
 ATM4/1/0.10: VC 10/110 -

 Service-policy output: prec-aggr-wred

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
        Exp-weight-constant: 9 (1/512)
```

```
Mean queue depth: 0
class       Transmitted     Random drop      Tail drop     Minimum   Maximum  Mark
          pkts/bytes       pkts/bytes       pkts/bytes    thresh  thresh  prob

0  1  2  3       0/0             0/0             0/0            10     100  1/10
4  5             0/0             0/0             0/0            40     400  1/10
6                0/0             0/0             0/0            60     600  1/10
7                0/0             0/0             0/0            70     700  1/10
```

Configuring Aggregate WRED Based on DSCP

To configure aggregate WRED to drop packets based on the differentiated services code point (DSCP) value, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`policy-map`** *`policy-map-name`*	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. • *policy-map-name*—Name of a service policy map to be created. The name can be a maximum of 40 alphanumeric characters.
Step 2	`Router(config-pmap)#` **`class`** `{`*`class-name`* `|` **`class-default`**`}`	Specifies the class policy to be configured. • *class-name*—Name of class you want to configure. Note that WRED can be defined for a user-defined class only if the class has the bandwidth/shape feature enabled. • **class-default**—Default class.
Step 3	`Router(config-pmap-c)#` **`random-detect dscp-based aggregate`** `[`**`minimum-thresh`** *`min-thresh`* **`maximum-thresh`** *`max-thresh`* **`mark-probability`** *`mark-prob`*`]`	Enables aggregate WRED based on DSCP values. If optional parameters for a default aggregate class are not defined, these parameters will be set based on interface (VC) bandwidth. • *min-thresh*—(Optional) Minimum threshold in number of packets. The value range of this argument is from 1 to 12288. • *max-thresh*—(Optional) Maximum threshold in number of packets. The value range of this argument is from the value of the minimum threshold argument to 12288. • *mark-prob*—(Optional) Denominator for the fraction of packets dropped when the average queue depth is at the maximum threshold. The value range is from 1 to 255.

	Command	Purpose
Step 4	`Router(config-pmap-c)#` **`random-detect dscp values`** *`sub-class-val1`* `[...[`*`sub-class-val8`*`]]` **`minimum-thresh`** *`min-thresh`* **`maximum-thresh`** *`max-thresh`* `[`**`mark-probability`** *`mark-prob`*`]`	Configures the WRED parameters for packets with one or more specific DSCP values. • *sub-class-val1* [...[*sub-class-val8*]] —One or more DSCP values to which the following WRED parameter specifications are to apply. [A maximum of 8 subclasses (IP precedence values) can be specified per CLI entry.] The DSCP value can be a number from 0 to 63, or it can be one of the following keywords: ef, af11, af12, af13, af21, af22, af23, af31, af32, af33, af41, af42, af43, cs1, cs2, cs3, cs4, cs5, or cs7 • *min-thresh*—Specifies the minimum threshold in number of packets. The value range of this argument is from 1 to 12288. • *max-thresh*—Specifies the maximum threshold in number of packets. The value range of this argument is from the value of the minimum threshold argument to 12288. • *mark-prob*—Specifies the denominator for the fraction of packets dropped when the average queue depth is at the maximum threshold. The value range is from 1 to 255. Repeat this command for each set of DSCP values that share WRED parameters.
Step 5	`Router(config-pmap-c)#` **`interface atm`** *`slot/subslot/port.subinterface`* **`point-to-point`**	Creates the specified point-to-point subinterface on the given port on the specified ATM SPA, and enters subinterface configuration mode. • *slot*—Chassis slot number where the SIP is installed. • *subslot*—Secondary slot of the SIP where the SPA is installed. • *port*—Number of the individual interface port on the SPA. • *.subinterface*—subinterface number. The number that precedes the period must match the number to which this subinterface belongs. The range is 1 to 4,294,967,293.
Step 6	`Router(config-subif)#` **`ip address`** *`address mask`*	Assigns the specified IP address and subnet mask to the interface. • *address*—IP address. • *mask*—Subnet mask.

	Command	Purpose
Step 7	`Router(config-subif)#` **`pvc`** `[`*`name`*`]` *`vpi/vci`* `[`**`ilmi`** `|` **`qsaal`**`]`	Configures a new ATM PVC by assigning an optional name and its VPI/VCI numbers. • *name*—(Optional) An arbitrary string that identifies this PVC. • *vpi*—VPI ID. The range is 0 to 255. • *vci*—VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except 5 for the QSAAL PVC and 16 for the ILMI PVC.
Step 8	`Router(config-subif)#` **`service-policy output`** *`policy-map-name`*	Attaches the specified policy map to the subinterface. • *policy-map-name*—Name of a service policy map to be attached. The name can be a maximum of 40 alphanumeric characters

Verifying the DSCP-Based Aggregate WRED Configuration

To verify a DSCP-based aggregate WRED configuration, use the **show policy-map interface** command. Note that the statistics for DSCP values 0 through 3, 4 through 7, and 8 through 11 have been aggregated into one line each.

```
Router# show policy-map interface a4/1/0.11
 ATM4/1/0.11: VC 11/101 -

  Service-policy output: dscp-aggr-wred

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
        Exp-weight-constant: 0 (1/1)
        Mean queue depth: 0
        class       Transmitted     Random drop      Tail drop     Minimum   Maximum  Mark
                  pkts/bytes       pkts/bytes       pkts/bytes    thresh  thresh  prob
        default          0/0               0/0              0/0            1       10  1/10
        0  1  2  3
        4  5  6  7       0/0               0/0              0/0           10       20  1/10
        8  9  10 11      0/0               0/0              0/0           10       40  1/10
```

Creating and Configuring Switched Virtual Circuits

A switched virtual circuit (SVC) is created and released dynamically, providing user bandwidth on demand. To enable the use of SVCs, you must configure a signaling protocol to be used between the Catalyst 6500 Series switch and the ATM switch. The ATM SPA supports versions 3.0, 3.1, and 4.0 of the User-Network Interface (UNI) signaling protocol, which uses the Integrated Local Management Interface (ILMI) to establish, maintain, and clear the ATM connections at the UNI.

The Catalyst 6500 Series switch does not perform ATM-level call routing when configured for UNI/ILMI operation. Instead, the ATM switch acts as the network and performs the call routing, while the Catalyst 6500 Series switch acts only as the user end-point of the call circuit and only routes packets through the resulting circuit.

Note The 1-Port OC-48c/STM-16 ATM SPA does not support SVCs,

To use UNI/ILMI signaling, you must create an ILMI PVC and a signaling PVC to be used for the SVC call-establishment and call-termination messages between the ATM switch and Catalyst 6500 Series switch. This also requires configuring the ATM interface with a network service access point (NSAP) address that uniquely identifies itself across the network.

The NSAP address consists of a network prefix (13 hexadecimal digits), a unique end station identifier (ESI) of 6 hexadecimal bytes, and a selector byte. If an ILMI PVC exists, the Catalyst 6500 Series switch can obtain the NSAP prefix from the ATM switch, and you must manually configure only the ESI and selector byte. If an ILMI PVC does not exist, or if the ATM switch does not support this feature, you must configure the entire address manually.

To create and configure an SVC, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`interface atm`** *`slot/subslot/port`*	Enters interface configuration mode for the indicated port on the specified ATM SPA.
Step 2	`Router(config-subif)# `**`pvc`**` [`*`name`*`] `**`0/5 qsaal`**	Configures a new ATM PVC to be used for SVC signaling: • *name*—(Optional) An arbitrary string that identifies this PVC. • *vpi*—Specifies the VPI ID. The valid range is 0 to 255, but the recommended value for *vpi* for the signaling PVC is 0. • *vci*—Specifies the VCI ID. The valid range is 1 to 65535, but the recommended value for *vci* for the QSAAL signaling PVC is 5. **Note** The ATM switch must be configured with the same VPI and VCI values for this PVC. • **qsaal**—Configures the signaling PVC to use QSAAL encapsulation.

	Command	Purpose
Step 3	`Router(config-subif)# pvc [name] 0/16 ilmi`	Creates a new ATM PVC to be used for ILMI signaling: • *name*—(Optional) An arbitrary string that identifies this PVC. • *vpi*—Specifies the VPI ID. The valid range is 0 to 255, but the recommended value for *vpi* for the ILMI PVC is 0. • *vci*—Specifies the VCI ID. The valid range is 1 to 65535, but the recommended value for *vci* for the ILMI PVC is 16. • **ilmi**—Configures the PVC to use ILMI encapsulation.
	Note The signaling and ILMI PVCs must be set up on the main ATM interface, not on a subinterface.	
Step 4	`Router(config-if-atm-vc)# exit`	Exits ATM PVC configuration mode and returns to interface configuration mode.
Step 5	`Router(config-if)# atm ilmi-keepalive [seconds] [retry counts]`	(Optional) Enables ILMI keepalive messages and sets the interval between them. ILMI keepalive messages are disabled by default. • *seconds*—(Optional) The amount of time, in seconds, between keepalive messages between the Catalyst 6500 Series switch and the ATM switch. The valid range is 1 to 65535, with a default of 3 seconds. • **retry** *counts*—(Optional) Specifies the number of times the Catalyst 6500 Series switch should resend a keepalive message if the first message is unacknowledged. The valid range is 2 to 5, with a default of 4.
Step 6	`Router(config-if)# atm esi-address esi.selector` or	Specifies the end station ID (ESI) and selector fields for the local portion of the ATM interface's NSAP address, and configures the interface to get the NSAP prefix from the ATM switch. • *esi*—Specifies a string of 12 hexadecimal digits, in dotted notation, for the ATM interface's ESI value. This value must be unique across the network. • *selector*—Specifies a string of 2 hexadecimal digits for the selector byte for this ATM interface. To configure the ATM address, you need to enter only the ESI (12 hexadecimal digits) and the selector byte (2 hexadecimal digits). The NSAP prefix (26 hexadecimal digits) is provided by the ATM switch. or

	Command	Purpose
	Router(config-if)# **atm nsap-address** *nsap-address*	Assigns a complete NSAP address (40 hexadecimal digits) to the ATM interface. The address consists of a network prefix, ESI, and selector byte, and must be in the following format: XX.XXXX.XX.XXXXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XX **Note** The above dotted hexadecimal format provides some validation that the address is a legal value. If you know that the NSAP address is correct, you may omit the dots.

Note The **atm esi-address** and **atm nsap-address** commands are mutually exclusive. Configuring the Catalyst 6500 Series switch with one of these commands automatically negates the other. Use the **show interface atm** command to display the NSAP address that is assigned to the interface.

Step 7	Router(config-if)# **interface atm** *slot/subslot/port.subinterface* [**multipoint** | **point-to-point**]	(Optional) Creates the specified subinterface on the specified ATM interface, and enters subinterface configuration mode. **Note** You can create SVCs on either the main ATM interface or on a multipoint subinterface.
Step 8	Router(config-subif)# **svc** [*name*] **nsap** *address*	Creates an SVC and specifies the destination NSAP address (40 hexadecimal digits in dotted notation). You can also configure the following option: • *name*—(Optional) An arbitrary string that identifies this SVC.
Step 9	Router(config-if-atm-vc)# **oam-svc** [**manage**] [*frequency*]	Enables end-to-end Operation, Administration, and Maintenance (OAM) loopback cell generation and management of the SVC. • **manage**—(Optional) Enables OAM management of the SVC. • *frequency*—(Optional) Specifies the delay between transmitting OAM loopback cells. The valid range is 0 to 600 seconds, with a default of 10 seconds.
Step 10	Router(config-if-atm-vc)# **protocol** *protocol* {*protocol-address* | **inarp**} [[**no**] **broadcast**]	Configures the SVC for a particular protocol and maps it to a specific *protocol-address*. • *protocol*—Typically set to either **ip** or **ppp**, but other values are possible. • *protocol-address*—Destination address or virtual interface template for this SVC (if appropriate for the *protocol*). • **inarp**—Specifies that the SVC uses Inverse ARP to determine its address. • [**no**] **broadcast**—(Optional) Specifies that this mapping should (or should not) be used for broadcast packets.

	Command	Purpose
Step 11	`Router(config-if-atm-vc)# **encapsulation aal5snap**`	(Optional) Configures the ATM adaptation layer (AAL) and encapsulation type. The default and only supported type is **aal5snap**.
	Note Repeat Step 7 through Step 11 for each SVC to be created.	
Step 12	`Router(config-if-atm-vc)# **end**`	Exits SVC configuration mode and returns to privileged EXEC mode.

Verifying the SVC Configuration

Use the **show atm svc** and **show atm ilmi-status** commands to verify the configuration of the SVCs that are currently configured on the Catalyst 6500 Series switch.

```
Router# show atm svc

           VCD /                                        Peak  Avg/Min Burst
Interface  Name         VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells  Sts
4/0/0      1              0     5  SVC    SAAL     UBR  155000                UP
4/0/2      4              0    35  SVC    SNAP     UBR  155000                UP
4/1/0      16             0    47  SVC    SNAP     UBR  155000                UP
4/1/0.1    593            0    80  SVC    SNAP      UBR  155000                 UP
```

Tip To display all SVCs on a particular ATM interface or subinterface, use the **show atm svc interface atm** command.

To display detailed information about a particular SVC, specify its VPI and VCI values:

```
Router# show atm svc 0/35

ATM5/1/0.200: VCD: 3384, VPI: 0, VCI: 35, Connection Name: SVC00
UBR, PeakRate: 155000
AAL5-MUX, etype:0x800, Flags: 0x44, VCmode: 0x0
OAM frequency: 10 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Received
OAM VC status: Verified
ILMI VC status: Not Managed
VC is managed by OAM.
InARP DISABLED
Transmit priority 6
InPkts: 0, OutPkts: 4, InBytes: 0, OutBytes: 400
InPRoc: 0, OutPRoc: 4, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
InPktDrops: 0,  OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 10
F5 InEndloop: 10, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 10
F5 OutEndloop: 10, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP
TTL: 4
interface =  ATM5/1/0.200, call locally initiated, call reference = 8094273
vcnum = 3384, vpi = 0, vci = 35, state = Active(U10)
, point-to-point call
```

```
Retry count: Current = 0
timer currently inactive, timer value = 00:00:00
Remote Atm Nsap address: 47.00918100000000107B2B4B01.111155550001.00
, VC owner: ATM_OWNER_SMAP
```

To display information about the ILMI status and NSAP addresses being used for the SVCs on an ATM interface, use the **show atm ilmi-status** command:

```
Router# show atm ilmi-status atm 4/1/0

Interface : ATM4/1/0 Interface Type : Private UNI (User-side)
ILMI VCC : (0, 16) ILMI Keepalive : Enabled/Up (5 Sec 4 Retries)
ILMI State:       UpAndNormal
Peer IP Addr:     10.10.13.1       Peer IF Name:     ATM 3/0/3
Peer MaxVPIbits:  8               Peer MaxVCIbits:  14
Active Prefix(s) :
47.0091.8100.0000.0010.11b8.c601
End-System Registered Address(s) :
47.0091.8100.0000.0010.11b8.c601.2222.2222.2222.22(Confirmed)
47.0091.8100.0000.0010.11b8.c601.aaaa.aaaa.aaaa.aa(Confirmed)
```

Tip To display information about the SVC signaling PVC and ILMI PVC, use the **show atm pvc 0/5** and **show atm pvc 0/16** commands.

Configuring Traffic Parameters for PVCs or SVCs

After creating a PVC or SVC, you can also configure it for the type of traffic quality of service (QoS) class to be used over the circuit:

- Constant Bit Rate (CBR)—Configures the CBR service class and specifies the average cell rate for the PVC or SVC.
- Unspecified Bit Rate (UBR)—Configures the UBR service class and specifies the output peak rate (PCR) for the PVC or SVC. This is the default configuration. SVCs can also be configured with similar input parameters.
- Unspecified Bit Rate Plus (UBR+)—Configures the UBR+ service class and specifies the output peak cell rate (PCR) and minimum cell rate (MCR) for the SVC. SVCs can also be configured with similar input parameters.

Note The 1-Port OC-48c/STM-16 ATM SPA does not support UBR+.

- Variable Bit Rate–Nonreal Time (VBR-nrt)—Configures the VBR-nrt service class and specifies the output PCR, output sustainable cell rate (SCR), and output maximum burst size (MBS) for the PVC or SVC. SVCs can also be configured with similar input parameters.
- Variable Bit Rate–Real Time (VBR-rt)—Configures the VBR-rt service class and the peak rate and average rate burst for the PVC or SVC.

Each service class is assigned a different transmit priority, which the Catalyst 6500 Series switch uses to determine which queued cell is chosen to be transmitted out of an interface during any particular cell time slot. This ensures that real-time QoS classes have a higher likelihood of being transmitted during periods of congestion. Table 7-1 lists the ATM QoS classes and their default transmit priorities.

Table 7-1 ATM Classes of Service and Default Transmit Priorities

Service Category	Transmit Priority[1]
Signaling, Operation, Administration, and Maintenance (OAM) cells, and other control cells	0 (highest)
CBR when greater than 5 percent of the line rate	1
CBR when less than 5 percent of the line rate	2
Voice traffic	3
VBR-rt	4
VBR-nrt	5
UBR	6
Unused and not available or configurable	7 (lowest)

[1] The default priorities can be changed for individual VCs using the **transmit-priority** VC configuration command.

Note When using a CBR VC that exceeds half of the interface line rate, it is possible in some cases that the shaping accuracy for the CBR traffic can drop from 99 percent to 98 percent when the interface is also configured for UBR VCs that are oversubscribed (that is, the UBR VCs are configured for a total line rate that exceeds the interface line rate). If this small drop in accuracy is not acceptable, then we recommend using VBR-rt or VBR-nrt instead of CBR when oversubscribing UBR traffic.

You can configure a PVC or SVC for only one QoS service class. If you enter more than one type, only the most recently configured QoS class takes effect on the circuit.

To configure the traffic parameters for a PVC or SVC, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface atm** *slot/subslot* or Router(config)# **interface atm** *slot/subslot/port.subinterface* [**multipoint** | **point-to-point**]	Enters interface or subinterface configuration mode for the indicated port on the specified ATM SPA.
Step 2	Router(config-if)# **pvc** [*name*] *vpi/vci* or Router(config-if)# **svc** [*name*] *nsap-address*	Specifies the PVC or SVC to be configured, and enters PVC/SVC configuration mode.
	Note When using the **pvc** command, remember that the *vpi/vci* combination forms a unique identifier for the interface and all of its subinterfaces. If you specify a *vpi/vci* combination that has been used on another subinterface, the Cisco IOS software assumes that you want to modify that PVC's configuration and automatically switches to its parent subinterface.	
Step 3	Router(config-if-atm-vc)# **cbr** *rate*	Configures constant bit rate (CBR) quality of service (QoS) and average cell rate for the PVC or SVC: • *rate*—Average cell rate in kbps. The valid range is 48 to 149760 (OC-3) or 599040 (OC-12). or

	Command	Purpose
	Router(config-if-atm-vc)# **ubr** *output-pcr* [*input-pcr*]	Configures unspecified bit rate (UBR) quality of service (QoS) and peak cell rate (PCR) for the PVC or SVC: • *output-pcr*—Output PCR in kbps. The valid range is 48 to 149760 (OC-3), 599040 (OC-12), or 2396160 (1-Port OC-48c/STM-16 ATM SPA). • *input-pcr*—(Optional for SVCs only) Input PCR in kbps. If omitted, *input-pcr* equals *output-pcr.*
	or	or
	Router(config-if-atm-vc)# **vbr-nrt** *output-pcr output-scr output-mbs* [*input-pcr*] [*input-scr*] [*input-mbs*]	Configures the variable bit rate–nonreal time (VBR-nrt) QoS, the peak cell rate (PCR), sustainable cell rate (SCR), and maximum burst cell size (MBS) for the PVC or SVC: • *output-pcr*—Output PCR in kbps. The valid range is 48 to 149760 (OC-3), 599040 (OC-12), or 2396160 (1-Port OC-48c/STM-16 ATM SPA). • *output-scr*—Output SCR in kbps. The valid range is 48 to PCR, and typically is less than the PCR value. • *output-mbs*—Output MBS in number of cells. The valid range is 1 to 65535, depending on the PCR and SCR values. If the PCR and SCR are configured to the same value, the only valid value for MBS is 1. • *input-pcr*—(Optional for SVCs only) Input PCR in kbps. • *input-scr*—(Optional for SVCs only) Input SCR in kbps. • *input-mbs*—(Optional for SVCs only) Input MBS in number of cells.
	or	or
	Router(config-if-atm-vc)# **vbr-rt** *pcr scr burst*	Configures the variable bit rate–real time (VBR-rt) QoS, and the PCR, average cell rate (ACR), and burst cell size (BCS) for the PVC or SVC: • *pcr*—PCR in kbps. The valid range is 48 to 149760 (OC-3), 599040 (OC-12), or 2396160 (1-Port OC-48c/STM-16 ATM SPA). • *scr*—SCR in kbps. The valid range is 48 to PCR, and typically is less than the PCR value. • *burst*—Burst size in number of cells. The valid range is 1 to 65535, depending on the PCR and SCR values. If the PCR and SCR are configured to the same value, the only valid value for *burst* is 1.
Step 4	Router(config-if-atm-vc)# **transmit-priority** *level*	(Optional) Configures the PVC for a new transmit priority level. • *level*—Priority level from 1 to 6. The default value is determined by the PVC’s configured service class (see Table 7-1 on page 7-38 for the default levels).

	Command	Purpose
	Note Repeat Step 2 through Step 4 for each PVC or SVC to be configured.	
Step 5	`Router(config-if-atm-vc)# end`	Exits PVC/SVC configuration mode and returns to privileged EXEC mode.

Verifying the Traffic Parameter Configuration

To verify the configuration of the traffic parameters for a PVC or SVC, use the **show atm vc** command:

```
Router# show atm vc 20

   ATM1/1/0.200: VCD: 20, VPI: 2, VCI: 200
   UBR, PeakRate: 44209
   AAL5-LLC/SNAP, etype:0x0, Flags: 0xC20, VCmode: 0x0
   OAM frequency: 0 second(s)
   InARP frequency: 5 minutes(s)
   Transmit priority 4
   InPkts: 10, OutPkts: 11, InBytes: 680, OutBytes: 708
   InPRoc: 10, OutPRoc: 5, Broadcasts: 0
   InFast: 0, OutFast: 0, InAS: 0, OutAS: 6
   InPktDrops: 0, OutPktDrops: 0
   CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0
   OAM cells received: 0
   OAM cells sent: 0
   Status: UP
```

To verify the configuration of all PVCs or SVCs on an interface, use the **show atm vc interface atm** command:

```
Router# show atm vc interface atm 2/1/0

ATM2/1/0.101: VCD: 201, VPI: 20, VCI: 101
UBR, PeakRate: 149760
AAL5-LLC/SNAP, etype:0x0, Flags: 0xC20, VCmode: 0x0
OAM frequency: 0 second(s)
InARP frequency: 15 minutes(s)
Transmit priority 4
InPkts: 3153520, OutPkts: 277787, InBytes: 402748610, OutBytes: 191349235
InPRoc: 0, OutPRoc: 0, Broadcasts: 0
InFast: 211151, OutFast: 0, InAS: 0, OutAS: 0
InPktDrops: 0, OutPktDrops: 17
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0
OAM cells received: 0
OAM cells sent: 0
Status: UP
```

Configuring Virtual Circuit Classes

When multiple PVCs or SVCs use the same or similar configurations, you can simplify the Catalyst 6500 Series switch's configuration file by creating virtual circuit (VC) classes. Each VC class acts as a template, which you can apply to an ATM interface or subinterface, or to individual PVCs or SVCs.

When you apply a VC class to an ATM interface or subinterface, all PVCs and SVCs created on that interface or subinterface inherit the VC class configuration. When you apply a VC class to an individual PVC or SVC, that particular PVC or SVC inherits the class configuration.

You can then customize individual PVCs and SVCs with further configuration commands. Any commands that you apply to individual PVCs and SVCs take precedence over those of the VC class that were applied to the interface or to the PVC/SVC.

To create and configure a VC class, and then apply it to an interface, subinterface, or individual PVC or SVC, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **vc-class atm** *vc-class-name*	Creates an ATM virtual circuit (VC) class and enters VC-class configuration mode. • *vc-class-name*—Arbitrary name to identify this particular VC class.
Step 2	Router(config-vc-class)# *configuration-commands*	Enter any PVC or SVC configuration commands for this VC class. See the "Creating a Permanent Virtual Circuit" section on page 7-6 and "Creating and Configuring Switched Virtual Circuits" section on page 7-33 for additional information. **Note** You can specify both PVC and SVC configuration commands in the same VC class. If a command is not appropriate for a PVC or SVC, it is ignored when the VC class is assigned to the PVC or SVC.
Step 3	Router(config-vc-class)# **interface atm** *slot/subslot/port* or Router(config-vc-class)# **interface atm** *slot/subslot/port.subinterface* [**multipoint** | **point-to-point**]	Enters subinterface configuration mode for the specified ATM interface or subinterface.
Step 4	Router(config-if)# **class-int** *vc-class-name*	(Optional) Applies a VC class on the ATM main interface or subinterface. This class then applies to all PVCs or SVCs that are created on that interface. • *vc-class-name*—Name of the VC class that was created in Step 1.
Step 5	Router(config-if)# **pvc** [*name*] *vpi/vci* or Router(config-if)# **svc** [*name*] *nsap-address*	Specifies the PVC or SVC to be configured, and enters ATM VC configuration mode.
	Note When using the **pvc** command, remember that the *vpi/vci* combination forms a unique identifier for the interface and all of its subinterfaces. If you specify a *vpi/vci* combination that has been used on another subinterface, the Cisco IOS software assumes that you want to modify that PVC's configuration and automatically switches to its parent subinterface.	
Step 6	Router(config-if-atm-vc)# **class-vc** *vc-class-name*	Assigns the specified VC class to this PVC or SVC. • *vc-class-name*—Name of the VC class that was created in Step 1.

	Command	Purpose

Step 7	`Router(config-if-atm-vc)#`
	`configuration-commands`
		Any other VC configuration commands to be applied to this particular PVC or SVC. Commands that are applied to the individual PVC or SVC supersede any conflicting commands that were specified in the VC class.

Step 8	`Router(config-if)#` **`end`**	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Virtual Circuit Class Configuration

To verify the virtual circuit class configuration, use the **show atm vc** command:

```
Router# show atm vc

             VCD /                                         Peak  Avg/Min Burst
Interface    Name        VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells  Sts
6/1/0        1             0     5  PVC    SAAL     UBR  155000               UP
6/1/0        2             0    16  PVC    ILMI     UBR  155000               UP
6/1/0.1      3             1    32  PVC-D  SNAP     UBR  155000               UP
6/1/0.2      4             2    32  PVC-D  SNAP     UBR  155000               UP
```

Configuring Virtual Circuit Bundles

Virtual circuit bundles are similar to VC classes because they allow you to configure a large group of PVCs by configuring a template (the VC bundle). The main difference between a VC bundle and a VC class is that the VC bundle management allows you to configure multiple VCs that have different QoS characteristics between any pair of ATM-connected switches.

Using VC bundles, you first create an ATM VC bundle and then add VCs to it, and each VC in the bundle can have its own ATM traffic class and ATM traffic parameters. You can configure the VCs collectively at the bundle level, or you can configure the individual VC bundle members. You can also apply a VC class to a bundle to apply the VC class configuration to all of the VCs in the bundle.

You can create differentiated service by mapping one or more MPLS EXP levels to each VC in the bundle, which enables individual VCs in the bundle to carry packets marked with different MPLS EXP levels. The ATM VC bundle manager determines which VC to use for a particular packet by matching the MPLS EXP level of the packet to the MPLS EXP levels assigned to the VCs in the bundle. The bundle manager can also use Weighted Random Early Detection (WRED) or distributed WRED (dWRED) to further differentiate service across traffic that has different MPLS EXP levels.

Virtual Circuit Bundles Configuration Guidelines

- VC bundles are supported only on ATM SPAs in a Cisco 7600 SIP-200. Bundles are not supported for ATM SPAs in a Cisco 7600 SIP-400.
- VC bundles can be used only for PVCs, not SVCs.
- VC bundles require ATM PVC management, as well as Forwarding Information Base (FIB) and Tag Forwarding Information Base (TFIB) switching functionality.
- The Catalyst 6500 Series switch at the remote end of the network must be using a version of Cisco IOS that supports MPLS and ATM PVC management.

Virtual Circuit Bundles Configuration Task

To create and configure a VC bundle and then apply it to an ATM interface or subinterface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **ip cef** [**distributed**]	Enables Cisco Express Forwarding (CEF) Layer 3 switching on the Catalyst 6500 Series switch. The Catalyst 6500 Series switch enables CEF by default. • **distributed**—(Optional) Enables distributed CEF (dCEF).
Step 2	Router(config)# **mpls label protocol ldp**	Specifies the default label distribution protocol for a platform.
Step 3	Router(config)# **interface atm** *slot/subslot/port* or Router(config)# **interface atm** *slot/subslot/port.subinterface* [**multipoint** | **point-to-point**]	Enters interface configuration mode for the specified ATM interface or subinterface.
Step 4	Router(config-if)# **mpls ip**	Enables MPLS forwarding of IPv4 packets along normally routed paths for the interface.
Step 5	Router(config-if)# **bundle** *bundle-name*	Creates an ATM virtual circuit (VC) bundle and enters bundle configuration mode. • *bundle-name*—Arbitrary name to identify this particular VC bundle.
Step 6	Router(config-if-atm-bundle)# **class-bundle** *vc-class-name*	(Optional) Applies a VC class to this bundle. The class configuration is then applied to all VCs in the bundle. • *vc-class-name*—Name of the VC class to be applied to this bundle and its PVCs or SVCs. See the "Configuring Virtual Circuit Classes" section on page 7-40 for information on creating VC classes.
Step 7	Router(config-if-atm-bundle)# *configuration-commands*	Enter any other PVC or SVC configuration commands for this VC bundle. See "Creating a Permanent Virtual Circuit" section on page 7-6 and "Creating and Configuring Switched Virtual Circuits" section on page 7-33 for additional information.
	Note Configuration commands applied directly to the VC bundle supersede a configuration that is applied through a VC class.	
Step 8	Router(config-if-atm-bundle)# **pvc-bundle** [*name*] *vpi/vci*	Creates a member PVC of the bundle and enters PVC bundle configuration mode.

	Command	Purpose
Step 9	`Router(config-if-atm-member)# mpls experimental [level | other | range]`	(Optional) Configures the MPLS EXP levels for the PVC bundle member. • *level*—MPLS EXP level for the PVC bundle member. The valid range is 0 to 7. • **other**—Any MPLS EXP levels in the range from 0 to 7 that are not explicitly configured (default). • *range*—A range of MPLS EXP levels between 0 and 7, separated by a hyphen.
Step 10	`Router(config-if-atm-member)# bump {implicit | explicit precedence-level | traffic}`	(Optional) Configures the bumping rules for the PVC bundle member. • **implicit**—Bumped traffic is carried by a VC with a lower precedence (default). • **explicit** *precedence-level*—Specifies the precedence level of the traffic that should be bumped when the PVC member goes down. The *precedence-level* can range from 0 to 9. • **traffic**—The PVC member accepts bumped traffic (default). Use **no bump traffic** to specify that the PVC member does not accept bumped traffic.
Step 11	`Router(config-if-atm-member)# protect {group | vc}`	(Optional) Specifies that the PVC bundle member is protected. • **group**—Specifies that the PVC bundle member is part of a protected group. When all members of a protected group go down, the bundle goes down. • **vc**—Specifies that the PVC bundle member is individually protected. When a protected VC goes down, it also takes the bundle down. By default, PVC bundle members are not protected.
Step 12	`Router(config-if-atm-member)# configuration-commands`	Any other VC configuration commands to be applied to this particular VC bundle member. Commands that are applied to a bundle member supersede any conflicting commands that were specified in the VC class or VC bundle.
	Note Repeat Step 8 through Step 12 for each PVC member of the bundle to be created.	
Step 13	`Router(config-if-atm-member)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Virtual Circuit Bundles Configuration

To verify the configuration of the virtual circuit bundles, display the configuration for its interface or subinterface, use the **show running-config interface atm** command, as in the following example:

```
Router# show running-config interface atm 4/1/0.2

interface ATM4/1/0.2 point-to-point
 ip address 10.10.10.1 255.255.255.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 bundle ABC
  class-bundle bundle-class
  pvc-bundle ABC-high 1/107
   class-vc high
  pvc-bundle ABC-med 1/105
   class-vc med
  pvc-bundle ABC-low 1/102
   class-vc low
 !
!
```

To verify the operation and current status of a virtual circuit bundle, specify the bundle name with the **show atm bundle** command:

```
Router# show atm bundle ABC

ABC on ATM4/1/0.2: UP

                       Config    Current   Bumping  PG/ Peak Avg/Min Burst
VC Name      VPI/ VCI  Prec/Exp  Prec/Exp  PrecExp/ PV  Kbps   kbps  Cells Sts
                                           Accept

ABC-high       1/107   7         7         - / Yes  PV  10000   5000   32   UP
ABC-med        1/105   6         6         - / Yes  PV  10000              UP
ABC-low        1/102   5-0       5-0       - / Yes  -   10000              UP
```

Configuring Multi-VLAN to VC Support

For information on configuring multi-VLAN to VC support, see the "Configuring QoS for ATM VC Access Trunk Emulation" topic at http://www.cisco.rw/univercd/cc/td/doc/product/core/cis7600/cfgnotes/flexport/combo/flexqos.htm#wp1162305.

Configuring Link Fragmentation and Interleaving with Virtual Templates

The ATM SPA supports Link Fragmentation and Interleaving (LFI) with the distributed Compressed Real-Time Protocol (dCRTP). This allows the ATM interfaces, which are cell-based, to efficiently transport packet-based IP traffic without an excessive amount of bandwidth being used for packet headers and other overhead.

The LFI/dCRTP feature requires the use of multilink PPP (MLP), which can be implemented either by using virtual templates or dialer templates.

Link Fragmentation and Interleaving with Virtual Templates Configuration Guidelines

- The 1-Port OC-48c/STM-16 ATM SPA does not support LFI.
- A functional multilink PPP (MLP) bundle requires one virtual access interface operating as a PPP interface, and a second virtual access interface operating as a multilink PPP bundle interface.
- The Cisco IOS software supports a maximum of 1,000 virtual template interfaces per Catalyst 6500 Series switch.
- When LFI is configured on a PVC, the output packets counter in the **show atm pvc** command counts all fragments of a packet as a single packet, and does not display the actual number of fragmented packets that were output. For example, if a packet is fragmented into four fragments, the output

packets counter shows only one packet, not four. The output bytes counter is accurate, however, and you can also display the total number of fragmented packets on all PVCs on the interface with the **show interface atm** command.

- LFI supports three protocol formats: AAL5CISCOPP, AAL5MUX, and AAL5SNAP
- For fragmentation to function, a QoS service policy having a minimum of two QoS queues must be applied to the virtual template interface.

Link Fragmentation and Interleaving with Virtual Templates Configuration Task

To configure LFI with virtual templates, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface virtual-template`** *`number`*	Creates a virtual template and enters interface configuration mode. • *number*—Arbitrary value to identify this virtual template.
Step 2	`Router(config-if)#` **`bandwidth`** *`value`*	Specifies the bandwidth, in kbps, for the interfaces that use this virtual template: • *value*—Bandwidth, in kilobits per second, for the interface.
Step 3	`Router(config-if)#` **`service-policy input`** *`policy-name`*	Attaches the specified policy map to the input interface that uses this virtual template: • *policy-name*—Name of the policy map that was created by the **policy-map** command to be used.
Step 4	`Router(config-if)#` **`service-policy output`** *`policy-name`*	Attaches the specified policy map to the output interface that uses this virtual template: • *policy-name*—Name of the policy map that was created by the **policy-map** command to be used.
Step 5	`Router(config-if)#` **`ppp multilink`** `[`**`bap`**`]`	Enables multilink PPP (MLP) on the interfaces that use this virtual template: • **bap**—(Optional) Enables bandwidth allocation control negotiation and dynamic allocation of bandwidth on a link, using the bandwidth allocation protocol (BAP).
Step 6	`Router(config-if)#` **`ppp multilink fragment delay`** *`max-delay`*	(Optional) Configures the maximum delay for the transmission of a packet fragment on an MLP bundle. • *max-delay*—Maximum amount of time, in milliseconds, that should be required to transmit a fragment. The range is from 1 to 1000, with a default value of 30 for MLP bundles.
Step 7	`Router(config-if)#` **`ppp multilink interleave`**	Enables interleaving of the fragments of larger packets on an MLP bundle.
Step 8	`Router(config-if)#` **`interface atm`** *`slot/subslot/port.subinterface`* **`point-to-point`**	Creates the specified point-to-point subinterface and enters interface configuration mode.

	Command	Purpose
Step 9	`Router(config-if)#` **`pvc`** `[`*`name`*`]` *`vpi/vci`* `[`**`ilmi`** `|` **`qsaal`**`]`	Configures a new ATM PVC by assigning its VPI/VCI numbers and enters ATM VC configuration mode. The valid values for *vpi/vci* are: • *vpi*—Specifies the VPI ID. The valid range is 0 to 255. • *vci*—Specifies the VCI ID. The valid range is 1 to 65535. Values 1 to 31 are reserved and should not be used, except for 5 for the QSAAL PVC and 16 for the ILMI PVC. You can also configure the following options: • *name*—(Optional) An arbitrary string that identifies this PVC. • **ilmi**—(Optional) Configures the PVC to use ILMI encapsulation (default). • **qsaal**—(Optional) Configures the PVC to use QSAAL encapsulation.
	Note When using the **pvc** command, remember that the *vpi/vci* combination forms a unique identifier for the interface and all of its subinterfaces. If you specify a *vpi/vci* combination that has been used on another subinterface, the Cisco IOS software assumes that you want to modify that PVC’s configuration and automatically switches to its parent subinterface.	
Step 10	`Router(config-if-atm-vc)#` **`protocol ppp virtual-template`** *`number`*	Configures the PVC for PPP with the parameters from the specified virtual template.
Step 11	`Router(config-if-atm-vc)#` **`end`**	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Link Fragmentation and Interleaving with Virtual Templates Configuration

To verify a virtual template configuration, display the running configuration for the configured ATM and virtual interfaces:

```
Router# show running-config interface virtual-template 1

!
interface Virtual-Template1
Current configuration : 373 bytes
!
interface Virtual-Template1
bandwidth 300
ip address 23.0.0.1 255.255.255.0
ppp chap hostname template1
ppp multilink
ppp multilink fragment-delay 8
ppp multilink interleave
service-policy output lfiqos
!

Router# show running-config interface atm 6/0/1

!
interface ATM6/0/1
 atm idle-cell-format itu
 atm enable-payload-scrambling
 no atm ilmi-keepalive
 pvc 32/32
  vbr-rt 640 640 256
  encapsulation aal5snap
  protocol ppp Virtual-Template1
```

To display run-time statistics and other information about the currently configured multilink PPP bundles, use the **show ppp multilink** command:

```
Router# show ppp multilink

Virtual-Access3, bundle name is north-2
  Bundle up for 00:01:51
  Bundle is Distributed
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received, 1/255 load
  0x0 received sequence, 0x0 sent sequence
  Member links: 1 (max not set, min not set)
    Vi1, since 00:01:38, no frags rcvd, 62 weight, 54 frag size

dLFI statistics:
            DLFI Packets    Pkts In             Pkts Out
              Fragmented 4294967288             3129990
            UnFragmented    1249071                   0
             Reassembled    1249071             1564994

        Reassembly Drops          0
     Fragmentation Drops          0
        Out of Seq Frags          0
```

Note The **show ppp multilink** command displays only the packet counters, and not byte counters, for a dLFI configuration on an ATM SPA interface. Also, the number of fragmented packets shows the number of fragments sent to the SAR assembly, not the number of fragments that are placed on the ATM line. It is possible that the SAR assembly might drop some of these fragments on the basis of Layer 3 QoS limits.

Configuring the Distributed Compressed Real-Time Protocol

The distributed Compressed Real-Time Protocol (dCRTP) compresses the 40 bytes of the IP/UDP/RTP packet headers down to between only two and four bytes in a distributed fast-switching and distributed Cisco Express Forwarding (dCEF) network. This compression reduces the packet size, improves the speed of packet transmission, and reduces packet latency, especially on cell-based interfaces, such as ATM interfaces.

Distributed Compressed Real-Time Protocol Configuration Guidelines

When configuring dCRTP, consider the following guidelines:

- Distributed CEF switching or distributed fast switching must be enabled on the interface.
- PPP must be used on the interface or subinterface.

Distributed Compressed Real-Time Protocol Configuration Task

To enable and configure dCRTP on an ATM interface, virtual template interface, or a dialer template interface, perform this task beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)#` **`interface atm`** *`slot/subslot/port`*
	`or`
	`Router(config)#` **`interface virtual-template`** *`number`*
	`or`
	`Router(config)#` **`interface dialer`** *`number`*
		Enters interface configuration mode for an interface on the ATM SPA, or for a virtual template or dialer template interface.

Step 2	`Router(config-if)#` **`ip rcp header-compression`** `[`**`passive`**`]`	Enables RCP header compression. • **passive**—(Optional) Compresses outgoing RCP packets only if incoming RCP packets on the same interface are compressed. The default compresses all RCP packets on the interface.
Step 3	`Router(config-if)#` **`ip tcp header-compression`** `[`**`passive`**`]`	Enables TCP header compression. • **passive**—(Optional) Compresses outgoing TCP packets only if incoming TCP packets on the same interface are compressed. The default compresses all TCP packets on the interface.
	Note By default, RCP and TCP header compression are enabled on ATM interfaces when they are configured with an IP address. You do not need to give the **ip rcp header-compression** and **ip tcp header-compression** commands unless you have previously disabled these features, or you want to use the **passive** options.	
Step 4	`Router(config-if)#` **`ip rcp compression-connections`** *`number`*	Specifies the total number of RCP header compression connections that can be supported on the interface. • *number*—Number of RCP header compression connections. The valid range is 3 to 1000, with a default of 32 connections (16 calls).
Step 5	`Router(config-if)#` **`ip tcp compression-connections`** *`number`*	Specifies the total number of TCP header compression connections that can be supported on the interface. • *number*—Number of TCP header compression connections. The valid range is 3 to 1000, with a default of 32 connections (16 calls).
Step 6	`Router(config-if)#` **`end`**	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Distributed Compressed Real-Time Protocol Configuration

To verify the dCRTP of an ATM interface, use the **show running-config interface interface virtual-template** command:

```
Router# show running-config interface interface virtual-template 1

!
interface Virtual-Template1
 bandwidth 2320
 ip unnumbered Loopback2
 max-reserved-bandwidth 100
```

```
ip tcp header-compression
ppp multilink
ppp multilink fragment delay 4
ppp multilink interleave
ip rtp header-compression
```

Configuring Automatic Protection Switching

The ATM SPAs support 1+1 Automatic Protection Switching (APS) on PVCs as described in section 5.3 of the Telcordia publication *GR-253-CORE SONET Transport Systems: Common Generic Criteria*. APS redundancy is supported at the line layer, so that when an OC-3c, OC-12c, or OC-48c link fails, all of the PVCs that are carried by that link are switched simultaneously.

Note APS is not supported for SVCs.

In an APS configuration, a redundant ATM interface (the Protect interface) is configured for every active ATM interface (the Working interface). If the Working interface goes down, the Protect interface automatically switches over and continues communication over the interface's PVCs.

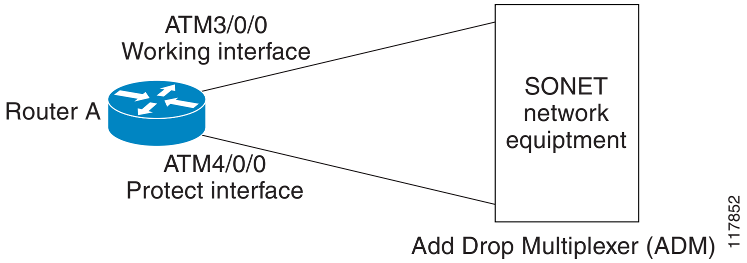
The APS Protect Group Protocol (PGP), which runs on top of User Datagram Protocol (UDP), provides communication between the Working and Protect interfaces. This communication occurs over a separate out-of-band (OOB) communication channel, such as an Ethernet link.

In the case of degradation, loss of channel signal, or manual intervention, the APS software on the Protect interface sends APS PGP commands to activate or deactivate the Working interface as necessary. If the communication channel between the Working and Protect interfaces is lost, the Working interface assumes full control, as if no Protect interface existed.

Note In the following figures, the devices with the ATM SPAs are shown as Cisco 7600 series routers, but they can also be Catalyst 6500 series switches.

Figure 7-4 shows a very simple example of a pair of Working and Protect interfaces on a single router.

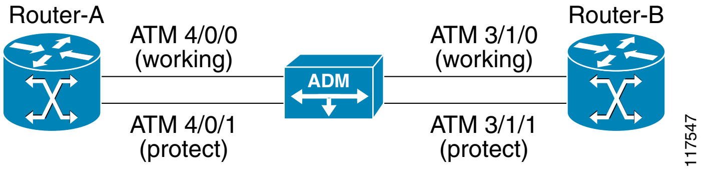
Figure 7-4 Basic Automatic Protection Switching Configuration



Tip If possible, use separate SPAs to provide the Working and Protect interfaces, as shown in Figure 7-4. This removes the SPA as a potential single point of failure, which would be the case if the same SPA provided both the Working and Protect interfaces.

Multiple switches can be using APS at the same time. For example, Figure 7-5 shows a simple example of two routers that each have one pair of Working and Protect interfaces. In this configuration, the two routers are independently configured.

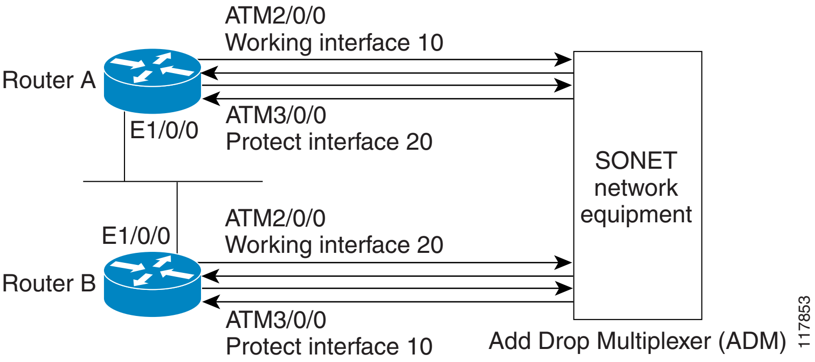
Figure 7-5 Sample Automatic Protection Switching Configuration with Multiple Routers



You can also configure multiple routers with APS so that interfaces on one router can provide protection for the interfaces on another router. This provides protection in case a router experiences a major system problem, such as a processor fault.

Figure 7-6 shows a basic example of two routers that each have one Working ATM interface. Each router also has one Protect interface that provides protection for the other router's Working interface. Note that this configuration requires a separate out-of-band (OOB) communication link between the two routers, which in this case is provided by the Ethernet network.

Figure 7-6 Sample Multiple Router Protection with Automatic Protection Switching



An APS configuration requires the following steps:

- Configure the Working interface with the desired IP addresses, subinterfaces, and PVCs. Also assign the interface to an APS group and designate it as the Working interface.
- Create a loopback circuit for communication between the Working and Protect interfaces. This is optional, because you can also use any valid IP address on the router. However, we recommend using a loopback interface because it is always up and provides connectivity between the two interfaces as long as any communication path exists between them.
- Configure the Protect interface with the same subinterfaces and PVCs that were configured on the Working interface. The Protect interface should also be configured with an IP address that is on the same subnet as the Working interface.

Tip Always configure the Working interface before the Protect interface, so as to prevent the Protect interface from becoming active and disabling the circuits on the Working interface.

Automatic Protection Switching Configuration Guidelines

When configuring APS, consider the following guidelines:

- The Working and Protect interfaces must be compatible (that is, both OC-3c or both OC-12c interfaces). The interfaces can be on the same SPA, different SPAs in the same router, or different SPAs in different routers.
- If using interfaces on different routers, the two routers must have a network connection other than the ATM connection (such as through an Ethernet LAN). Because the APS PGP is UDP traffic, this network connection should be reliable with a minimum number of hops.
- Configure the Working ATM interface with the desired IP addresses and other parameters, as described in the "Required Configuration Tasks" section on page 7-2 and the "Configuring SONET and SDH Framing" section on page 7-56.
- Configure the desired PVCs on the Working interface, as described in the different procedures that are listed in the "Creating a Permanent Virtual Circuit" section on page 7-6.
- The IP addresses on the Working and Protect interfaces should be in the same subnet.
- APS is not supported on SVCs.

Automatic Protection Switching Configuration Task

To configure the Working and Protect interfaces on the ATM SPAs for basic APS operation, perform this task beginning in global configuration mode. For complete information on APS, including information on additional APS features, refer to the "Configuring Serial Interfaces" chapter in the *Cisco IOS Interface Configuration Guide, Release 12.2*.

	Command or Action	Purpose
Step 1	`Router(config)# interface loopback interface-number`	Creates a loopback interface and enters interface configuration mode: • *interface-number*—An arbitrary value from 0 to 2,147,483,647 that uniquely identifies this loopback interface.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies the IP address and subnet mask for this loopback interface. If the Working and Protect interfaces are on the same router, this IP address should be in the same subnet as the Working interface. If the Working and Protect interfaces are on different routers, this IP address should be in the same subnet as the Ethernet interface that provides the connectivity between the two routers. Repeat this command with the **secondary** keyword to specify additional IP addresses to be used for this interface.
Step 3	`Router(config-if)# interface atm slot/subslot/port`	Enters interface configuration mode for the Working interface on the ATM SPA.
Step 4	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies the IP address and subnet mask for the Working interface. Repeat this command with the **secondary** keyword to specify additional IP addresses to be used for the interface.

	Command or Action	Purpose
Step 5	Router(config-if)# **aps group** *group-number*	Enables the use of the APS Protect Group Protocol for this Working interface. • *group-number*—Unique number identifying this pair of Working and Protect interfaces. **Note** The **aps group** command is optional if this is the only pair of Working and Protect interfaces on the router, but is required when you configure more than one pair of Working and Protect interfaces on the same router.
Step 6	Router(config-if)# **aps working** *circuit-number*	Identifies the interface as the Working interface. • *circuit-number*—Identification number for this particular channel in the APS pair. Because only 1+1 redundancy is supported, the only valid values are 0 or 1, and the Working interface defaults to 1.
Step 7	Router(config-if)# **aps authentication** *security-string*	(Optional) Specifies a security string that must be included in every OOB message sent between the Working and Protect interfaces. • *security-string*—Arbitrary string to be used as a password between the Working and Protect interfaces. This string must match the one configured on the Protect interface.
Step 8	Router(config-if)# **interface atm** *slot/subslot/port*	Enters interface configuration mode for the Protect interface on the ATM SPA.
Step 9	Router(config-if)# **ip address** *ip-address mask* [**secondary**]	Specifies the IP address and subnet mask for the Protect interface. **Note** This should be the same address that was configured on the Working interface in Step 4. Repeat this command with the **secondary** keyword to specify additional IP addresses to be used for the interface. These should match the secondary IP addresses that are configured on the Working interface.
Step 10	Router(config-if)# **aps group** *group-number*	Enables the use of the APS Protect Group Protocol for this Protect interface. • *group-number*—Unique number identifying this pair of Working and Protect interfaces. **Note** The **aps group** command is optional if this is the only pair of Working and Protect interfaces on the router, but is required when you configure more than one pair of Working and Protect interfaces on the same router.

	Command or Action	Purpose
Step 11	Router(config-if)# **aps protect** *circuit-number ip-address*	Identifies this interface as the Protect interface: • *circuit-number*—Identification number for this particular channel in the APS pair. Because only 1+1 redundancy is supported, the only valid values are 0 or 1, and the Protect interface defaults to 0. • *ip-address*—IP address for the loopback interface that was configured in Step 2. The Protect interface uses this IP address to communicate with the Working interface. **Note** If you do not want to use a loopback interface for this configuration, this IP address should be the address of the Working interface if the Protect and Working interfaces are on the same router. If the Working and Protect interfaces are on different routers, this should be the IP address of the Ethernet interface that provides interconnectivity between the two routers.
Step 12	Router(config-if)# **aps authentication** *security-string*	(Optional) Specifies a security string that must be included in every OOB message sent between the Working and Protect interfaces. • *security-string*—Arbitrary string to be used as a password between the Working and Protect interfaces. This string must match the one configured on the Working interface.
Step 13	Router(config-if)# **aps revert** *minutes*	(Optional) Enables the Protect interface to automatically switch back to the Working interface after the Working interface has been up for a specified number of minutes. • *minutes*—Number of minutes until the interface is switched back to the Working interface after the Working interface comes back up. **Note** If this command is not given, you must manually switch back to the Working interface using either the **aps force** *circuit-number* or the **aps manual** *circuit-number* command.
Step 14	Router(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Automatic Protection Switching Configuration

To verify the APS configuration on the router, use the **show aps** command without any options. The following example shows a typical configuration in which the Working interface is the active interface:

```
Router# show aps

ATM4/0/1 APS Group 1: protect channel 0 (inactive)
        bidirectional, revertive (2 min)
        PGP timers (default): hello time=1; hold time=3
        state:
        authentication = (default)
        PGP versions (native/negotiated): 2/2
        SONET framing; SONET APS signalling by default
        Received K1K2: 0x00 0x05
                No Request (Null)
        Transmitted K1K2: 0x20 0x05
                Reverse Request (protect)
        Working channel 1 at 10.10.10.41 Enabled
        Remote APS configuration: (null)

ATM4/0/0 APS Group 1: working channel 1 (active)
        PGP timers (from protect): hello time=3; hold time=6
        state: Enabled
        authentication = (default)
        PGP versions (native/negotiated): 2/2
        SONET framing; SONET APS signalling by default
        Protect at 10.10.10.41
        Remote APS configuration: (null)
```

The following sample output is for the same interfaces, except that the Working interface has gone down and the Protect interface is now active:

```
Router# show aps

ATM4/0/1 APS Group 1: protect channel 0 (active)
        bidirectional, revertive (2 min)
        PGP timers (default): hello time=1; hold time=3
        state:
        authentication = (default)
        PGP versions (native/negotiated): 2/2
        SONET framing; SONET APS signalling by default
        Received K1K2: 0x00 0x05
                No Request (Null)
        Transmitted K1K2: 0xC1 0x05
                Signal Failure - Low Priority (working)
        Working channel 1 at 10.10.10.41 Disabled SF
        Pending local request(s):
                0xC (, channel(s) 1)
        Remote APS configuration: (null)

ATM4/0/0 APS Group 1: working channel 1 (Interface down)
        PGP timers (from protect): hello time=3; hold time=6
        state: Disabled
        authentication = (default)
        PGP versions (native/negotiated): 2/2
        SONET framing; SONET APS signalling by default
        Protect at 10.10.10.41
        Remote APS configuration: (null)
```

Tip To obtain APS information for a specific ATM interface, use the **show aps atm** *slot/subslot/port* command. To display information about the APS groups that are configured on the router, use the **show aps group** command.

Configuring SONET and SDH Framing

The default framing on the ATM OC-3c and OC-12c SPAs is SONET, but the interfaces also support SDH framing.

Note In ATM environments, the key difference between SONET and SDH framing modes is the type of cell transmitted when no user or data cells are available. The ATM forum specifies the use of idle cells when unassigned cells are not being generated. More specifically, in Synchronous Transport Module-X (STM-X) mode, an ATM interface sends idle cells for cell-rate decoupling. In Synchronous Transport Signal-Xc (STS-Xc) mode, the ATM interface sends unassigned cells for cell-rate decoupling.

To change the framing type and configure optional parameters, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface atm** *slot/subslot/port*	Enters interface configuration mode for the indicated port on the specified ATM SPAs.
Step 2	Router(config-if)# **atm clock internal**	(Optional) Configures the interface to use its own internal (onboard) clock to clock transmitted data. The default (**no atm clock internal**) configures the interface to use the transmit clock signal that is recovered from the receive data stream, allowing the switch to provide the clocking source.
Step 3	Router(config-if)# **atm framing** {**sdh** | **sonet**}	(Optional) Configures the interface for either SDH or SONET framing. The default is SONET.
Step 4	Router(config-if)# [**no**] **atm sonet report** {**all** | **b1-tca** | **b2-tca** | **b3-tca** | **default** | **lais** | **lrdi** | **pais** | **plop** | **pplm** | **prdi** | **ptim** | **puneq** | **sd-ber** | **sf-ber** | **slof** | **slos**}	(Optional) Enables ATM SONET alarm reporting on the interface. The default is for all reports to be disabled. You can enable an individual alarm, or you can enable all alarms with the **all** keyword. **Note** This command also supports a **none** [**ignore**] option, which cannot be used with any of the other options. See the "Configuring for Transmit-Only Mode" section on page 7-57 for details.
Step 5	Router(config-if)# **no**] **atm sonet-threshold** {**b1-tca** *value* | **b2-tca** *value* | **b3-tca** *value* | **sd-ber** *value* | **sf-ber** *value*}	(Optional) Configures the BER threshold values on the interface. The value specifies a negative exponent to the power of 10 (10 to the power of minus *value*) for the threshold value. The default values are the following: • **b1-tca** = 6 (10e–6) • **b2-tca** = 6 (10e–6) • **b3-tca** = 6 (10e–6) • **sd-ber** = 6 (10e–6) • **sf-ber** = 3 (10e–3)
Step 6	Router(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the SONET and SDH Framing Configuration

To verify the framing configuration, use the **show controllers atm** command:

```
Router# show controllers atm 5/0/1

Interface ATM5/0/1 is up
 Framing mode: SONET OC3 STS-3c

SONET Subblock:
SECTION
  LOF = 0          LOS    = 0                            BIP(B1) = 603
LINE
  AIS = 0          RDI    = 2          FEBE = 2332       BIP(B2) = 1018
PATH
  AIS = 0          RDI    = 1          FEBE = 28         BIP(B3) = 228
  LOP = 0          NEWPTR = 0          PSE  = 1          NSE     = 2

Active Defects: None
Active Alarms:  None
Alarm reporting enabled for: LOF LOS B1-TCA B2-TCA SF LOP B3-TCA

ATM framing errors:
  HCS (correctable):   0
  HCS (uncorrectable): 0

APS

  COAPS = 0          PSBF = 0
  State: PSBF_state = False
  Rx(K1/K2): 00/00  Tx(K1/K2): 00/00
  Rx Synchronization Status S1 = 00
  S1S0 = 00, C2 = 00

PATH TRACE BUFFER : STABLE

BER thresholds:  SF = 10e-3  SD = 10e-6
TCA thresholds:  B1 = 10e-7 B2 = 10e-6  B3 = 10e-6

  Clock source:  line
```

Configuring for Transmit-Only Mode

The ATM SPAs support operation in a transmit-only mode, where a receive fiber does not need to be connected. This mode is typically used for one-way applications, such as video-on-demand.

By default, the lack of a receive path generates continuous framing errors, which bring the ATM interface down. To prevent this, you must configure the ATM interface to disable and ignore all ATM SONET alarms. The 1-Port OC-48c/STM-16 ATM SPA default framing is sonet

Note This configuration violates the ATM specifications for alarm reporting.

Transmit-Only Mode Configuration Guidelines

When an ATM interface has been configured to ignore ATM SONET alarms, you cannot configure an IP address (or other Layer 3 parameter) on the interface. Similarly, you must remove all IP addresses (and all other Layer 3 parameters) from the interface before beginning this procedure.

Transmit-Only Mode Configuration Task

To configure the ATM interface to disable and ignore all ATM SONET alarms, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface atm`** *`slot/subslot/port[.subinterface]`*	Enters interface (or subinterface) configuration mode for the indicated port on the specified ATM SPA.
Step 2	`Router(config-if)#` **`no ip address`** *`ip-address mask`*	Removes the IP address that is assigned to this interface (if one has been configured). All IP and other Layer 3 configurations must be removed from the interface before ATM SONET alarms can be ignored.
Step 3	`Router(config-if)#` **`atm sonet report none ignore`**	Disables the generation of all ATM SONET alarms, and instructs the ATM interface to remain up and operational when such alarm conditions exist.
Step 4	`Router(config-if)#` **`end`**	Exits interface configuration mode and returns to privileged EXEC mode.

Saving the Configuration

To save your running configuration to nonvolatile random-access memory (NVRAM), use the following command in privileged EXEC configuration mode:

Command	Purpose
`Router#` **`copy running-config startup-config`**	Writes the new configuration to NVRAM.

Note To permanently save your configuration changes, you must write them to the nonvolatile RAM (NVRAM) by entering the **copy running-config startup-config** command in privileged EXEC mode.

For more information about managing configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

Shutting Down and Restarting an Interface on a SPA

Shutting down an interface puts it into the administratively down mode and takes it offline, stopping all traffic that is passing through the interface. Shutting down an interface, though, does not change the interface configuration.

As a general rule, you do not need to shut down an interface if you are removing it and replacing it with the same exact model of SPA in an online insertion and removal (OIR) operation. However, we recommend shutting down an interface whenever you are performing one of the following tasks:

- When you do not need to use the interface in the network.
- Preparing for future testing or troubleshooting.

- Changing the interface configuration in a way that would affect the traffic flow, such as changing the encapsulation.
- Changing the interface cables.
- Removing a SPA that you do not expect to replace.
- Replacing the SIP with another type of SIP (such as replacing a Cisco 7600 SIP-200 with a Cisco 7600 SIP-400.
- Replacing an interface card with a different model of card.

Shutting down the interface in these situations prevents anomalies from occurring when you reinstall the new card or cables. It also reduces the number of error messages and system messages that might otherwise appear.

Tip If you are planning on physically removing the SPA from the SIP, also shut down the SPA, using the procedure given in the "Shutting Down an ATM Shared Port Adapter" section on page 7-60.

Note If you plan to replace an existing ATM port adapter with an ATM SPA in the Catalyst 6500 Series switch and want to use the same configuration, save the slot's configuration before physically replacing the hardware. This is because all slot configuration is lost when you replace one card type with another card type, even if the two cards are functionally equivalent. You can then reenter the previous configuration after you have inserted the ATM SPA.

To shut down an interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`interface atm`** *`slot/subslot/port`*	Enters interface configuration mode for the indicated port on the specified ATM SPA.
Step 2	`Router(config-if)# `**`shutdown`**	Shuts down the interface.
	Repeat Step 1 and Step 2 for each interface to be shut down.	
Step 3	`Router(config-if)# `**`end`**	Exits interface configuration mode and returns to privileged EXEC mode.

Tip When you shut down an interface, the **show interface** command indicates that the interface is administratively down until the SPA is physically removed from the chassis or until the SPA is re-enabled.

The following shows a typical example of shutting down an ATM SPA interface:

```
Router> enable
Router# configure terminal
Router(config)# interface atm 4/0/0
Router(config-if)# shutdown
Router(config-if)# end
Router# show interface atm 4/0/0

ATM4/0/0 is administratively down, line protocol is down
  Hardware is SPA-4XOC3-ATM, address is 000d.2959.d5ca (bia 000d.2959.d5ca)
  Internet address is 10.10.10.16/24
```

```
MTU 4470 bytes, sub MTU 4470, BW 599040 Kbit, DLY 80 usec,
   reliability 255/255, txload 42/255, rxload 1/255
Encapsulation ATM, loopback not set
Encapsulation(s): AAL5
4095 maximum active VCs, 1 current VCCs
VC idle disconnect time: 300 seconds
0 carrier transitions
Last input 01:01:16, output 01:01:16, output hang never
Last clearing of "show interface" counters 01:10:21
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
30 second input rate 0 bits/sec, 0 packets/sec
30 second output rate 702176000 bits/sec, 1415679 packets/sec
   1000 packets input, 112000 bytes, 0 no buffer
   Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   2948203354 packets output, 182788653886 bytes, 0 underruns
   0 output errors, 0 collisions, 0 interface resets
   0 output buffer failures, 0 output buffers swapped out
```

Shutting Down an ATM Shared Port Adapter

Shutting down an ATM SPA shuts down all ATM interfaces on the SPA, and puts the SPA and its interfaces into the administratively down state. This takes all interfaces offline, stopping all traffic that is passing through the SPA. Shutting down an ATM SPA, though, does not change the configuration of the SPA and its interfaces.

As a general rule, you do not need to shut down an ATM SPA if you are removing it and replacing it with the same exact model of SPA, in an online insertion and removal (OIR) operation. However, you should shut down the ATM SPA whenever you are performing one of the following tasks:

- Removing an interface that you do not expect to replace.
- Replacing the SIP with another type of SIP (such as replacing a Cisco 7600 SIP-200 with a Cisco 7600 SIP-400).
- Replacing the ATM SPA with a different model of SPA.

To shut down the ATM SPA, perform this task beginning in global configuration mode:

<table>
<tr><th></th><th>Command</th><th>Purpose</th></tr>
<tr><td><b>Step 1</b></td><td><code>Router(config)# <b>hw-module subslot</b> <i>slot/subslot</i> <b>shutdown</b> [<b>powered</b> | <b>unpowered</b>]</code></td><td>Shuts down the ATM SPA.
• <b>powered</b>—(Optional) Shuts down the ATM SPA and leaves it in the reset state. This is the default and is typically done when you want to shut down the SPA but leave it physically installed and cabled in the Catalyst 6500 Series switch.
• <b>unpowered</b>—(Optional) Shuts down the ATM SPA and leaves it in the unpowered state. Typically, this is done before removing the ATM SPA card from the chassis.
<b>Note</b> Repeat this step for each ATM SPA to be shut down.</td></tr>
<tr><td></td><td colspan="2">The <b>hw-module subslot shutdown</b> command can be given in both the global configuration and privileged EXEC modes. If this command is given in global configuration mode, it can be saved to the startup configuration so that it is automatically executed after each reload of the switch. If given in privileged EXEC mode, the command takes effect immediately, but it is not saved to the configuration. In either case, the <b>hw-module subslot shutdown</b> command remains in effect during the current session of the Catalyst 6500 Series switch until it is reversed using the <b>no</b> form of the command.</td></tr>
<tr><td><b>Step 2</b></td><td><code>Router(config)# <b>end</b></code></td><td>Exits configuration mode and returns to privileged EXEC mode.</td></tr>
</table>

The following shows a typical example of shutting down ATM SPAs. In this example, the SPA in subslot 0 is put into the reset mode, while the SPA in subslot 1 is powered down.

```
Router> enable
Router# hw-module subslot 4/0 shutdown powered
Router# hw-module subslot 4/1 shutdown unpowered
Router#
```

Tip The ATM SPA remains shut down, even after a new card is installed or after a reset of the Catalyst 6500 Series switch, until you re-enable the SPA using the **no hw-module subslot shutdown** command.

Verifying the Interface Configuration

See the following sections to obtain configuration and operational information about the ATM SPA and its interfaces:

- Verifying Per-Port Interface Status, page 7-62
- Monitoring Per-Port Interface Statistics, page 7-62

For additional information on using these and other commands to obtain information about the configuration and operation of the ATM SPA and interfaces, see Chapter 8, “Troubleshooting the ATM Shared Port Adapter.”

Verifying Per-Port Interface Status

Use the **show interfaces atm** command to display detailed status information about an interface port in an ATM SPA that is installed in the Catalyst 6500 Series switch. The following example provides sample output for interface port 1 (the second port) on the ATM SPA that is located in subslot 0 (the left-most subslot), of the SIP that is installed in slot 3 of a Catalyst 6500 Series switch:

```
Router# show interface atm 3/0/1

ATM3/0/1 is up, line protocol is up
  Hardware is SPA-4XOC3-ATM, address is 000a.f330.7dc0 (bia 000a.f330.7dca)
  Internet address is 10.13.21.31/24
  MTU 4470 bytes, sub MTU 4470, BW 599040 Kbit, DLY 80 usec,
     reliability 255/255, txload 140/255, rxload 129/255
  Encapsulation ATM, loopback not set
  Encapsulation(s): AAL5
  4095 maximum active VCs, 1 current VCCs
  VC idle disconnect time: 300 seconds
  0 carrier transitions
  Last input never, output never, output hang never
  Last clearing of "show interface" counters 00:45:35
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 304387000 bits/sec, 396342 packets/sec
  5 minute output rate 329747000 bits/sec, 396334 packets/sec
     1239456438 packets input, 118987818048 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1239456287 packets output, 128903453848 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Monitoring Per-Port Interface Statistics

Use the **show controllers atm** command to display detailed status and statistical information on a per-port basis for an ATM SPA. The following example provides sample output for interface port 0 (the first port) on the ATM SPA that is located in subslot 0 (the left-most subslot) of the SIP that is installed in slot 4 of a Catalyst 6500 Series switch:

```
Router# show controllers atm 4/0/0

Interface ATM4/0/0 is up
 Framing mode: SONET OC3 STS-3c

SONET Subblock:
SECTION
  LOF = 0          LOS    = 0                            BIP(B1) = 603
LINE
  AIS = 0          RDI    = 2          FEBE = 2332       BIP(B2) = 1018
PATH
  AIS = 0          RDI    = 1          FEBE = 28         BIP(B3) = 228
  LOP = 0          NEWPTR = 0          PSE  = 1          NSE     = 2

Active Defects: None
Active Alarms:  None
Alarm reporting enabled for: SF SLOS SLOF B1-TCA B2-TCA PLOP B3-TCA

ATM framing errors:
  HCS (correctable):   0
  HCS (uncorrectable): 0

APS

  COAPS = 0          PSBF = 0
  State: PSBF_state = False
  Rx(K1/K2): 00/00  Tx(K1/K2): 00/00
  Rx Synchronization Status S1 = 00
  S1S0 = 00, C2 = 00

PATH TRACE BUFFER : STABLE
  Remote hostname : fecao7609_2
  Remote interface: ATM9/0/0
  Remote IP addr  : 0.0.0.0
  Remote Rx(K1/K2): 00/00  Tx(K1/K2): 00/00


BER thresholds:  SF = 10e-3  SD = 10e-6
TCA thresholds:  B1 = 10e-6  B2 = 10e-6  B3 = 10e-6

  Clock source:  line
```

Configuration Examples

This section includes the following configuration examples for the ATM SPAs:

- Basic Interface Configuration Example, page 7-64
- MTU Configuration Example, page 7-64
- Permanent Virtual Circuit Configuration Example, page 7-65
- PVC on a Point-to-Point Subinterface Configuration Example, page 7-65
- PVC on a Multipoint Subinterface Configuration Example, page 7-66
- RFC 1483 Bridging for PVCs Configuration Example, page 7-67
- RFC 1483 Bridging for PVCs with IEEE 802.1Q Tunneling Configuration Example, page 7-68
- ATM RFC 1483 Half-Bridging Configuration Example, page 7-68
- ATM Routed Bridge Encapsulation Configuration Example, page 7-68
- ATM RFC 1483 Half-Bridging Configuration Example, page 7-68
- ATM Routed Bridge Encapsulation Configuration Example, page 7-68
- Precedence-Based Aggregate WRED Configuration Example, page 7-69
- DSCP-Based Aggregate WRED Configuration Example, page 7-70
- Switched Virtual Circuits Configuration Example, page 7-70
- Traffic Parameters for PVCs or SVCs Configuration Example, page 7-71
- Virtual Circuit Classes Configuration Example, page 7-72
- Virtual Circuit Bundles Configuration Example, page 7-72
- Link Fragmentation and Interleaving with Virtual Templates Configuration Example, page 7-73
- Distributed Compressed Real-Time Protocol Configuration Example, page 7-74
- Automatic Protection Switching Configuration Example, page 7-75

- SONET and SDH Framing Configuration Example, page 7-75
- Layer 2 Protocol Tunneling Topology with a Cisco 7600, Catalyst 5500, and Catalyst 6500 Configuration Example, page 7-76
- Layer 2 Protocol Tunneling Topology with a Cisco 7600 and Cisco 7200 Configuration Example, page 7-77
- Cisco 7600 Basic Back-to-Back Configuration Example, page 7-78
- Catalyst 5500 Switch and Cisco 7600 Series Routers in Back-to-Back Topology Configuration Example, page 7-78
- Cisco 7600 and Cisco 7200 in Back-to-Back Topology Configuration Example, page 7-79

Basic Interface Configuration Example

```
!
interface ATM5/1/0
 mtu 9216
 no ip address
 atm clock INTERNAL
!
interface ATM5/1/0.1 point-to-point
 mtu 9216
 ip address 70.1.1.1 255.255.0.0
 pvc 52/100
!
!
interface ATM5/1/1
 mtu 9216
 no ip address
 atm clock INTERNAL
!
interface ATM5/1/1.1 point-to-point
 mtu 9216
 ip address 70.2.1.1 255.255.0.0
 pvc 53/100
!
!
interface ATM5/1/2
 no ip address
 atm clock INTERNAL
!
interface ATM5/1/3
 no ip address
 atm clock INTERNAL
!
```

MTU Configuration Example

```
!
interface ATM4/1/0
 ip address 192.168.100.13 255.255.255.0
 mtu 9216
 ip mtu 9188
 mpls mtu 9288
 atm clock INTERNAL
!
```

Permanent Virtual Circuit Configuration Example

```
!
interface ATM5/0/0
 no ip address
 pvc 1/100
  protocol ip 1.1.1.3
  protocol ip 20.1.1.1
  broadcast
!
!
interface ATM5/0/1
 no ip address
!
interface ATM5/1/1
 ip address 1.1.1.1 255.255.255.0
 load-interval 30
 pvc 1/100
  protocol ip 1.1.1.3
  protocol ip 20.1.1.1
  cbr 140000
  broadcast
  oam-pvc manage
!
 pvc 1/101
  protocol ip 9.9.9.2
  encapsulation aal5ciscoppp Virtual-Template1
 !
```

PVC on a Point-to-Point Subinterface Configuration Example

The following example shows a simple configuration of several PVCs that are configured on point-to-point subinterfaces:

```
interface ATM3/1/0
 no ip address
!
interface ATM3/1/0.1 point-to-point
 pvc 4/44 l2transport
  mpls l2transport route 22.22.22.22 400
!
!
interface ATM3/1/0.2 point-to-point
 pvc 5/55 l2transport
  encapsulation aal0
  mpls l2transport route 22.22.22.22 500
 !
!
interface ATM3/1/0.3 point-to-point
 ip address 99.0.0.2 255.0.0.0
 pvc 9/99
 !
!
interface ATM5/0/0
 description flexwan_6_0_0
 no ip address
 logging event link-status
 atm clock INTERNAL
!
interface ATM5/0/0.1 point-to-point
```

```
 ip address 50.1.1.1 255.255.255.0
 pvc 50/11
!
!
interface ATM5/0/0.2 point-to-point
 ip address 50.2.2.1 255.255.255.0
 pvc 50/12
!
!
interface ATM5/0/0.3 point-to-point
 ip address 50.3.3.1 255.255.255.0
 pvc 50/13
 !
!
interface ATM5/0/0.4 point-to-point
 ip address 50.4.4.1 255.255.255.0
 pvc 50/14
 !
!
interface ATM5/0/0.5 point-to-point
 ip address 50.5.5.1 255.255.255.0
 pvc 50/15
 !
!
interface ATM5/1/0.1 point-to-point
 ip address 2.0.0.2 255.255.255.0
!
interface ATM5/1/0.2 point-to-point
 ip address 2.0.1.2 255.255.255.0
!
interface ATM5/1/0.3 point-to-point
 ip address 39.0.0.1 255.0.0.0
!
```

PVC on a Multipoint Subinterface Configuration Example

```
!
interface ATM4/1/0
 no ip address
 atm clock INTERNAL
!
interface ATM4/1/0.2 multipoint
 ip address 1.1.1.1 255.0.0.0
 pvc 0/121
  protocol ip 1.1.1.23 broadcast
  vbr-nrt 2358 2358
  encapsulation aal5snap
!
 pvc 0/122
  protocol ip 1.1.1.24 broadcast
  vbr-nrt 2358 2358
  encapsulation aal5snap
 !
 pvc 0/123
  protocol ip 1.1.1.25 broadcast
  vbr-nrt 2358 2358
  encapsulation aal5snap
!
 pvc 0/124
  protocol ip 1.1.1.26 broadcast
  vbr-nrt 2358 2358
  encapsulation aal5snap
!
 pvc 0/125
  protocol ip 1.1.1.27 broadcast
!
...
interface ATM5/1/1
 ip address 1.1.1.1 255.255.255.0
 load-interval 30
 pvc 1/100
  protocol ip 1.1.1.3
  protocol ip 20.1.1.1
  cbr 140000
  broadcast
  oam-pvc manage
!
 pvc 1/101
  protocol ip 9.9.9.2
  encapsulation aal5ciscoppp Virtual-Template1
!
!
interface ATM5/1/1.200 multipoint
 ip address 7.7.7.1 255.255.255.0
 bundle bundle
  pvc-bundle high 2/100
   class-vc high
  pvc-bundle med 2/101
   class-vc med
  pvc-bundle low 2/102
   class-vc low
!
!
interface ATM5/1/2
 no ip address
!
interface ATM5/1/3
 no ip address
!
```

RFC 1483 Bridging for PVCs Configuration Example

The following shows a simple example of an ATM interface and PVC that have been configured for RFC 1483 bridging with a Fast Ethernet interface:

```
vlan 30
!
interface FastEthernet7/1
 no ip address
 duplex full
 speed 100
 switchport
 switchport access vlan 30
 switchport mode access
!
interface ATM9/1/0
 no ip address
 mtu 4096
  bandwidth 2000
  pvc 0/39
 bridge-domain 30
 encapsulation aal5snap
```

```
!
interface ATM9/1/0.2 point-to-point
 ip address 10.10.12.2 255.255.255.0
 ip access-group rbe-list in
 atm route-bridged ip
 no mls ip
 pvc 10/200
!
router rip
 network 10.0.0.0
 network 30.0.0.0
!
```

RFC 1483 Bridging for PVCs with IEEE 802.1Q Tunneling Configuration Example

The following shows a simple example of an ATM interface that has been configured for RFC 1483 bridging using IEEE 802.1Q tunneling:

```
interface ATM6/2/0
 no ip address
 shutdown
 atm clock INTERNAL
 atm mtu-reject-call
 no atm ilmi-keepalive
 pvc 2/101
  bridge-domain 99 dot1q-tunnel
!
 mls qos trust dscp
 spanning-tree bpdufilter enable
```

ATM RFC 1483 Half-Bridging Configuration Example

The following simple example shows an ATM subinterface configured for half-bridging:

```
!
interface ATM5/1/0.100 multipoint
  ip address 192.168.100.14 255.255.0.0
  mtu 1500
  pvc 10/200
   encapsulation aal5snap bridge
!
```

ATM Routed Bridge Encapsulation Configuration Example

The following simple example shows an ATM subinterface configured for RBE, also known as RFC 1483 half-bridging:

```
!
interface ATM5/1/0.100 point-to-point
  ip address 10.10.10.121 255.255.0.0
  mtu 1500
  atm route-bridged ip
  pvc 100/100
   encapsulation aal5snap
!
```

Precedence-Based Aggregate WRED Configuration Example

The following example shows a precedence-based aggregate WRED configuration:

```
! Create a policy map named prec-aggr-wred.
!
Router(config)# policy-map prec-aggr-wred
!
! Configure a default class for the policy map.
!
Router(config-pmap)# class class-default
!
! Enable precedence-based (the default setting) aggregate WRED for the default class.
!
Router(config-pmap-c)# random-detect aggregate
!
! Define an aggregate subclass for packets with IP Precedence values of 0-3 and assign the
! WRED profile parameter values for this subclass.
!
Router(config-pmap-c)# random-detect precedence values 0 1 2 3 minimum thresh 10
maximum-thresh 100 mark-prob 10
!
! Define an aggregate subclass for packets with IP Precedence values of 4 and 5 and assign
! the WRED profile parameter values for this subclass.
!
Router(config-pmap-c)# random-detect precedence values 4 5 minimum-thresh 40
maximum-thresh 400 mark-prob 10
!
! Define an aggregate subclass for packets with an IP Precedence value of 6 and assign the
! WRED profile parameter values for this subclass.
!
Router(config-pmap-c)# random-detect precedence values 6 minimum-thresh 60 maximum-thresh
600 mark-prob 10
!
! Define an aggregate subclass for packets with an IP Precedence value of 7 and assign the
! WRED profile parameter values for this subclass.
!
Router(config-pmap-c)# random-detect precedence values 7 minimum-thresh 70 maximum-thresh
700 mark-prob 10
!
! Attach the policy map prec-aggr-wred to the interface. Note all ATM SPA service policies
! are applied at the atm vc level.
!
Router(config-pmap-c)# interface ATM4/1/0.10 point-to-point
Router(config-subif)# ip address 10.0.0.2 255.255.255.0
Router(config-subif)# pvc 10/110
Router(config-subif)# service policy output prec-aggr-wred
```

DSCP-Based Aggregate WRED Configuration Example

The following example shows a DSCP-based aggregate WRED configuration:

```
! Create a policy map named dscp-aggr-wred.
!
Router(config)# policy-map dscp-aggr-wred
!
! Configure a default class for the policy map.
!
Router(config-pmap)# class class-default
!
! Enable dscp-based aggregate WRED for the default class and assign the
! default WRED profile parameter values to be used for all subclasses that have not been
! specifically configured..
!
Router(config-pmap-c)# random-detect dscp-based aggregate minimum-thresh 1 maximum-thresh
10 mark-prob 10
!
! Define an aggregate subclass for packets with DSCP values of 0-7 and assign the WRED
! profile parameter values for this subclass
!
Router(config-pmap-c)# random-detect dscp values 0 1 2 3 4 5 6 7 minimum-thresh 10
maximum-thresh 20 mark-prob 10
!
! Define an aggregate subclass for packets with DSCP values of 8-11 and assign the WRED
! profile parameter values for this subclass.
!
Router(config-pmap-c)# random-detect dscp values 8 9 10 11 minimum-thresh 10
maximum-thresh 40 mark-prob 10
!
! Attach the policy map dscp-aggr-wred to the interface. Note all ATM SPA service policies
! are applied at the atm vc level.
!
Router(config)# interface ATM4/1/0.11 point-to-point
Router(config-subif)# ip address 10.0.0.2 255.255.255.0
Router(config-subif) pvc 11/101
Router(config-subif)# service policy output dscp-aggr-wred
```

Switched Virtual Circuits Configuration Example

```
interface ATM4/0/2
 ip address 10.23.33.2 255.255.255.0
 atm clock INTERNAL
 atm pvp 244
 atm esi-address 111111111111.11
 pvc 0/5 qsaal
!
 pvc 0/16 ilmi
!
!
interface ATM4/0/2.1 multipoint
 ip address 10.20.0.2 255.0.0.0
 atm esi-address 333333333333.33
!
 svc nsap 47.0091810000000001011B8C601.222222222222.22
  protocol ip 10.20.0.1
  ubr 1000
!
!
interface ATM4/0/2.2 multipoint
```

```
 ip address 10.13.3.1 255.255.255.0
 atm esi-address 510211111111.11
!
 svc nsap 47.009181000000001011B8C601.410233333333.33
  protocol ip 10.13.3.3
!
interface ATM4/0/2.3 multipoint
 svc SVC1 nsap 47.009181000000BBBBBB000001.222222222222.22
  protocol ip 33.33.33.1
  broadcast
  encapsulation aal5snap
```

Traffic Parameters for PVCs or SVCs Configuration Example

```
!
interface ATM5/1/1.100 point-to-point
 ip address 10.1.1.1 255.255.255.0
 load-interval 30
 pvc 1/100
  protocol ip 1.1.1.3
  protocol ip 20.1.1.1
  cbr 100
  broadcast
!
!
interface ATM5/1/1.110 point-to-point
 ip address 10.2.2.2 255.255.255.0
 pvc 1/110
  ubr 1000
!
!
interface ATM5/1/1.120 point-to-point
 ip address 10.3.3.3 255.255.255.0
 no ip directed-broadcast
 pvc 1/120
  vbr-nrt 50000 50000
  encapsulation aal5snap
!
!
interface ATM5/1/1.130 point-to-point
 ip address 10.4.4.4 255.255.255.0
 pvc 1/130
  vbr-rt 445 445
  encapsulation aal5snap
!
!
interface ATM5/1/1.140 point-to-point
 ip address 10.5.5.5 255.255.255.0
 atm arp-server nsap 47.00918100000000107B2B4B01.111155550000.00
 atm esi-address 111155550001.00
!
 svc SVC00 nsap 47.00918100000000107B2B4B01.222255550001.00
  protocol ip 10.5.5.6 broadcast
  oam-svc manage
  encapsulation aal5mux ip
 ubr 1000
!
```

Virtual Circuit Classes Configuration Example

```
vc-class atm high-class
  ilmi manage
  oam-pvc manage 5
  oam retry 10 7 3
!
vc-class atm low-class
!
interface ATM4/1/0
 no ip address
 class-int high-class
 atm ilmi-pvc-discovery subinterface
 pvc 0/5 qsaal
!
 pvc 0/16 ilmi
!
!
interface ATM4/1/0.1 multipoint
 pvc 1/110
  protocol 10.10.10.14
!
interface ATM4/1/1
 ip address 10.10.11.2 255.255.255.0
 class-int low-class
 atm uni-version 4.0
 atm pvp 1
 atm esi-address AAAAAAAAAAAA.AA
interface ATM4/1/1.2 multipoint
 pvc 2/100
  protocol ip 10.10.11.1
!
```

Virtual Circuit Bundles Configuration Example

```
!
interface ATM5/1/1
 ip address 1.1.1.1 255.255.255.0
 load-interval 30
 pvc 1/100
  protocol ip 1.1.1.3
  protocol ip 20.1.1.1
  cbr 140000
  broadcast
  oam-pvc manage
!
 pvc 1/101
  protocol ip 9.9.9.2
  encapsulation aal5ciscoppp Virtual-Template1
!
!
interface ATM5/1/1.200 multipoint
 ip address 7.7.7.1 255.255.255.0
 bundle atm-bundle
  pvc-bundle high 2/100
   class-vc high
  pvc-bundle med 2/101
   class-vc med
  pvc-bundle low 2/102
   class-vc low
!
```

Link Fragmentation and Interleaving with Virtual Templates Configuration Example

The following simple example shows a sample LFI configuration using a virtual template interface:

```
!
vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
class-map match-all prec4
  match ip precedence 4
class-map match-all prec5
  match ip precedence 5
class-map match-all prec6
  match ip precedence 6
class-map match-all prec7
  match ip precedence 7
class-map match-all prec0
  match ip precedence 0
class-map match-all prec1
  match ip precedence 1
class-map match-all prec2
  match ip precedence 2
class-map match-all dscp2
  match dscp 2
class-map match-all prec3
  match ip precedence 3
class-map match-all prec8
  match precedence 0  2  4  6
class-map match-any all
class-map match-all any
  match any
!
!
policy-map pmap1
  class prec1
    bandwidth percent 10
  class prec2
     police 100000000 3125000 3125000 conform-action transmit exceed-action drop
    priority
!
!
!
interface ATM2/1/0
 no ip address
 atm clock INTERNAL
!
interface ATM2/1/0.1 point-to-point
 pvc 0/100
  encapsulation aal5snap
  protocol ppp Virtual-Template1
!
!
interface ATM2/1/0.1000 point-to-point
 pvc 1/1000
  encapsulation aal5ciscoppp Virtual-Template2
!
!
interface ATM2/1/0.1001 point-to-point
 pvc 1/1001
  protocol ip 10.10.11.12
  encapsulation aal5ciscoppp Virtual-Template3
```

```
!
interface ATM2/1/1
 no ip address
 shutdown
!
interface ATM2/1/2
 no ip address
 shutdown
!
interface ATM2/1/3
 no ip address
!
interface Virtual-Template1
 bandwidth 100
 ip address 10.34.0.2 255.255.255.0
 no keepalive
 ppp chap hostname north-21
 ppp multilink
 ppp multilink fragment-delay 5
 ppp multilink interleave
 multilink max-fragments 16
 service-policy output pmap1
!
interface Virtual-Template2
 ip address 10.36.0.2 255.255.255.0
 no keepalive
 ppp chap hostname north-22
 ppp multilink
 ppp multilink fragment-delay 5
 ppp multilink interleave
 service-policy output pmap1
!
interface Virtual-Template3
 ppp chap hostname north-23
 ppp multilink
 ppp multilink fragment-delay 5
 ppp multilink interleave
 service-policy output pmap1
!
interface Vlan1
 no ip address
 shutdown
!
```

Distributed Compressed Real-Time Protocol Configuration Example

```
!
interface ATM5/1/0.200 point-to-point
 pvc 10/300
  encapsulation aal5mux ppp Virtual-Template200
!
...
!
interface Virtual-Template200
 bandwidth 2000
 ip address 10.1.200.2 255.255.255.0
 ip rcp header-compression passive
 ip tcp header-compression passive
 ppp chap hostname template200
 ppp multilink
 ppp multilink fragment-delay 8
 ppp multilink interleave
 ip rtp header-compression passive
 ip tcp compression-connections 64
!
```

Automatic Protection Switching Configuration Example

```
!
interface ATM4/0/0
 description working
 ip address 10.5.5.1 255.255.255.0
 no shutdown
 aps group 1
 aps working 1
 pvc 1/100
  protocol ip 10.5.5.2
!

interface ATM4/0/1
 description protect
 ip address 10.5.5.1 255.255.255.0
 aps group 1
 aps revert 2
 aps protect 0 10.7.7.7
 pvc 1/100
  protocol ip 10.5.5.2
!
interface Loopback1
 ip address 10.7.7.7 255.255.255.0
```

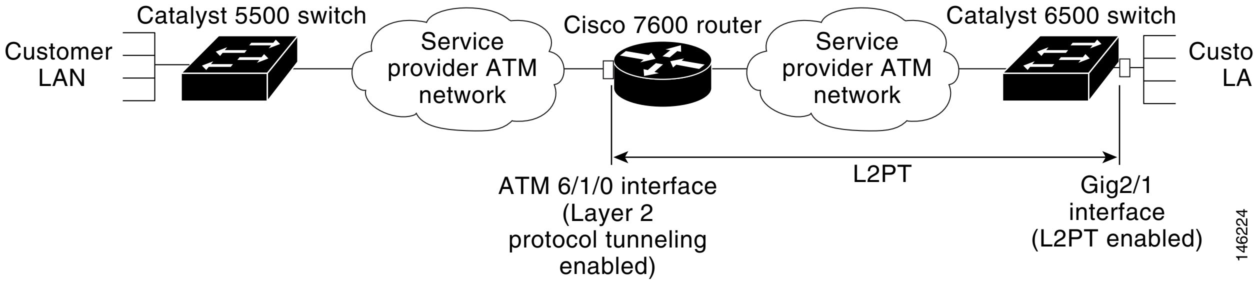
SONET and SDH Framing Configuration Example

```
!
interface ATM2/0/0
 description Example of SONET framing-“atm framing sonet” is default and doesn’t appear
 ip address 10.16.2.2 255.255.255.0
 logging event link-status
 atm sonet report all
 atm sonet threshold sd-ber 3
 atm sonet threshold sf-ber 6
 atm sonet overhead c2 0x00
!
interface ATM2/0/1
 description Example of SDH framing-”atm framing sdh” appears in configuration
 ip address 10.16.3.3 255.255.255.0
 logging event link-status
 atm framing sdh
 atm sonet report all
 atm sonet overhead c2 0x00
!
```

Layer 2 Protocol Tunneling Topology with a Cisco 7600, Catalyst 5500, and Catalyst 6500 Configuration Example

Figure 7-7 shows one sample network topology in which data packets are sent between a Catalyst 6500 switch and a Cisco 7600 series router.

Figure 7-7 Catalyst 5500 Switch, 6500 Switch, and Cisco 7600 Series Router in an L2PT Topology


Catalyst 5500 switch
Customer LAN
Service provider ATM network
Cisco 7600 router
Service provider ATM network
Catalyst 6500 switch
Customer LAN
L2PT
ATM 6/1/0 interface (Layer 2 protocol tunneling enabled)
Gig2/1 interface (L2PT enabled)
146224

As shown in Figure 7-7, Layer 2 Protocol Tunneling (L2PT) is configured at the Cisco 7600 ATM 6/1/0 interface and also at the Catalyst 6500 switch Gig 2/1 interface.

PVST packets are sent from the Catalyst 5500 switch to the Cisco 7600 series router. The Cisco 7600 router transports those BPDUs by way of L2PT and sends them to the Catalyst 6500 switch. Those BPDUs are decapsulated and restored before sending the packets out to the customer network.

The Cisco 7600 router and the Catalyst 6500 switch are provider edge (PE) devices and the rest are customer edge (CE) devices.

ATM Configuration Example

Any traffic coming in must be sent via a dot1q-tunnel. If the PE VLAN is 200 and the CE VLAN is 100, you have the following configuration:

```
Router(config)# interface atm 6/1/0
Router(config-if)# pvc 6/200
Router(config-if-atm-vc)# bridge-domain 200 dot1q-tunnel ignore-bpdu-pid pvst-tlv 100
```

Ethernet Configuration Example

An example of the Ethernet configuration follows:

```
Router(config)# interface gig2/1
Router(config-if)#switchport
Router(config-if)#switchport access vlan 200
Router(config-if)#switchport mode dot1q-tunnel
Router(config-if)# l2protocol-tunnel
```

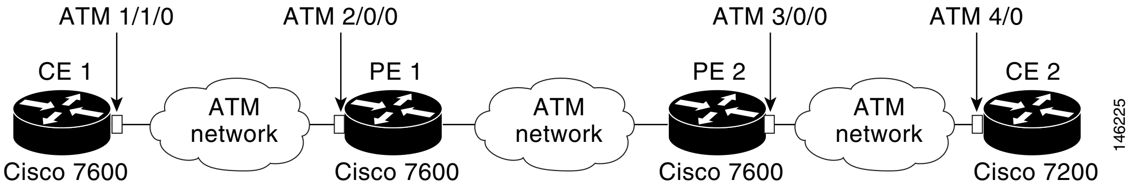
CE VLAN 100 is what is used at the customer sites. The Catalyst 5500 switch sends the IEEE BPDU in data format. The Cisco 7600 router receives the BPDU and first converts it to PVST+ format. Then the destination address (DA) MAC of the frame is changed to the protocol tunnel MAC address and sent out into the Layer 2 cloud.

At the other end, when the frame leaves the Gig 2/1 interface, the DA MAC is changed back to the PVST+ DA MAC and the PVST+ BPDU is sent to the CPE device.

Layer 2 Protocol Tunneling Topology with a Cisco 7600 and Cisco 7200 Configuration Example

In this example, a Cisco 7600 series router needs to communicate with a Cisco 7200 series router.

Figure 7-8 Cisco 7600 and Cisco 7200 Routers in an L2PT Topology



PE Configuration

On the PE routers, the configuration appears as follows:

```
!On PE 1
interface ATM2/0/0
    no ip address
    atm mtu-reject-call
    pvc 7/101
    bridge-domain 200 dot1q-tunnel
    !
end
!On PE 2
interface ATM3/0/0
    no ip address
    pvc 2/101
    bridge-domain 200 dot1q-tunnel pvst-tlv 100
    !
end
```

Cisco 7600 CE Configuration

The configuration for the Cisco 7600 CE 1 router would be as follows:

```
!On CE 1
interface ATM1/1/0
    no ip address
    atm mtu-reject-call
    pvc 7/101
    bridge-domain 101
    !
end
```

Cisco 7200 CE Configuration

The configuration for the Cisco 7200 CE 2 router would be as follows:

```
!On CE 2
interface ATM4/0
    no ip address
    no atm ilmi-keepalive
    pvc 2/101
    !
    bridge-group 101
end
```

Data Transmission Sequence from the Cisco 7200 CE to the Cisco 7600 CE

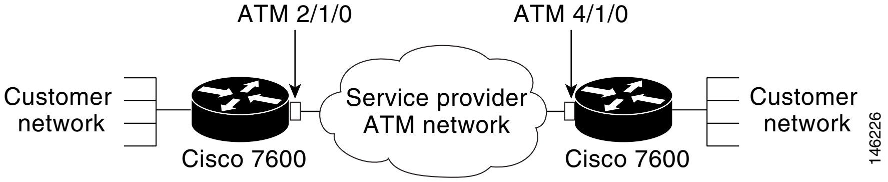
With the configurations and topologies shown in these examples, the data transmission sequence from the Cisco 7200 CE to the Cisco 7600 CE is as follows:

1. The Cisco 7200 CE 2 router sends BPDUs without the MAC header in RFC 1483 format.
2. The Cisco 7600 PE router receives the packets and then translates the IEEE BPDU into PVST+ BPDU format.
3. VLAN 100 is inserted into the PVST+ BPDU.
4. The frame's destination address (DA) MAC value is rewritten to use the protocol tunnel DA MAC and is sent out into the ATM network cloud.
5. The L2PT BPDU must go out of the PE 1 ATM 2/0/0 interface. The DA MAC is restored to the PVST+ DA MAC.
6. The PVST+ BPDU is sent to the Cisco 7600 CE 1 router.

Cisco 7600 Basic Back-to-Back Configuration Example

Figure 7-9 shows an example of a basic back-to-back configuration.

Figure 7-9 Cisco 7600 Routers in Basic Back-to-Back Topology



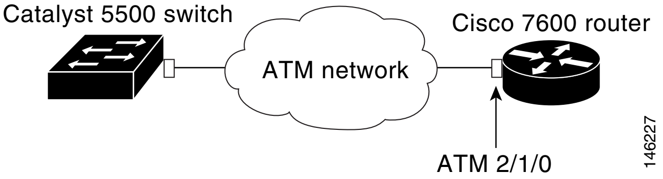
The PDUs exchanged are PVST+ BPDUs. The PVST+ BPDUs are sent using a PID of 0x00-07. The configuration is set as follows:

```
Router(config)# interface atm 2/1/0
Router(config-if)# pvc 2/202
Router(config-if-atm-vc)# bridge-domain 101
```

Catalyst 5500 Switch and Cisco 7600 Series Routers in Back-to-Back Topology Configuration Example

Another sample topology is a simple back-to-back setup, which tests basic Catalyst 5500 and Cisco 7600 interoperability.

Figure 7-10 Catalyst 5500 Switch and Cisco 7600 Routers in Back-to-Back Topology



When connected to a device that sends and receives IEEE BPDUs in data format (PID 0x00-07) such as the Catalyst 5000's ATM module, the configuration must be similar to the following:

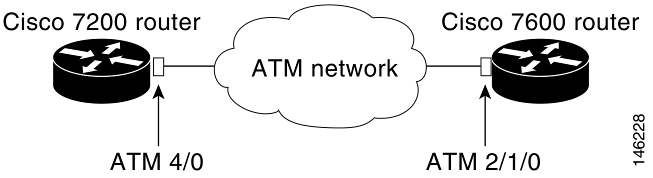
```
Router(config)# interface atm 2/1/0
Router(config-if)# pvc 2/202
Router(config-if-atm-vc)# bridge-domain 101 ignore-bpdu-pid pvst-tlv 101
```

The Cisco 7600 router translates its outgoing PVST+ BPDUs into IEEE BPDUs. Because the **ignore-bpdu-pid** keyword is also enabled, the BPDU uses a PID of 0x00-07, which is exactly what the Catalyst 5500 switch requires.

Cisco 7600 and Cisco 7200 in Back-to-Back Topology Configuration Example

When connecting to a device that is completely RFC 1483 compliant, in which the IEEE BPDUs are sent using a PID of 0x00-0E, you must use the new **ignore-bpdu-pid** keyword in the **bridge-domain** command.

Figure 7-11 Cisco 7600 Router Series and Cisco 7200 Router Series in Back-to-Back Topology



For example, when a Cisco 7600 series router is connected to a Cisco 7200 series router, the configuration would be as follows:

```
Router(config)# interface atm 2/1/0
Router(config-if)# pvc 2/202
Router(config-if-atm-vc)# bridge-domain 101 pvst-tlv 101
```

Note In this configuration scenario, the CE's VLAN number must be identical to the **bridge-domain** VLAN number.

An example of the Ethernet configuration follows:

```
Router(config)# interface gig2/1
Router(config-if)#switchport
Router(config-if)#switchport access vlan 200
Router(config-if)#switchport mode dot1q-tunnel
Router(config-if)# l2protocol-tunnel
```

CHAPTER 8

Troubleshooting the ATM SPAs

This chapter describes how to monitor and troubleshoot the asynchronous transfer mode (ATM) shared port adapters (SPAs) in a Catalyst 6500 Series switch. This document covers the 1-Port OC-48c/STM-16 ATM SPA, 1-Port OC-12c/STM-4 ATM SPA, and the 2-Port and 4-Port OC-3c/STM-1 ATM SPA.

- General Troubleshooting Information, page 8-1
- Monitoring the ATM SPA, page 8-2
- Troubleshooting the ATM Shared Port Adapter, page 8-15
- Preparing for Online Insertion and Removal of a SPA, page 8-26

For more information about troubleshooting your hardware installation, refer to the *Cisco 7600 Series Router SIP, SSC, and SPA Hardware Installation Guide*.

General Troubleshooting Information

This section provides the following general information for troubleshooting ATM SPA cards and their SPA interface processor (SIP) carrier cards:

- Interpreting Console Error and System Messages, page 8-1
- Using debug Commands, page 8-2
- Using show Commands, page 8-2

Interpreting Console Error and System Messages

To view the explanations and recommended actions for Catalyst 6500 Series switch error messages, including messages related to Catalyst 6500 Series switch SIPs and SPAs, refer to the *Catalyst 6500 Series Cisco IOS System Message Guide, 12.2SX*.

System error messages are organized in the documentation according to the particular system facility that produces the messages. The SIP and SPA error messages use the following facility names:

- Cisco 7600 SIP-200
- Cisco 7600 SIP-400
- 1-Port OC-12c/STM-4 ATM SPA
- 1-Port OC-48c/STM-16 ATM SPA
- 2-Port and 4-Port OC-3c/STM-1 ATM SPA

Using debug Commands

Along with the other debug commands supported on the Catalyst 6500 Series switch, you can obtain specific debug information for SPAs on the Catalyst 6500 Series switch using the **debug hw-module subslot** privileged exec command.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead can affect system use.

The **debug hw-module subslot** command is intended for use by Cisco Systems technical support personnel. For more information about the **debug hw-module subslot** command and other **debug** commands, see the *Cisco IOS Debug Command Reference, Release 12.2*. For more information about other commands that can be used on a Catalyst 6500 Series switch, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

Using show Commands

There are several **show** commands that you can use to monitor and troubleshoot the SIP and SPA cards on a Catalyst 6500 Series switch. For more information on these commands, see the "Monitoring the ATM SPA" section on page 8-2.

Also see Chapter 7, "Configuring the ATM SPAs" for additional information about these **show** commands.

Monitoring the ATM SPA

This section contains the following subsections that describe commands that can be used to display information about the ATM SPA hardware, interfaces, PVCs, SVCs, and APS configuration:

- Displaying Hardware Information, page 8-2
- Displaying Information About ATM Interfaces, page 8-5
- Displaying Information About PVCs and SVCs, page 8-7
- Displaying Information About Automatic Protection Switching, page 8-13

Note The outputs in this document are samples only. The actual output that appears on your switch depends on the model of switch, type of cards that are installed, and their configuration.

Displaying Hardware Information

Use the following commands to display different types of hardware and system information:

- **show version**—Displaying System Information, page 8-3

- **show hw-module subslot fpd** and **show idprom module**—Displaying Information About the ATM SPA Hardware Revision Levels, page 8-3
- **show controllers atm**—Displaying Information About the ATM Controller Hardware, page 8-4
- **show diagbus**—Displaying Information About ATM Ports, page 8-5

Displaying System Information

To display information about the switch, its system hardware and software, and the number of each type of interface that is installed, use the **show version** command. The following sample output shows a Cisco 7606 router that has two four-port OC-3c ATM SPA cards installed in a Cisco 7600 SIP-400 carrier card, along with a number of Gigabit Ethernet interfaces:

```
Router# show version

Cisco Internetwork Operating System Software
IOS (tm) c6sup2_rp Software (c6sup2_rp-JSV-M), Released Version 12.2(XX) [BLD-sipedon2
187]
Copyright (c) 1986-2004 by cisco Systems, Inc.
Compiled Tue 16-Mar-04 05:13 by jrstu
Image text-base: 0x40020F94, data-base: 0x424B0000

ROM: System Bootstrap, Version 12.2(14r)S1, RELEASE SOFTWARE (fc1)

sup2_7606 uptime is 44 minutes
Time since sup2_7606 switched to active is 43 minutes
System returned to ROM by power-on (SP by power-on)
System image file is "disk0:c6k222-jsv-mz_022204"

cisco CISCO7606 (R7000) processor (revision 1.0) with 458752K/65536K bytes of memory.
Processor board ID TBM06402027
SR71000 CPU at 600Mhz, Implementation 0x504, Rev 1.2, 512KB L2, 2048KB L3 Cache
Last reset from power-on
Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
1 FlexWAN controller (2 ATM).
2 SIP-400 controllers (7 ATM).
1 Dual-port OC12c ATM controller (2 ATM).
1 Virtual Ethernet/IEEE 802.3 interface(s)
8 Gigabit Ethernet/IEEE 802.3 interface(s)
11 ATM network interface(s)
1917K bytes of non-volatile configuration memory.
8192K bytes of packet buffer memory.
65536K bytes of Flash internal SIMM (Sector size 512K).
Configuration register is 0x2102
```

Displaying Information About the ATM SPA Hardware Revision Levels

To display information about the hardware revision of the SPA, as well as the version of the field-programmable device (FPD) that is onboard the SPA, use the **show hw-module subslot fpd** command. Cisco technical engineers might need this information to debug or troubleshoot problems with a SPA installation.

```
Router# show hw-module subslot fpd

==== ====================== ====== =============================================
                             H/W    Field Programmable   Current   Min. Required
Slot Card Type               Ver.   Device: "ID-Name"    Version      Version
```

```
==== ====================== ====== ================== =========== ==============
 5/0 4xOC-3 ATM SPA          1.0   1-I/O FPGA             0.70        0.70
---- ---------------------- ------ ------------------ ----------- --------------
 5/1 4xOC-3 ATM SPA          1.0   1-I/O FPGA             0.70        0.70
==== ====================== ====== =============================================
```

In addition, the **show idprom module** command also displays the serial number and board revisions for the ATM SPA.

```
Router# show idprom module 5/2

IDPROM for SPA module #5/2
        (FRU is '4-port OC3/STM1 ATM Shared Port Adapter')
        Product Identifier (PID) : SPA-4XOC3-ATM
        Version Identifier (VID) : V01
        PCB Serial Number        : PRTA0304088
        Top Assy. Part Number    : 68-2177-01
        73/68 Board Revision     : 04
        73/68 Board Revision     : 10
        Hardware Revision        : 0.17
        CLEI Code                : UNASSIGNED
```

Displaying Information About the ATM Controller Hardware

To display information about the controller hardware for an ATM interface, including framing and alarm configuration, as well as port, packet, and channel performance statistics, use the **show controllers atm** command, which has the following syntax:

show controllers atm *slot/sublot/port*

The following example shows typical output for an ATM SPA interface:

```
Router# show controllers atm 5/1/0

Interface ATM5/1/0 is up
 Framing mode: SONET OC3 STS-3c

SONET Subblock:
SECTION
  LOF = 0          LOS    = 0                            BIP(B1) = 603
LINE
  AIS = 0          RDI    = 2          FEBE = 2332       BIP(B2) = 1018
PATH
  AIS = 0          RDI    = 1          FEBE = 28         BIP(B3) = 228
  LOP = 0          NEWPTR = 0          PSE  = 1          NSE     = 2

Active Defects: None
Active Alarms:  None
Alarm reporting enabled for: SF SLOS SLOF B1-TCA B2-TCA PLOP B3-TCA

ATM framing errors:
  HCS (correctable):   0
  HCS (uncorrectable): 0

APS

  COAPS = 0          PSBF = 0
  State: PSBF_state = False
  Rx(K1/K2): 00/00  Tx(K1/K2): 00/00
  Rx Synchronization Status S1 = 00
  S1S0 = 00, C2 = 00
```

```
PATH TRACE BUFFER : STABLE

BER thresholds:  SF = 10e-3  SD = 10e-6
TCA thresholds:  B1 = 10e-6  B2 = 10e-6  B3 = 10e-6

  Clock source:  line
```

Note The ATM SPA does not support automatic updates of the remote host information, if any, in the Path Trace Buffer section of the **show controllers atm** command.

Displaying Information About ATM Ports

To display information about the type of port adapters that are installed in the switch, use the **show diagbus** command, which has the following syntax:

show diagbus *slot*

The *slot* argument is the slot number that contains the port adapter. The following example shows typical output for a 4-port OC-3c ATM SPA that is in slot 4 in the switch:

```
Router# show diagbus 4

Slot 4: Logical_index 8
        4-adapter SIP-200 controller
        Board is analyzed ipc ready
        HW rev 0.300, board revision 08
        Serial Number:  Part number: 73-8272-03

        Slot database information:
        Flags: 0x2004   Insertion time: 0x1961C (01:16:54 ago)

        Controller Memory Size:
                384 MBytes CPU Memory
                128 MBytes Packet Memory
                512 MBytes Total on Board SDRAM
        IOS (tm) cwlc Software (sip1-DW-M), Released Version 12.2(17)SX [BLD-sipedon2 107]

        SPA Information:
        subslot 4/0: 4xOC-3 ATM SPA (0x3E1), status: ok
        subslot 4/1: 4xOC-3 ATM SPA (0x3E1), status: ok
```

Displaying Information About ATM Interfaces

Use the following commands to display information about ATM interfaces:

- **show interface atm**—Displaying Layer 2 Information About an ATM Interface, page 8-6
- **show atm interface atm**—Displaying ATM-Specific Information About an ATM Interface, page 8-7
- **show ip interface**—Displaying Layer 3 IP Information About an ATM Interface, page 8-7

Displaying Layer 2 Information About an ATM Interface

To display Layer 2 information about an ATM interface or subinterface, along with the current status and packet counters, use the **show interface atm** command. The following example shows sample output for an ATM interface on an ATM SPA:

```
Router# show interface atm 5/1/0

ATM5/1/0 is up, line protocol is up
  Hardware is ATM SPA, address is 000a.f330.2a80 (bia 000a.f330.2a80)
  MTU 4470 bytes, sub MTU 4470, BW 149760 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM, loopback not set
  Encapsulation(s): AAL5
  4095 maximum active VCs, 21 current VCCs
  VC idle disconnect time: 300 seconds
  Signalling vc = 1, vpi = 0, vci = 5
         UNI Version = 4.0, Link Side = user
  6 carrier transitions
  Last input 01:47:05, output 00:00:01, output hang never
  Last clearing of "show interface" counters 01:03:35
  Input queue: 0/75/33439/80 (size/max/drops/flushes); Total output drops: 963306
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     9502306 packets input, 6654982829 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     45011 input errors, 131042 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     27827569 packets output, 21072150159 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

The following example shows sample output for a subinterface on this same ATM interface:

```
Router# show interface atm 5/1/0.200

ATM5/1/0.200 is up, line protocol is up
  Hardware is ATM SPA, address is 000a.f330.2a80 (bia 000a.f330.2a80)
  Internet address is 10.10.10.16/24
  MTU 4470 bytes, BW 149760 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  NSAP address: 47.0091810000000107B2B4B01.222255550001.00
  Encapsulation ATM
  12630 packets input, 10521156 bytes
  4994 packets output, 4176213 bytes
  3753 OAM cells input, 4366 OAM cells output
  AAL5 CRC errors : 0
  AAL5 SAR Timeouts : 0
  AAL5 Oversized SDUs : 0
```

Note The value for “packets output” in the default version of the **show interfaces atm** command includes the bytes used for ATM AAL5 padding, trailer and ATM cell header. To see the packet count without the padding, header, and trailer information, use the **show interfaces atm statistics** or **show atm pvc** commands.

Displaying ATM-Specific Information About an ATM Interface

To display Layer 2 ATM-specific information about an ATM interface or subinterface, use the **show atm interface atm** command:

```
Router# show atm interface atm 3/1/0

Interface ATM3/1/0:
AAL enabled:  AAL5  , Maximum VCs: 1023, Current VCCs: 1

Maximum Transmit Channels: 64
Max. Datagram Size: 4528
PLIM Type: SONET - 155000Kbps, TX clocking: LINE
Cell-payload scrambling: ON
sts-stream scrambling: ON
0 input, 0 output, 0 IN fast, 0 OUT fast, 0 out drop
 Avail bw = 155000
Config. is ACTIVE
```

Displaying Layer 3 IP Information About an ATM Interface

To display Layer 3 (IP-layer) information about an ATM interface, use the **show ip interface** command. To display a brief summary about all interfaces, use the **show ip interface brief** command.

To display information about a specific ATM interface, use the **show ip interface atm** *slot/subslot/port* command.

The following output shows a typical example for the **show ip interface brief** command:

```
Router# show ip interface brief

Interface                  IP-Address      OK? Method Status                Protocol
Vlan1                      unassigned      YES NVRAM  down                  down
GigabitEthernet1/1         172.18.76.57    YES NVRAM  up                    up
GigabitEthernet1/2         unassigned      YES NVRAM  administratively down down
ATM3/0/0                   unassigned      YES manual up                    up
ATM3/0/0.1                 unassigned      YES manual up                    up
ATM3/0/0.2                 10.1.1.1        YES manual up                    up
ATM3/1/0                   unassigned      YES manual up                    up
ATM3/1/0.1                 unassigned      YES manual up                    up
ATM3/1/0.2                 unassigned      YES unset  up                    up
ATM3/1/0.3                 11.1.1.1        YES manual up                    up
```

Displaying Information About PVCs and SVCs

Use the following commands to display information about PVCs and SVCs, including mapping, traffic, and VLAN configuration information:

- **show atm vp**—Displaying Information About Virtual Paths, page 8-8
- **show atm vc**—Displaying Information About Virtual Channels, page 8-8
- **show atm pvc**—Displaying Information About PVCs, page 8-9
- **show atm svc** and **show atm ilmi-status**—Displaying Information About SVCs, page 8-10
- **show atm map**—Displaying Information About Layer 2/Layer 3 Mappings, page 8-11
- **show atm traffic**—Displaying Information About ATM Traffic, page 8-12
- **show atm vlan**—Displaying Information About VLAN Mappings, page 8-12

- **show atm class-links**—Displaying Information About VC Bundles, page 8-13

Displaying Information About Virtual Paths

To display information about the virtual paths (VPs) that are configured on the switch's ATM interfaces, use the **show atm vp** command:

```
Router# show atm vp

                    Data  CES    Peak    CES
Interface    VPI    VCs   VCs    Kbps    Kbps     Status
ATM5/0/3     1      1     0      149760  0        ACTIVE
ATM5/0/3     1      2     0      299520  299000   ACTIVE
ATM5/0/3     2      0     0      1000    0        ACTIVE

Router#
```

To display detailed information about a specific virtual path, including its current PVCs and SVCs, specify the VPI with the **show atm vp** command:

```
Router# show atm vp 30

ATM8/1/0 VPI: 30,
ATM8/1/0 VPI: 30, PeakRate: 149760, CesRate: 0, DataVCs: 1, CesVCs: 0, Status: ACTIVE

  VCD    VCI   Type   InPkts   OutPkts   AAL/Encap     Status
  2      3     PVC    0        0         F4 OAM        ACTIVE
  3      4     PVC    0        0         F4 OAM        ACTIVE
  4      300   PVC    5        5         AAL5-SNAP     ACTIVE
  6      11    PVC    12       1         AAL5-SNAP     ACTIVE

TotalInPkts: 17, TotalOutPkts: 6, TotalInFast: 0, TotalOutFast: 6, TotalBroadcasts: 0
TotalInPktDrops: 0, TotalOutPktDrops: 0
```

Displaying Information About Virtual Channels

To display information about all of the virtual channels that are currently configured on the ATM interfaces, use the **show atm vc** command without any options:

```
Router# show atm vc

           VCD /                                        Peak  Avg/Min Burst
Interface  Name         VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells  Sts
3/0/0      1              1   100  PVC    SNAP     UBR  149760                UP
3/0/1      1              2   100  PVC    SNAP     UBR  149760                UP
3/0/2      1              3   100  PVC    SNAP     UBR  149760                UP
3/0/2      2              3   300  PVC    SNAP     UBR  149760                UP
3/0/3      1              4   100  PVC    SNAP     UBR  149760                UP
```

To display detailed information about a specific virtual connection, specify its VC descriptor (VCD) along with the command:

```
Router# show atm vc 20

   ATM1/1/0.200: VCD: 20, VPI: 2, VCI: 200
   UBR, PeakRate: 44209
   AAL5-LLC/SNAP, etype:0x0, Flags: 0xC20, VCmode: 0x0
   OAM frequency: 0 second(s)
   InARP frequency: 5 minutes(s)
   Transmit priority 4
   InPkts: 10, OutPkts: 11, InBytes: 680, OutBytes: 708
   InPRoc: 10, OutPRoc: 5, Broadcasts: 0
   InFast: 0, OutFast: 0, InAS: 0, OutAS: 6
   InPktDrops: 0, OutPktDrops: 0
   CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0
   OAM cells received: 0
   OAM cells sent: 0
   Status: UP
```

You can also display information about the VCs on a specific ATM interface and its subinterfaces:

```
Router# show atm vc interface atm 2/1/0

ATM2/0.101: VCD: 201, VPI: 20, VCI: 101
UBR, PeakRate: 149760
AAL5-LLC/SNAP, etype:0x0, Flags: 0xC20, VCmode: 0x0
OAM frequency: 0 second(s)
InARP frequency: 15 minutes(s)
Transmit priority 4
InPkts: 3153520, OutPkts: 277787, InBytes: 402748610, OutBytes: 191349235
InPRoc: 0, OutPRoc: 0, Broadcasts: 0
InFast: 211151, OutFast: 0, InAS: 0, OutAS: 0
InPktDrops: 0, OutPktDrops: 17
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0
OAM cells received: 0
OAM cells sent: 0
Status: UP
```

To display information about the traffic over a particular VC, use the **show atm vc** command with the following syntax:

show atm vc traffic interface atm *slot/subslot/port vpi vci*

```
Router# show atm vc traffic interface atm 1/0/1 1 101

Interface         VPI  VCI   Type      rx-cell-cnts     tx-cell-cnts
ATM1/0/1          1    101   PVC            9345              7231
```

Displaying Information About PVCs

Use the **show atm pvc** command to provide information about the PVCs that are currently configured on the switch. To display all PVCs that are currently configured on the switch's ATM interfaces and subinterfaces, use the **show atm pvc** command:

```
Router# show atm pvc

           VCD /                                        Peak  Avg/Min Burst
Interface  Name         VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells  Sts
2/1/0      1              2    32  PVC    SNAP     UBR     0                  UP
2/1/0.1    0              0    33  PVC    MUX      UBR 599040                 UP
2/1/0.2    2              0    34  PVC    MUX      UBR 599040               INAC
2/1/0.3    3              0    35  PVC    MUX      UBR 599040               INAC
2/1/0.4    4              0    36  PVC    MUX      UBR 599040               INAC
2/1/1.1    0              0    33  PVC    MUX      UBR 599040                 UP
2/1/1.2    2              0    34  PVC    MUX      UBR 599040               INAC
2/1/1.3    3              0    35  PVC    MUX      UBR 599040               INAC
2/1/1.4    4              0    36  PVC    MUX      UBR 599040               INAC
```

Tip To display all PVCs on a particular ATM interface or subinterface, use the **show atm pvc interface atm** command.

To display detailed information about a particular PVC, specify its VPI/VCI values:

```
Router# show atm pvc 1/100

ATM3/0/0: VCD: 1, VPI: 1, VCI: 100
UBR, PeakRate: 149760
AAL5-LLC/SNAP, etype:0x0, Flags: 0xC20, VCmode: 0x0
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC status: Not Managed
ILMI VC status: Not Managed
InARP frequency: 15 minutes(s)
Transmit priority 6
InPkts: 94964567, OutPkts: 95069747, InBytes: 833119350, OutBytes: 838799016
InPRoc: 1, OutPRoc: 1, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 94964566, OutAS: 95069746
InPktDrops: 0,  OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP

VC 1/100 doesn't exist on 7 of 8 ATM interface(s)
```

Displaying Information About SVCs

Use the **show atm vc** and **show atm ilmi-status** commands to provide information about the SVCs that are currently configured on the switch. To display all SVCs that are currently configured on the switch's ATM interfaces and subinterfaces, use the **show atm svc** command:

```
Router# show atm svc

           VCD /                                        Peak  Avg/Min Burst
Interface  Name         VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells  Sts
4/0/0      1              0     5  SVC    SAAL     UBR  155000                UP
4/0/2      4              0    35  SVC    SNAP     UBR  155000                UP
4/1/0      16             0    47  SVC    SNAP     UBR  155000                UP
4/1/0.1    593            0    80  SVC    SNAP     UBR  599040                UP
```

Tip To display all SVCs on a particular ATM interface or subinterface, use the **show atm svc interface atm** command.

To display detailed information about a particular SVC, specify its VPI/VCI values:

```
Router# show atm svc 0/35

ATM5/1/0.200: VCD: 3384, VPI: 0, VCI: 35, Connection Name: SVC00
UBR, PeakRate: 155000
AAL5-MUX, etype:0x800, Flags: 0x44, VCmode: 0x0
OAM frequency: 10 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Received
OAM VC status: Verified
ILMI VC status: Not Managed
```

```
VC is managed by OAM.
InARP DISABLED
Transmit priority 6
InPkts: 0, OutPkts: 4, InBytes: 0, OutBytes: 400
InPRoc: 0, OutPRoc: 4, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
InPktDrops: 0,  OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 10
F5 InEndloop: 10, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 10
F5 OutEndloop: 10, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP
TTL: 4
interface =  ATM5/1/0.200, call locally initiated, call reference = 8094273
vcnum = 3384, vpi = 0, vci = 35, state = Active(U10)
, point-to-point call
Retry count: Current = 0
timer currently inactive, timer value = 00:00:00
Remote Atm Nsap address: 47.00918100000000107B2B4B01.111155550001.00
, VC owner: ATM_OWNER_SMAP
```

To display information about the ILMI status and NSAP addresses being used for the SVCs on an ATM interface, use the **show atm ilmi-status** command:

```
Router# show atm ilmi-status atm 4/1/0

Interface : ATM4/1/0 Interface Type : Private UNI (User-side)
ILMI VCC : (0, 16) ILMI Keepalive : Enabled/Up (5 Sec 4 Retries)
ILMI State:       UpAndNormal
Peer IP Addr:     10.10.13.1       Peer IF Name:     ATM 3/0/3
Peer MaxVPIbits:  8               Peer MaxVCIbits:  14
Active Prefix(s) :
47.0091.8100.0000.0010.11b8.c601
End-System Registered Address(s) :
47.0091.8100.0000.0010.11b8.c601.2222.2222.2222.22(Confirmed)
47.0091.8100.0000.0010.11b8.c601.aaaa.aaaa.aaaa.aa(Confirmed)
```

Tip To display information about the SVC signaling PVC and ILMI PVC, use the **show atm pvc 0/5** and **show atm pvc 0/16** commands.

Displaying Information About Layer 2/Layer 3 Mappings

To display the mapping between the mappings between virtual circuits and Layer 3 IP addresses, use the **show atm map** command:

```
Router# show atm map

Map list ATM3/1/0.100_ATM_INARP : DYNAMIC
ip 10.11.11.2 maps to VC 19, VPI 2, VCI 100, ATM3/1/0.100
ip 10.11.11.1 maps to VC 4, VPI 0, VCI 60, ATM3/1/0.102
ip 10.11.13.4 maps to VC 1, VPI 5, VCI 33, ATM3/1/0
ip 10.10.9.20 maps to bundle vc-group1, 0/32, 0/33, 0/34, ATM3/1/0.1, broadcast

Map list ATM3/1/1.200_ATM_INARP : DYNAMIC
ip 10.2.3.2 maps to VC 20, VPI 2, VCI 200, ATM1/1/0.200
ip 10.2.3.10 maps to bundle vc-group2, 0/32, 0/33, 0/34, ATM3/1/1.1, broadcast
```

```
Map list ATM4/0/3.95_pvc1 : PERMANENT
ip 10.4.4.4 maps to NSAP CD.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.12, broadcast,
aal5mux, multipoint connection up, VC 6
ip 10.4.4.6 maps to NSAP DE.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.12, broadcast,
aal5mux, connection up, VC 15, multipoint connection up, VC 6
ip 10.4.4.16 maps to VC 1, VPI 13, VCI 95, ATM4/0/3.95, aal5mux
```

Displaying Information About ATM Traffic

To display general information about the traffic over the ATM interfaces, use the **show atm traffic** command:

```
Router# show atm traffic

276875 Input packets
272965 Output packets
2 Broadcast packets
0 Packets received on non-existent VC
6 Packets attempted to send on non-existent VC
272523 OAM cells received
F5 InEndloop: 272523, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
272963 OAM cells sent
F5 OutEndloop: 272963, F5 OutSegloop: 0, F5 OutRDI: 0
0 OAM cell drops
```

To display information about traffic shaping on the ATM interfaces in a particular slot, use the **show atm traffic shaping slot** command:

```
Router# show atm traffic shaping slot 3

Traffic Shaping CAM State : ACTIVE
Shaper Configuration Status :
    Shapers In Use By Config : 3, Shapers Available for Config : 3
Shaper Status in Hardware :
    Shaper 0 : In Use - Port : 0/0/0  Class : best-effort
    Shaper 1 : Not In Use
    Shaper 2 : Not In Use
    Shaper 3 : Not In Use
Statistics :
    Total cell discards : 0, clp0 discards : 0,  clp1 discards : 0
    Free cell buffers : 262143
    Total cells queued : 0
```

Tip You can also use the **show atm vc traffic** command to display traffic information for a particular VC.

Displaying Information About VLAN Mappings

To display the mappings of VLAN IDs to VCs, use the **show atm vlan** command:

```
Router# show atm vlan

        VCD     VLAN-ID
        101     1
        102     2
        103     3
        104     4
        105     5
        106     6
        107     7
        108     8
        109     9
        110     10
        111     11
        112     12
        113     13
        114     14
        115     15
        116     16
        117     17
        118     18
        119     19
        120     20
        121     21
        122     22
...
        800     11
        801     11
        802     11
        803     11
        804     326
        805     326
        806     326
        807     326
        808     327
        809     327
        810     327
        811     327
```

Tip To display the ports being used by a VLAN, use the **show vlan id** command.

Displaying Information About VC Bundles

To display the relationship between a particular VC and its parent VC class, including the parameters that were inherited from the class and those that were set manually, use the **show atm class-link** command:

```
Router# show atm class-links 0/66

Displaying vc-class inheritance for ATM2/0.3, vc 0/66:
broadcast - VC-class configured on main-interface
encapsulation aal5mux ip - VC-class configured on subinterface
no ilmi manage - Not configured - using default
oam-pvc manage 3 - VC-class configured on vc
oam retry 3 5 1 - Not configured - using default
ubr 10000 - Configured on vc directly
```

Displaying Information About Automatic Protection Switching

When you have configured automatic protection switching (APS) on one or more switch, you can show the current APS configuration and status with the **show aps** command, which has the following syntax:

show aps [**atm** *interface* | **controller** | **group** [*number*]]

You can display information about the overall APS configuration and about the specific APS groups that include interfaces that are present in the switch.

Displaying the Current APS Status

The **show aps** command, without any options, displays information for all interfaces in the switch that are configured as Working or Protect APS interfaces. The following shows sample output for a switch with one Working interface and one Protect interface:

```
Router# show aps

ATM4/0/1 APS Group 1: protect channel 0 (inactive)
        bidirectional, revertive (2 min)
        PGP timers (default): hello time=1; hold time=3
        state:
        authentication = (default)
        PGP versions (native/negotiated): 2/2
        SONET framing; SONET APS signalling by default
        Received K1K2: 0x00 0x05
                No Request (Null)
        Transmitted K1K2: 0x20 0x05
                Reverse Request (protect)
        Working channel 1 at 10.10.10.41 Enabled
        Remote APS configuration: (null)

ATM4/0/0 APS Group 1: working channel 1 (active)
        PGP timers (from protect): hello time=3; hold time=6
        state: Enabled
        authentication = (default)
        PGP versions (native/negotiated): 2/2
        SONET framing; SONET APS signalling by default
        Protect at 10.10.10.41
        Remote APS configuration: (null)
```

The following sample output is for the same interfaces, except that the Working interface has gone down and the Protect interface is now active:

```
Router# show aps

ATM4/0/1 APS Group 1: protect channel 0 (active)
        bidirectional, revertive (2 min)
        PGP timers (default): hello time=1; hold time=3
        state:
        authentication = (default)
        PGP versions (native/negotiated): 2/2
        SONET framing; SONET APS signalling by default
        Received K1K2: 0x00 0x05
                No Request (Null)
        Transmitted K1K2: 0xC1 0x05
                Signal Failure - Low Priority (working)
        Working channel 1 at 10.10.10.41 Disabled SF
        Pending local request(s):
                0xC (, channel(s) 1)
        Remote APS configuration: (null)

ATM4/0/0 APS Group 1: working channel 1 (Interface down)
        PGP timers (from protect): hello time=3; hold time=6
        state: Disabled
        authentication = (default)
        PGP versions (native/negotiated): 2/2
        SONET framing; SONET APS signalling by default
        Protect at 10.10.10.41
        Remote APS configuration: (null)
```

Tip To display the same information for a specific ATM interface, use the **show aps atm** *slot/subslot/port* command.

Displaying Information About APS Groups

To display information about the APS groups that are configured on the switch, use the **show aps group** command. You can display information for all groups or for a single group. For example, the following example shows a typical display for an individual group:

```
Router# show aps group 2

ATM4/0/0 APS Group 2: working channel 1 (active)
        PGP timers (from protect): hello time=3; hold time=6
        SONET framing; SONET APS signalling by default
        Protect at 10.10.10.7
        Remote APS configuration: (null)

ATM4/0/1 APS Group 2: protect channel 0 (inactive)
        bidirectional, revertive (2 min)
        PGP timers (default): hello time=1; hold time=3
        SONET framing; SONET APS signalling by default
        Received K1K2: 0x00 0x05
                No Request (Null)
        Transmitted K1K2: 0x20 0x05
                Reverse Request (protect)
        Working channel 1 at 10.10.10.7  Enabled
        Remote APS configuration: (null)
```

Note In the above example, both the Working and Protect interfaces in the APS group are on the same switch. If the two interfaces are on different switches, the **show aps group** command shows information only for the local interface that is a member of the APS group.

Troubleshooting the ATM Shared Port Adapter

This section describes the following commands and messages that can provide information in troubleshooting the ATM SPA and its interfaces:

- Understanding Line Coding Errors, page 8-16
- Using the Ping Command to Verify Network Connectivity, page 8-16
- Using the Ping Command to Verify Network Connectivity, page 8-16
- Using Loopback Commands, page 8-17
- Using ATM Debug Commands, page 8-25
- Using the Cisco IOS Event Tracer to Troubleshoot Problems, page 8-25

Tip For additional information on troubleshooting specific problems related to PVCs and SVCs, see the TAC tech note web page, at the following URL:

http://www.cisco.com/en/US/tech/tk39/tk48/tech_tech_notes_list.html

Understanding Line Coding Errors

This section provides a brief description of line coding and of the types of errors and alarms that can occur on a line:

- Alarm Indication Signal (AIS)—An AIS alarm indicates that an alarm was raised by a device on a line upstream to the ATM interface. Typically, the device creating the alarm is the adjacent network neighbor, but the AIS signal could also be generated by a device in the service provider's ATM cloud.
- Loss of Frame (LOF)—An LOF alarm occurs when the local interface is using a framing format that does not match the framing format being used on the line. LOF errors could also occur when the line or a device on the line is generating bit errors that are corrupting frames.
- Rx Cell HCS Error (HCSE)—The interface detected an error in the cell's header checksum (HCS) field, which indicates that one or more header bits were corrupted. (This field does not indicate whether any errors occurred in the cell's 48-bit payload.)
- Remote Alarm Indication (RAI) and Far-end Receive Failure (FERF)—An RAI/FERF error indicates that a problem exists between the local ATM interface and the far end, and that the error might not be in the local segment between the local interface and adjacent node.

Using the Ping Command to Verify Network Connectivity

The **ping** command is a convenient way to test the ability of an interface to send and receive packets over the network. The **ping** command sends ICMP echo request packets to a specified destination address, which should send an equal number of ICMP echo reply packets in reply. By measuring the numbering of packets that are successfully returned, as well as how long each packet takes to be returned, you can quickly obtain a rough idea of the Layer 3 to Layer 3 connectivity between two interfaces.

The IP **ping** command has the following syntax:

ping

or

ping *ip-address* [**repeat** *count*] [**data** *hex*] [**size** *datagram-size*]

If you enter just **ping**, the command interactively prompts you for all other parameters. Otherwise, you must specify at least a specific IP address as the destination for the ping. You can also optionally specify the following parameters:

- **repeat** *count*—Number of ICMP echo request packets to send. The default is five packets.
- **data** *hex*—The data pattern, in hexadecimal, to be sent in the ICMP echo request packets.
- **size** *datagram-size*—Specifies the size, in bytes, of the ICMP echo request packets to be sent. The range is 40 to 18024 bytes, with a default of 100 bytes.

Examples

The following shows a typical example of the **ping** command:

```
Router# ping 10.10.10.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echoes to 10.10.10.10, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/15/64 ms
```

Note You must have at least one PVC or SVC defined on an ATM interface before it can respond to an ICMP ping packet.

Using Loopback Commands

The **loopback** commands place an interface in loopback mode, which enables you to use the **ping** command to send packets through the local interface and line, so as to test connectivity. These commands are especially useful when an interface is experiencing a high number of cyclic redundancy check (CRC) errors, so that you can pinpoint where the errors are occurring.

Use the following procedures to perform the different loopback tests:

- Using loopback diagnostic to Create a Local Loopback, page 8-17
- Using loopback line, page 8-21

Tip For more information about using loopbacks to troubleshoot CRC errors on an interface, see the *CRC Troubleshooting Guide for ATM Interfaces* tech note, at the following URL:

http://www.cisco.com/en/US/tech/tk39/tk48/technologies_tech_note09186a00800c93ef.shtml

Using loopback diagnostic to Create a Local Loopback

To perform a local loopback test, in which the transmit data is looped back to the receive data at the physical (PHY) layer, use the **loopback diagnostic** command on an ATM interface. This loopback tests connectivity on the local ATM interface, verifying that the interface’s framing circuitry and segmentation and reassembly (SAR) circuitry is operating correctly. This loopback, however, does not test the interface’s optics circuitry and ports.

Tip If an ATM interface is currently connected to another ATM interface and passing traffic, shut down the remote ATM interface before giving the **loopback diagnostic** command on the local ATM interface. Otherwise, the remote interface continues to send traffic to the local interface, and the remote network could also start reporting interface and network errors.

Figure 8-1 shows a router-level diagram of a local loopback. In this example, the device with the ATM SPA is shown as a Cisco 7600 series router, but it can also be a Catalyst 6500 series switch. Figure 8-2 shows a block-level diagram of a local loopback, as it is performed within the ATM interface circuitry.

Figure 8-1 Performing a Local Loopback—Router Level

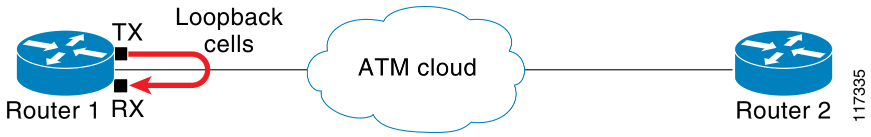
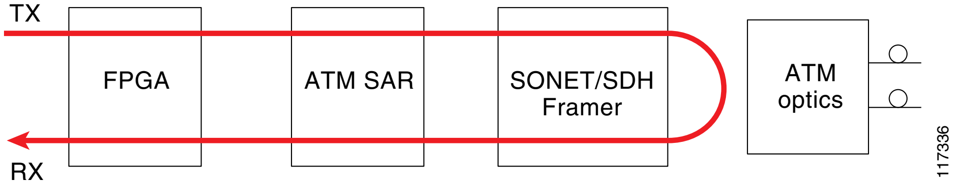


Figure 8-2 Performing a Local Loopback—Block Level



To configure a local loopback diagnostic test, perform the following task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface atm slot/subslot/port`	Enters interface configuration mode for the indicated port on the specified ATM SPA card.
Step 3	`Router(config-if)# loopback diagnostic`	Puts the ATM interface into the local loopback mode, so that data that is transmitted out the interface is internally routed back into the receive data line.
Step 4	`Router(config-if)# atm clock internal`	Specifies that the ATM interface should derive its clocking from its local oscillator, which is required, because the loopback command isolates the interface from the network and from the clocking signals that are derived from the network line.
Step 5	`Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	`Router# show interface atm slot/subslot/port`	(Optional) Verifies that the interface has been configured for loopback mode. The output should show the words "loopback set" when the interface is operating in loopback mode.
Step 7	`Router# debug atm packet interface atm slot/subslot/port`	(Optional) Enables packet debugging on the ATM interface. **Note** This command generates several lines of debug output for each packet transmitted and received on the interface. Do not use it on a live network, or you could force the processor to 100% utilization.

	Command	Purpose
Step 8	`Router(config-if)#` **`ping`** *`ip-address`* `[`**`repeat`** *`count`*`]` `[`**`data`** *`hex`*`]` `[`**`size`** *`datagram-size`*`]`	Sends an ICMP echo request packet to the specified IP address. • *ip-address*—Destination IP address for the ICMP echo request packet. Because the interface has been put into loopback mode, the exact IP address does not matter—any valid IP address can be specified. • **repeat** *count*—(Optional) Specifies the number of ICMP echo request packets to be sent. The default is 5. • **data** *hex*—(Optional) The data pattern, in hexadecimal, to be sent in the ICMP echo request packets. • **size** *datagram-size*—(Optional) Specifies the size, in bytes, of the ICMP echo request packets to be sent. The range is 40 to 18024 bytes, with a default of 100 bytes. **Note** Because the interface is in loopback mode, the ping command will report that it failed. This is to be expected.

Step 9	`Router#` **`show interface atm`** *`slot/subslot/port`*	Displays interface statistics, including whether any CRC or other errors occurred during the ping test. For example:
		`Router#` **`show interface atm 5/0/1`**
		`...`
		`Received 0 broadcasts, 0 runts, 0 giants, 0 throttles`
		`5 input errors, 5 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort`
		`...`
		`Router#`

Step 10	`Router(config)#` **`interface atm`** *`slot/subslot/port`*	Enters interface configuration mode for the indicated port on the specified ATM SPA card.
Step 11	`Router(config-if)#` **`no loopback diagnostic`**	Removes the local loopback and return the ATM interface to normal operations.

Note Also remember to restore the proper clocking on the local ATM interface and to reenable the remote ATM interface.

The following sample output shows a local loopback being set with the **loopback diagnostic** command. The **ping** command then sends two PING packets, and the resulting output from the **show interface** command shows that two CRC errors occurred.

```
Router# configure terminal
Router(config)# interface atm 4/1/0
Router(config-if)# loopback diagnostic
Router(config-if)# atm clock internal
Router(config-if)# end
Router# show interface atm 4/1/0

ATM4/1/0 is up, line protocol is up
  Hardware is ATM SPA, address is 000a.f330.2a80 (bia 000a.f330.2a80)
  MTU 4470 bytes, sub MTU 4470, BW 149760 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM, loopback set
  Encapsulation(s): AAL5
  4095 maximum active VCs, 21 current VCCs
  VC idle disconnect time: 300 seconds
  Signalling vc = 1, vpi = 0, vci = 5
         UNI Version = 4.0, Link Side = user
  6 carrier transitions
  Last input 01:47:05, output 00:00:01, output hang never
  Last clearing of "show interface" counters 01:03:35
  Input queue: 0/75/33439/80 (size/max/drops/flushes); Total output drops: 963306
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     9502306 packets input, 6654982829 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     27827569 packets output, 21072150159 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 output buffer failures, 0 output buffers swapped out

Router# debug atm packet interface atm 4/1/0

ATM packets debugging is on
Displaying packets on interface ATM4/1/0

Router# ping 10.10.10.10 count 2
Type escape sequence to abort.
Sending 2, 100-byte ICMP Echos to 10.10.10.10, timeout is 2 seconds:

1w1d: ATM4/1/0(O):
VCD:0x5 VPI:0x0 VCI:0x55 DM:0x100 SAP:AAAA CTL:03 OUI:000000 TYPE:0800 Length:0x70
1w1d: 4500 0064 001A 0000 FF01 B77A 0101 0102 0101 0101 0800 119A 13A2 07C5 0000
1w1d: 0000 2D41 2408 ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
1w1d: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
1w1d: ABCD ABCD ABCD ABCD ABCD
1w1d:
1w1d: ATM4/1/0(I):
VCD:0x5 VPI:0x0 VCI:0x55 Type:0x0 SAP:AAAA CTL:03 OUI:000000 TYPE:0800 Length:0x70
1w1d: 4500 0064 001A 0000 0101 B57B 0101 0102 0101 0101 0800 119A 13A2 07C5 0000
1w1d: 0000 2D41 2408 ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
1w1d: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
1w1d: ABCD ABCD ABCD ABCD ABCD
1w1d: .
1w1d: ATM4/1/0(O):
VCD:0x5 VPI:0x0 VCI:0x55 DM:0x100 SAP:AAAA CTL:03 OUI:000000 TYPE:0800 Length:0x70
1w1d: 4500 0064 001B 0000 FF01 B779 0101 0102 0101 0101 0800 09C9 13A3 07C5 0000
1w1d: 0000 2D41 2BD8 ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
1w1d: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
1w1d: ABCD ABCD ABCD ABCD ABCD
1w1d:
1w1d: ATM4/1/0(I):
VCD:0x5 VPI:0x0 VCI:0x55 Type:0x0 SAP:AAAA CTL:03 OUI:000000 TYPE:0800 Length:0x70
1w1d: 4500 0064 001B 0000 0101 B57A 0101 0102 0101 0101 0800 09C9 13A3 07C5 0000
1w1d: 0000 2D41 2BD8 ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
1w1d: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
1w1d: ABCD ABCD ABCD ABCD ABCD
1w1d: .
Success rate is 0 percent (0/2)

Router# configure terminal
Router(config)# interface atm 4/1/0
Router(config-if)# no loopback diagnostic
Router(config-if)# end
Router# show interface atm 4/1/0
```

```
ATM4/1/0 is up, line protocol is up
  Hardware is ATM SPA, address is 000a.f330.2a80 (bia 000a.f330.2a80)
  MTU 4470 bytes, sub MTU 4470, BW 149760 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM, loopback not set
  Encapsulation(s): AAL5
  4095 maximum active VCs, 21 current VCCs
  VC idle disconnect time: 300 seconds
  Signalling vc = 1, vpi = 0, vci = 5
         UNI Version = 4.0, Link Side = user
  6 carrier transitions
  Last input 01:47:05, output 00:00:01, output hang never
  Last clearing of "show interface" counters 01:03:35
  Input queue: 0/75/33439/80 (size/max/drops/flushes); Total output drops: 963306
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     9502306 packets input, 6654982829 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     2 input errors, 2 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     27827569 packets output, 21072150159 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Using loopback line

If an ATM interface can perform a local loopback successfully, without reporting errors, you can next try a line loopback (**loopback line** command) to determine if packet errors are being generated by the ATM network between the local and remote router. In a line loopback, the interface on the remote router is configured with the **loopback line** command, so that it reflects every packet that it receives back to the originating router. The local router then generates traffic with the **ping** command to determine whether the line through the network is generating the packet errors.

Figure 8-3 shows a router-level diagram of a line loopback. In this example, the device with the ATM SPA is shown as a Cisco 7600 series router, but it can also be a Catalyst 6500 series switch. Figure 8-4 shows a block-level diagram of a line loopback, as it is performed within the ATM interface circuitry.

Figure 8-3 Performing a Local Loopback—Router Level

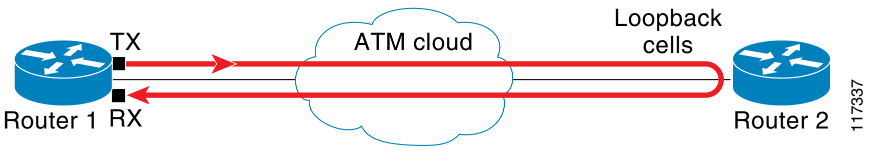
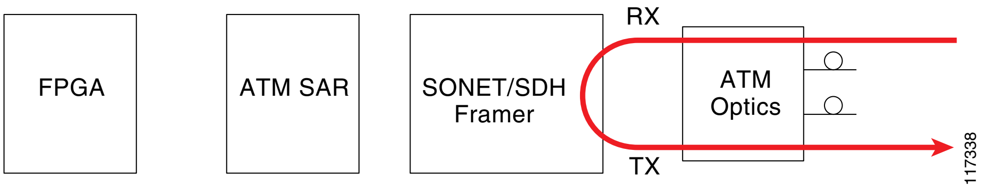


Figure 8-4 Performing a Line Loopback—Block Level



To configure a line loopback test, perform the following task.

	Command	Purpose
	Perform the following steps on the remote router:	
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface atm** *slot/subslot/port*	Enters interface configuration mode for the indicated port on the specified ATM SPA card.
Step 3	Router(config-if)# **loopback line**	Puts the ATM interface into the line loopback mode, so that it reflects any data it receives back to the originator.
Step 4	Router(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.
Step 5	Router# **show interface atm** *slot/subslot/port*	(Optional) Verifies that the interface has been configured for loopback mode. The output should show the words "loopback set" when the interface is operating in loopback mode.
	Perform the following steps on the local router:	
Step 1	Router# **debug atm packet interface atm** *slot/subslot/port*	(Optional) Enables packet debugging on the ATM interface. **Note** This command generates several lines of debug output for each packet transmitted and received on the interface. Do not use it on a live network, or you could force the processor to 100% utilization.
Step 2	Router(config-if)# **ping** *ip-address* [**repeat** *count*] [**data** *hex*] [**size** *datagram-size*]	Sends an ICMP echo request packet to the specified IP address. • *ip-address*—Destination IP address for the ICMP echo request packet. Because the interface has been put into loopback mode, the exact IP address does not matter—any valid IP address can be specified. • **repeat** *count*—(Optional) Specifies the number of ICMP echo request packets to be sent. The default is 5. • **data** *hex*—(Optional) The data pattern, in hexadecimal, to be sent in the ICMP echo request packets. The default is 0x0000. • **size** *datagram-size*—(Optional) Specifies the size, in bytes, of the ICMP echo request packets to be sent. The range is 40 to 18024 bytes, with a default of 100 bytes. **Note** Because the interface is in loopback mode, the ping command will report that it failed. This is to be expected.
Step 3	Router(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command	Purpose

Step 4	`Router#` **`show interface atm`** *`slot/subslot/port`*	Displays interface statistics, including whether any CRC or other errors during the ping test. For example:
		`Router#` **`show interface atm 5/0/1`**
		`...`
		`Received 0 broadcasts, 0 runts, 0 giants, 0 throttles`
		`5 input errors, 5 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort`
		`...`
		`Router#`

Note Also remember to remove the loopback mode on the remote ATM interface, using the **no loopback line** command.

The following example shows typical output when performing a line loopback. The following is the output on the remote router:

```
Router# configure terminal
Router(config)# interface atm 3/1/2
Router(config)# loopback line
Router(config)# end
Router# show interface atm 3/1/2

ATM3/1/2 is up, line protocol is up
  Hardware is ATM SPA, address is 000a.330e.2b08 (bia 000a.330e.2b08)
  MTU 4470 bytes, sub MTU 4470, BW 149760 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM, loopback set
  Encapsulation(s): AAL5
  4095 maximum active VCs, 103 current VCCs
  VC idle disconnect time: 300 seconds
  Signalling vc = 1, vpi = 0, vci = 5
         UNI Version = 4.0, Link Side = user
  6 carrier transitions
  Last input 00:00:02, output 00:00:01, output hang never
  Last clearing of "show interface" counters 01:03:35
  Input queue: 0/75/13/80 (size/max/drops/flushes); Total output drops: 37
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     932603 packets input, 6798282 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     387275 packets output, 371031501 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

On the Local Router

Perform the following on the local router:

```
Router# debug atm packet interface atm 4/0/0
ATM packets debugging is on
Displaying packets on interface ATM4/0/0

Router# ping 192.168.100.13 repeat 2 size 128

Type escape sequence to abort.
Sending 2, 128-byte ICMP Echos to 192.168.100.13, timeout is 2 seconds:
```

```
..
Success rate is 0 percent (0/2)

00:52:00: ATM4/0/0(O):
VCD:0x1 VPI:0x0 VCI:0x55 DM:0x100 SAP:AAAA CTL:03 OUI:000000 TYPE:0800 Length:0x70
00:52:00: 4500 0064 000F 0000 FF01 B785 0101 0102 0101 0101 0800 CE44 121D 0009 0000
00:52:00: 0000 002F 9DB0 ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:00: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:00: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:00: ABCD ABCD ABCD ABCD
00:52:00:
00:52:00: ATM4/0/0(I):
VCD:0x1 VPI:0x0 VCI:0x55 Type:0x0 SAP:AAAA CTL:03 OUI:000000 TYPE:0800 Length:0x70
00:52:00: 4500 0064 000F 0000 0101 B586 0101 0102 0101 0101 0800 CE44 121D 0009 0000
00:52:00: 0000 002F 9DB0 ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:00: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:00: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:00: ABCD ABCD ABCD ABCD
00:52:00:
00:52:02: ATM4/0/0(O):
VCD:0x1 VPI:0x0 VCI:0x55 DM:0x100 SAP:AAAA CTL:03 OUI:000000 TYPE:0800 Length:0x70
00:52:02: 4500 0064 0010 0000 FF01 B784 0101 0102 0101 0101 0800 C673 121E 0009 0000
00:52:02: 0000 002F A580 ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:02: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:00: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:00: ABCD ABCD ABCD ABCD
00:52:02:
00:52:02: ATM4/0/0(I):
VCD:0x1 VPI:0x0 VCI:0x55 Type:0x0 SAP:AAAA CTL:03 OUI:000000 TYPE:0800 Length:0x70
00:52:02: 4500 0064 0010 0000 0101 B585 0101 0102 0101 0101 0800 C673 121E 0009 0000
00:52:02: 0000 002F A580 ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:02: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:00: ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD ABCD
00:52:00: ABCD ABCD ABCD ABCD

Router# show interface atm 4/0/0

ATM4/0/0 is up, line protocol is up
  Hardware is ATM SPA, address is 000a.12f0.80b1 (bia 000a.12f0.80b1)
  MTU 4470 bytes, sub MTU 4470, BW 149760 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM, loopback not set
  Encapsulation(s): AAL5
  4095 maximum active VCs, 103 current VCCs
  VC idle disconnect time: 300 seconds
  Signalling vc = 1, vpi = 0, vci = 5
         UNI Version = 4.0, Link Side = user
  6 carrier transitions
  Last input 00:00:02, output 00:00:01, output hang never
  Last clearing of "show interface" counters 01:03:35
  Input queue: 0/75/13/80 (size/max/drops/flushes); Total output drops: 37
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
    5 minute input rate 0 bits/sec, 0 packets/sec
    5 minute output rate 0 bits/sec, 0 packets/sec
    94917 packets input, 1638383 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 2 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    102898 packets output, 2042785 bytes, 0 underruns
   0 output errors, 0 collisions, 5 interface resets
   0 ouput buffer failures, 0 output buffers swapped out
```

Using ATM Debug Commands

The following debug commands can be useful when troubleshooting problems on an ATM interface or subinterface:

- **debug atm bundle errors—**Displays information about VC bundle errors.
- **debug atm bundle events—**Displays information about events related to the configuration and operation of VC bundles, such as VC bumping, when bundles are brought up, when they are taken down, and so forth.
- **debug atm errors**—Displays errors that occur on an ATM interface, such as encapsulation and framing errors, as well as any errors that might occur during configuration of the ATM interfaces.
- **debug atm events**—Displays information about events that occur on the ATM interfaces, such as changes to the ATM SPA and ATM interface configuration, card and interface resets, and PVC or SVC creation.

Note The output of **debug atm events** can be extremely verbose and can cause problems if large numbers of ATM VCs are configured. The command should only be used when a few VCs are configured.

- **debug atm oam**—Displays the contents of ATM operation and maintenance (OAM) cells as they arrive from the ATM network.
- **debug atm packet**—Displays a hexadecimal dump of each packet's SNAP/NLPID/SMDS header, followed by the first 40 bytes of the packet.

Tip Use the **no debug all** command to turn off all debugging displays.

For more information about these commands, see the *Cisco IOS Debug Command Reference, Release 12.2*.

Using the Cisco IOS Event Tracer to Troubleshoot Problems

Note This feature is intended for use as a software diagnostic tool and should be configured only under the direction of a Cisco Technical Assistance Center (TAC) representative.

The Event Tracer feature provides a binary trace facility for troubleshooting Cisco IOS software. This feature gives Cisco service representatives additional insight into the operation of the Cisco IOS software and can be useful in helping to diagnose problems in the unlikely event of an operating system malfunction or, in the case of redundant systems, Route Processor switchover.

Event tracing reads informational messages from specific Cisco IOS software subsystem components that have been preprogrammed to work with event tracing, and logs messages from those components into system memory. Trace messages stored in memory can be displayed on the screen or saved to a file for later analysis.

The SPAs currently support the spa component to trace SPA OIR-related events.

For more information about using the Event Tracer feature, refer to the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1829/products_feature_guide09186a0080087164.html

Preparing for Online Insertion and Removal of a SPA

The Catalyst 6500 Series switch supports online insertion and removal (OIR) of the SIP, in addition to each of the SPAs. Therefore, you can remove a SIP with its SPAs still intact, or you can remove a SPA independently from the SIP, leaving the SIP installed in the switch.

This means that a SIP can remain installed in the switch with one SPA remaining active, while you remove another SPA from one of the SIP subslots. If you are not planning to immediately replace a SPA into the SIP, then be sure to install a blank filler plate in the subslot. The SIP should always be fully installed with either functional SPAs or blank filler plates.

For more information about activating and deactivating SPAs in preparation for OIR, see the “Preparing for Online Insertion and Removal of SIPs and SPAs” topic in the “Troubleshooting a SIP” chapter in this guide.

PART 4

Ethernet Shared Port Adapters

CHAPTER 9

Overview of the Fast Ethernet and Gigabit Ethernet SPAs

This chapter provides an overview of the release history, and feature and Management Information Base (MIB) support for the Fast Ethernet and Gigabit Ethernet SPAs on the Catalyst 6500 Series switch.

This chapter includes the following sections:

Release History

Release	Modification
12.2(33)SXI2	Support for the following SPA was introduced on the Cisco 7600 SIP-400 on the Catalyst 6500 series switch: • 1-Port 10 Gigabit Ethernet SPA, Version 2 (SPA-1X10GE-L-V2)
12.2(33)SXI	Support was restored for the Cisco 7600 SIP-600. Support for the following SPA was introduced on the Cisco 7600 SIP-400 on the Catalyst 6500 series switch: • 5-Port Gigabit Ethernet SPA (V2) The Any Transport over MPLS over GRE (AToMoGRE) feature was introduced on the Cisco 7600 SIP-400.

12.2(33)SXH	Support was removed for the Cisco 7600 SIP-600. Support for the following SPAs was introduced on the Cisco 7600 SIP-200 on the Catalyst 6500 series switch: • 4-Port Fast Ethernet SPA • 8-Port Fast Ethernet SPA The Multipoint Bridging (MPB) feature was introduced on the Cisco 7600 SIP-400 on the Cisco Catalyst 6500 series switch. The Scalable EoMPLS feature was increased from 4K to 12K on the Cisco 7600 SIP-400 on the Cisco Catalyst 6500 series switch.
12.2(18)SXF	Support for the following SPAs was introduced on the Cisco 7600 SIP-600 on the Cisco 7600 series router and Catalyst 6500 series switch: • 1-Port 10-Gigabit Ethernet SPA • 5-Port Gigabit Ethernet SPA • 10-Port Gigabit Ethernet SPA Support for the following SPA was introduced on the Cisco 7600 SIP-400 on the Cisco 7600 series router and Catalyst 6500 series switch: • 2-Port Gigabit Ethernet SPA

Supported Features

The following is a list of some of the significant hardware and software features supported by the Fast Ethernet and Gigabit Ethernet SPAs on the Catalyst 6500 Series switch:

- Autonegotiation
- Full-duplex operation
- 802.1Q VLAN termination
- Jumbo frames support (9188 bytes)
- Support for command-line interface (CLI)-controlled OIR
- 802.3x flow control
- Up to 4000 VLANs per SPA
- Up to 5000 MAC Accounting Entries per SPA (Source MAC Accounting on the ingress and Destination MAC Accounting on the egress)
- Per-port byte and packet counters for policy drops, oversubscription drops, CRC error drops, packet sizes, unicast, multicast, and broadcast packets
- Per-VLAN byte and packet counters for policy drops, oversubscription drops, unicast, multicast, and broadcast packets
- Per-port byte counters for good bytes and dropped bytes
- Any Transport over MPLS over GRE (AToMoGRE)
- Ethernet over Multiprotocol Label Switching (EoMPLS)
- Quality of service (QoS)
- Hot Standby Router Protocol (HSRP)

- Virtual Router Redundancy Protocol (VRRP)
- Hierarchal Virtual Private Lan Service (H-VPLS)
- Multipoint Bridging

Restrictions

Note For other SIP-specific features and restrictions see also Chapter 3, "Overview of the SIPs and SSC" in this guide.

The following restrictions apply to Cisco IOS Release 12.2(18)SXF:

- EtherChannel is not supported on Fast Ethernet SPAs or the 2-Port Gigabit Ethernet SPA on the Cisco 7600 SIP-400.

Supported MIBs

The following MIBs are supported by the Fast Ethernet and Gigabit Ethernet SPAs on the Catalyst 6500 Series switch:

- Entity-MIB (RFC 2737)
- Cisco-entity-asset-MIB
- Cisco-entity-field-replaceable unit (FRU)-control-MIB
- Cisco-entity-alarm-MIB
- Cisco-entity-sensor-MIB
- IF-MIB
- Etherlike-MIB (RFC 2665)
- Remote Monitoring (RMON)-MIB (RFC 1757)
- Cisco-class-based-QoS-MIB
- MPLS-related MIBs
- Ethernet MIB/RMON

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

SPA Architecture

This section provides an overview of the architecture of the Fast Ethernet and Gigabit Ethernet SPAs and describes the path of a packet in the ingress and egress directions. The SPA software references some of these architecture areas. Understanding the architecture is helpful when troubleshooting or interpreting the SPA CLI and **show** command output.

Every incoming and outgoing packet on the Fast Ethernet SPAs goes through the physical port (PHY RJ-45), the Media Access Controller (MAC), and a Layer 2 Filtering/Accounting ASIC. Every incoming and outgoing packet on the Gigabit Ethernet SPAs goes through the physical (PHY) SFP optics, Media Access Control (MAC), and ASIC devices.

Path of a Packet in the Ingress Direction

The following steps describe the path of an ingress packet through the Fast Ethernet or Gigabit Ethernet SPAs:

1. For Fast Ethernet SPAs, each of the ports receives incoming frames from one of the RJ-45 interface connectors. For Gigabit Ethernet SPAs, the SFP optics receive incoming frames on a per-port basis from one of the optical fiber interface connectors.
2. For Fast Ethernet SPAs, the PHY device processes the frame and sends it over a serial interface to the MAC device. For Gigabit Ethernet SPAs, the SFP PHY device processes the frame and sends it over a serial interface to the MAC device.
3. The MAC device receives the frame, strips the CRCs, and sends the packet via the SPI 4.2 bus to the ASIC.
4. The ASIC takes the packet from the MAC devices and classifies the Ethernet information. CAM lookups based on Ethernet type, port, VLAN, and source and destination address information determine whether the packet is dropped or forwarded to the SPA interface.

Path of a Packet in the Egress Direction

The following steps describe the path of an egress packet from the SIP through the Fast Ethernet and Gigabit Ethernet SPAs:

1. The packet is sent to the ASIC using the SPI 4.2 bus. The packets are received with Layer 2 and Layer 3 headers in addition to the packet data.
2. The ASIC uses port number, destination MAC address, destination address type, and VLAN ID to perform parallel CAM lookups. If the packet is forwarded, it is forwarded via the SPI 4.2 bus to the MAC device.
3. For Fast Ethernet SPAs, the MAC device forwards the packets to the PHY RJ-45 interface, which transmits the packet. For Gigabit Ethernet SPAs, the MAC device forwards the packets to the PHY laser-optic interface, which transmits the packet.

Displaying the SPA Hardware Type

To verify the SPA hardware type that is installed in your Catalyst 6500 Series switch, you can use the **show interfaces** command. For more information about these commands, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX.*

Table 9-1 shows the hardware description that appears in the **show** command output for each type of Gigabit Ethernet SPA that is supported on the Catalyst 6500 Series switch.

Table 9-1 SPA Hardware Descriptions in show Commands

SPA	Description in show interfaces command
4-Port Fast Ethernet SPA	Hardware is FastEthernet SPA
8-Port Fast Ethernet SPA	Hardware is FastEthernet SPA
1-Port 10-Gigabit Ethernet SPA	Hardware is TenGigEther SPA
2-Port Gigabit Ethernet SPA	Hardware is GigEther SPA
5-Port Gigabit Ethernet SPA	Hardware is GigEther SPA
10-Port Gigabit Ethernet SPA	Hardware is GigEther SPA

Example of the show interfaces Command

The following example shows output from the **show interfaces gigabitethernet** command on a Catalyst 6500 Series switch with a 2-Port Gigabit Ethernet SPA installed in slot 2:

```
Router# show interfaces gigabitethernet 2/0/1
GigabitEthernet2/0/1 is down, line protocol is down
  Hardware is GigEther SPA, address is 000a.f330.2e40 (bia 000a.f330.2e40)
  Internet address is 2.2.2.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Full-duplex, 1000Mb/s, link type is force-up, media type is SX
  output flow-control is on, input flow-control is on
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 03:19:34, output 03:19:29, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1703 packets input, 638959 bytes, 0 no buffer
     Received 23 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 1670 multicast, 0 pause input
     1715 packets output, 656528 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
.
.
.
```

The following example shows output from the **show interfaces tengigabitethernet** command on a Catalyst 6500 Series switch with a 1-Port 10-Gigabit Ethernet SPA installed in slot 7:

```
Router# show interfaces tengigabitethernet7/0/0
TenGigabitEthernet7/0/0 is up, line protocol is up (connected)
  Hardware is TenGigEther SPA, address is 0000.0c00.0102 (bia 000f.342f.c340)
  Internet address is 15.1.1.2/24
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  Full-duplex, 10Gb/s
  input flow-control is on, output flow-control is on
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:10, output hang never
  Last clearing of "show interface" counters 20:24:30
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  L2 Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes
  L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes mcast
  L3 out Switched: ucast: 0 pkt, 0 bytes mcast: 0 pkt, 0 bytes
     237450882 packets input, 15340005588 bytes, 0 no buffer
     Received 25 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     1676 packets output, 198290 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

CHAPTER 10

Configuring the Fast Ethernet and Gigabit Ethernet SPAs

This chapter provides information about configuring the 4-Port Fast Ethernet SPA, 8-Port Fast Ethernet SPA, 1-Port 10-Gigabit Ethernet SPA, 2-Port Gigabit Ethernet SPA, 5-Port Gigabit Ethernet SPA, and 10-Port Gigabit Ethernet SPA on the Catalyst 6500 Series switch. It includes the following sections:

- Configuration Tasks, page 10-1
- Verifying the Interface Configuration, page 10-40
- Configuration Examples, page 10-41

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

Configuration Tasks

This section describes how to configure the Fast Ethernet and Gigabit Ethernet SPAs and includes information about verifying the configuration.

This section includes the following topics:

- Required Configuration Tasks, page 10-2
- Specifying the Interface Address on a SPA, page 10-4
- Modifying the MAC Address on the Interface, page 10-5
- Obtaining MAC Address Accounting Statistics, page 10-5
- Configuring HSRP, page 10-6
- Modifying the Interface MTU Size, page 10-6
- Configuring the Encapsulation Type, page 10-8
- Configuring Autonegotiation on an Interface, page 10-8
- Configuring an Ethernet VLAN, page 10-10
- Configuring a Subinterface on a VLAN, page 10-10

- Configuring Layer 2 Switching Features, page 10-12
- Configuring Flow Control Support on the Link, page 10-18
- Configuring EtherChannels, page 10-21
- Configuring H-VPLS, page 10-21
- Configuring Ethernet Operations, Administration, and Maintenance, page 10-21
- QoS Configuration Guidelines for the Ethernet SPA, page 10-39
- Saving the Configuration, page 10-39
- Shutting Down and Restarting an Interface on a SPA, page 10-40
- Verifying Per-Port Interface Status, page 10-40

Required Configuration Tasks

This section lists the required configuration steps to configure the Fast Ethernet and Gigabit Ethernet SPAs. The commands in the section are applicable for both Fast Ethernet and Gigabit Ethernet SPAs; however, the examples below are for configuring a Gigabit Ethernet SPA. If you are configuring a Fast Ethernet SPA, replace the **gigabitethernet** command with the **fastethernet** command.

Some of the required configuration commands implement default values that might be appropriate for your network. If the default value is correct for your network, then you do not need to configure the command. These commands are indicated by "(As Required)" in the Purpose column.

Note Cisco Discovery Protocol (CDP) is disabled by default on the Cisco 7600 SIP-400 interfaces.

To configure the Fast Ethernet or Gigabit Ethernet SPAs, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface fastethernet** *slot/subslot/port*[.*subinterface-number*] or Router(config)# **interface gigabitethernet** *slot/subslot/port*[.*subinterface-number*] or Router(config)# **interface tengigabitethernet** *slot/subslot/port*[.*subinterface-number*]	Specifies the Fast Ethernet, Gigabit Ethernet, or the 10-Gigabit Ethernet interface to configure, where: • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 10-4. • *.subinterface-number*—(Optional) Specifies a secondary interface (subinterface) number.

	Command	Purpose
Step 3	`Router(config-if)#` **`ip address`** `[`*`ip-address mask`*` {`**`secondary`**`} | `**`dhcp`**` {`**`client-id`**` `*`interface-name`*`}{`**`hostname`**` `*`host-name`*`}]`	Sets a primary or secondary IP address for an interface that is using IPv4, where: • *ip-address*—Specifies the IP address for the interface. • *mask*—Specifies the mask for the associated IP subnet. • **secondary**—(Optional) Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address. • **dhcp**—Specifies that IP addresses will be assigned dynamically using DHCP. • **client-id** *interface-name*—Specifies the client identifier. The *interface-name* sets the client identifier to the hexadecimal MAC address of the named interface. • **hostname** *host-name*—Specifies the hostname for the DHCP purposes. The *host-name* is the name of the host to be placed in the DHCP option 12 field. **Note** The DHCP options with this command are not available for all Gigabit Ethernet SPAs and Fast Ethernet SPAs.
Step 4	`Router(config)#` **`ip accounting mac-address`** `{`**`input`**` | `**`output`**`}`	(Optional) Enables MAC address accounting. MAC address accounting provides accounting information for IP traffic based on the source and destination MAC addresses of the LAN interfaces, where: • **input**—Specifies MAC address accounting for traffic entering the interface. • **output**—Specifies MAC address accounting for traffic leaving the interface.
Step 5	`Router(config-if)#` **`mtu`** *`bytes`*	(As Required) Specifies the maximum packet size for an interface, where: • *bytes*—Specifies the maximum number of bytes for a packet. The default is 1500 bytes.

	Command	Purpose
Step 6	`Router(config-if)# standby [group-number] ip [ip-address [secondary]]`	(Required for HSRP Configuration Only) Creates (or enables) the HSRP group using its number and virtual IP address, where: • (Optional) *group-number*—The group number on the interface for which HSRP is being enabled. The range is 0 to 255; the default is 0. If there is only one HSRP group, you do not need to enter a group number. • (Optional on all but one interface if configuring HSRP) *ip-address*—The virtual IP address of the hot standby switch interface. You must enter the virtual IP address for at least one of the interfaces; it can be learned on the other interfaces. • (Optional) **secondary**—The IP address is a secondary hot standby switch interface. If neither switch is designated as a secondary or standby switch and no priorities are set, the primary IP addresses are compared and the higher IP address is the active switch, with the next highest as the standby switch. This command enables HSRP but does not configure it further. For additional information on configuring HSRP, see the "Configuring Hot Standby Router Protocol" section of the Cisco IP Configuration Guide, Release 12.2.
Step 7	`Router(config-if)# no shutdown`	Enables the interface.

Specifying the Interface Address on a SPA

SPA interface ports begin numbering with 0 from left to right. Single-port SPAs use only the port number 0. To configure or monitor SPA interfaces, you need to specify the physical location of the SIP, SPA, and interface in the CLI. The interface address format is *slot*/*subslot*/*port*, where:

- *slot*—Specifies the chassis slot number in the Catalyst 6500 Series switch where the SIP is installed.
- *subslot*—Specifies the secondary slot of the SIP where the SPA is installed.
- *port*—Specifies the number of the individual interface port on a SPA.

The following example shows how to specify the first interface (0) on a SPA installed in the first subslot of a SIP (0) installed in chassis slot 3:

```
Router(config)# interface serial 3/0/0
```

This command shows a serial SPA as a representative example; however, the same *slot*/*subslot*/*port* format is similarly used for other SPAs (such as ATM and POS) and other non-channelized SPAs.

Modifying the MAC Address on the Interface

The Gigabit Ethernet SPAs use a default MAC address for each port that is derived from the base address that is stored in the electrically erasable programmable read-only memory (EEPROM) on the backplane of the Catalyst 6500 Series switch.

To modify the default MAC address of an interface to some user-defined address, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# **mac-address** *ieee-address*`	Modifies the default MAC address of an interface to some user-defined address, where: • *ieee-address*—Specifies the 48-bit Institute of Electrical and Electronics Engineers (IEEE) MAC address written as a dotted triple of four-digit hexadecimal numbers (*xxxx.yyyy.zzzz*).

To return to the default MAC address on the interface, use the **no** form of the command.

Verifying the MAC Address

To verify the MAC address of an interface, use the **show interfaces gigabitethernet** privileged EXEC command and observe the value shown in the “address is” field.

The following example shows that the MAC address is 000a.f330.2e40 for interface 1 on the SPA installed in subslot 0 of the SIP installed in slot 2 of the Catalyst 6500 Series switch:

```
Router# show interfaces gigabitethernet 2/0/1
GigabitEthernet2/0/1 is down, line protocol is down
  Hardware is GigEther SPA, address is 000a.f330.2e40 (bia 000a.f330.2e40)
  Internet address is 2.2.2.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  Full-duplex, 1000Mb/s, link type is force-up, media type is SX
  output flow-control is on, input flow-control is on
(Additional output removed for readability)
```

Obtaining MAC Address Accounting Statistics

The **ip accounting mac-address** [**input** | **output**] command can be entered to enable MAC address accounting on an interface.

After MAC address accounting is enabled, MAC address statistics can be obtained by entering the **show interfaces mac-accounting** command.

Configuring HSRP

Hot Standby Router Protocol (HSRP) provides high network availability because it routes IP traffic from hosts without relying on the availability of any single switch. HSRP is used in a group of switches for selecting an active switch and a standby switch. (An *active switch* is the switch of choice for routing packets; a *standby switch* is a switch that takes over the switching duties when an active switch fails, or when preset conditions are met).

HSRP is enabled on an interface by entering the **standby** [*group-number*] **ip** [*ip-address* [**secondary**]] command. The **standby** command is also used to configure various HSRP elements. This document does not discuss more complex HSRP configurations. For additional information on configuring HSRP, see the "Configuring Hot Standby Router Protocol" section of the Cisco IP Configuration Guide, Release 12.2.

In the following HSRP configuration, standby group 2 on GigabitEthernet port 2/1/0 is configured at a priority of 110 and is also configured to have a preemptive delay should a switchover to this port occur.

```
Router(config)# interface GigabitEthernet 2/1/0
Router(config-if)# standby 2 ip 120.12.1.200
Router(config-if)# standby 2 priority 110
Router(config-if)# standby 2 preempt
```

Verifying HSRP

To display HSRP information, use the **show standby** command in EXEC mode:

```
Router# show standby
Ethernet0 - Group 0
Local state is Active, priority 100, may preempt
Hellotime 3 holdtime 10
Next hello sent in 0:00:00
Hot standby IP address is 198.92.72.29 configured
Active router is local
Standby router is 198.92.72.21 expires in 0:00:07
Standby virtual mac address is 0000.0c07.ac00
Tracking interface states for 2 interfaces, 2 up:
UpSerial0
UpSerial1
```

Modifying the Interface MTU Size

The Cisco IOS software supports three different types of configurable maximum transmission unit (MTU) options at different levels of the protocol stack:

- Interface MTU—Checked by the SPA on traffic coming in from the network. Different interface types support different interface MTU sizes and defaults. The interface MTU defines the maximum packet size allowable (in bytes) for an interface before drops occur. If the frame is smaller than the interface MTU size, but is not smaller than the minimum frame size for the interface type (such as 64 bytes for Ethernet), then the frame continues to process.
- IP MTU—Can be configured on a subinterface and is used by the Cisco IOS software to determine whether fragmentation of a packet takes place. If an IP packet exceeds the IP MTU size, then the packet is fragmented.
- Tag or Multiprotocol Label Switching (MPLS) MTU—Can be configured on a subinterface and allows up to six different labels, or tag headers, to be attached to a packet. The maximum number of labels is dependent on your Cisco IOS software release.

Different encapsulation methods and the number of MPLS MTU labels add additional overhead to a packet. For example, Subnetwork Access Protocol (SNAP) encapsulation adds an 8-byte header, dot1q encapsulation adds a 2-byte header, and each MPLS label adds a 4-byte header (*n* labels x 4 bytes).

For the Fast Ethernet and Gigabit Ethernet SPAs on the Catalyst 6500 Series switch, the default MTU size is 1500 bytes. When the interface is being used as a Layer 2 port, the The maximum configurable MTU is 9216 bytes. The SPA automatically adds an additional 38 bytes to the configured MTU size to accommodate some of the additional overhead.

Interface MTU Configuration Guidelines

When configuring the interface MTU size on a Fast Ethernet and Gigabit Ethernet SPA on a Catalyst 6500 Series switch, consider the following guidelines:

- The default interface MTU size accommodates a 1500-byte packet, plus 38 additional bytes to cover the following additional overhead:
  - Layer 2 header—14 bytes
  - SNAP header—8 bytes
  - Dot1q header—4 bytes
  - 2 MPLS labels—8 bytes
  - CRC—4 bytes

Note Depending on your Cisco IOS software release, a certain maximum number of MPLS labels are supported. If you need to support more than two MPLS labels, then you need to increase the default interface MTU size.

- If you are using MPLS, be sure that the **mpls mtu** command is configured for a value less than or equal to the interface MTU.
- If you are using MPLS labels, then you should increase the default interface MTU size to accommodate the number of MPLS labels. Each MPLS label adds 4 bytes of overhead to a packet.

Interface MTU Guidelines for Layer 2 Ports

On Layer 2 ports, it is important to understand the concept of the jumbo MTU. The jumbo MTU can be configured using the **system jumbomtu** command, although this command is only supported in the following situations:

- The port is a member of a Layer 2 EtherChannel.
- The new MTU size on the Layer 2 port is less than the currently configured maximum MTU for the port.

Note Fast Ethernet SPAs cannot function as Layer 2 ports.

Interface MTU Configuration Task

To modify the MTU size on an interface, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# mtu bytes`	Configures the maximum packet size for an interface, where: • *bytes*—Specifies the maximum number of bytes for a packet. The default is 1500 bytes and the maximum configurable MTU is 9216 bytes.

To return to the default MTU size, use the **no** form of the command.

Verifying the MTU Size

To verify the MTU size for an interface, use the **show interfaces gigabitethernet** privileged EXEC command and observe the value shown in the MTU field.

The following example shows an MTU size of 1500 bytes for interface port 1 (the second port) on the Gigabit Ethernet SPA installed in the top subslot (0) of the SIP that is located in slot 2 of the Catalyst 6500 Series switch:

```
Router# show interfaces gigabitethernet 2/0/1
GigabitEthernet2/0/1 is down, line protocol is down
  Hardware is GigEther SPA, address is 000a.f330.2e40 (bia 000a.f330.2e40)
  Internet address is 2.2.2.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
```

Configuring the Encapsulation Type

By default, the interfaces on the Fast Ethernet and Gigabit Ethernet SPAs support Advanced Research Projects Agency (ARPA) encapsulation. They do not support configuration of service access point (SAP) or SNAP encapsulation for transmission of frames; however, the interfaces will properly receive frames that use SAP and SNAP encapsulation.

The only other encapsulation supported by the SPA interfaces is IEEE 802.1Q encapsulation for virtual LANs (VLANs).

Configuring Autonegotiation on an Interface

Fast Ethernet and Gigabit Ethernet interfaces use a connection-setup algorithm called *autonegotiation.* Autonegotiation allows the local and remote devices to configure compatible settings for communication over the link. Using autonegotiation, each device advertises its transmission capabilities and then agrees upon the settings to be used for the link.

For the Fast Ethernet and Gigabit Ethernet interfaces on the Catalyst 6500 Series switch, flow control is autonegotiated when autonegotiation is enabled. Autonegotiation is enabled by default.

The following guidelines should be followed regarding autonegotiation:

- If autonegotiation is disabled on one end of a link, it must be disabled on the other end of the link. If one end of a link has autonegotiation disabled while the other end of the link does not, the link will not come up properly on both ends.
- Autonegotiation is not supported on the 10-Port Gigabit Ethernet SPA on the Cisco 7600 SIP-600.
- Flow control can be configured separately of autonegotiation when Ethernet SPAs are installed in a SIP-600.
- Flow control is enabled by default.
- Flow control will be on if autonegotiation is disabled on both ends of the link.
- Flow control cannot be disabled on a Fast Ethernet SPA.

Disabling Autonegotiation

Autonegotiation is automatically enabled and can be disabled on the Fast Ethernet interfaces on the Cisco 7600 SIP-200, and the Gigabit Ethernet interfaces on the Cisco 7600 SIP-400 or Cisco 7600 SIP-600. During autonegotiation, advertisement for flow control, speed, and duplex is advertised. If the Gigabit Ethernet interface is connected to a link that has autonegotiation disabled, autonegotiation should either be reenabled on the other end of the link or disabled on the Fast Ethernet or Gigabit Ethernet SPA if possible. Both ends of the link will not come up properly if only one end of the link has disabled autonegotiation.

Note Speed and duplex configurations are negotiated using autonegotiation. However, the only values that are negotiated are 100 Mbps for speed and full-duplex for duplex for Fast Ethernet SPAs, and 1000 Mbps for speed and full-duplex for duplex for Gigabit Ethernet SPAs. From a user's perspective, these settings are not negotiated, but are enabled using autonegotiation.

To disable autonegotiation on Fast Ethernet or Gigabit Ethernet SPAs, perform this task in interface configuration mode. Autonegotiation cannot be disabled on the 1-Port 10-Gigabit Ethernet SPA and 10-Port Gigabit Ethernet SPA when used in a SIP-400.

	Command	Purpose
Step 1	`Router(config-if)# no negotiation auto`	Disables autonegotiation on a Fast Ethernet SPA interface on the Cisco 7600 SIP-200 or a Gigabit Ethernet SPA interfaces on the Cisco 7600 SIP-400. No advertisement of flow control occurs.
Step 2	`Router(config-if)# speed nonegotiate`	Disables autonegotation of speed. This command first became available for SPAs when run in the SIP-600 and is not available in many setups.

Enabling Autonegotiation

Autonegotiation is automatically enabled and cannot be disabled (autonegotiation for the 10-Port Gigabit Ethernet SPA can be disabled when the SPA is installed in a SIP-600). During autonegotiation, advertisement and configuration of flow control, speed, and duplex occurs (flow control configuration is possible independently of autonegotiation when the Gigabit Ethernet SPA is installed in a SIP-600.

See the Configuring Flow Control for an Ethernet SPA Interface in a SIP-600, page 10-20). To reenable autonegotiation on a Fast Ethernet or Gigabit Ethernet interface, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)#` **`negotiation auto`**	Enables autonegotiation on a Fast Ethernet SPA interface on a Cisco 7600 SIP-200 or a Gigabit Ethernet SPA interfaces on the Cisco 7600 SIP-400. Advertisement of flow control occurs.
Step 2	`Router(config-if)#` **`no speed nonegotiate`**	Reenables autonegotation of speed. This command first became available for SPAs when run in the SIP-600 and is not available in many setups.

Configuring an Ethernet VLAN

For information on configuring Ethernet VLANs, see the “Creating or Modifying an Ethernet VLAN” section of the “Configuring VLANs” chapter in the *Cisco IOS Software Configuration Guide, Release 12.2(33)SXH and Later Releases*.

Configuring a Subinterface on a VLAN

You can configure subinterfaces on the Fast Ethernet SPA interfaces and Gigabit Ethernet SPA interfaces on a VLAN using IEEE 802.1Q encapsulation. Cisco Discovery Protocol (CDP) is disabled by default on the 2-Port Gigabit Ethernet SPA interfaces and subinterfaces on the Cisco 7600 SIP-400.

Note On any Cisco 7600 SIP-600 Ethernet port subinterface using VLANs, a unique VLAN ID must be assigned. This VLAN ID cannot be in use by any other interface on the Catalyst 6500 Series switch.

To configure a SPA subinterface on a VLAN, perform this task beginning in interface configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface fastethernet** *slot/subslot/port.subinterface-number* or Router(config)# **interface gigabitethernet** *slot/subslot/port.subinterface-number* or Router(config)# **interface tengigabitethernet** *slot/subslot/port.subinterface-number*	Specifies the Fast Ethernet, Gigabit Ethernet or 10-Gigabit Ethernet interface to configure, where: • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 10-4. • *.subinterface-number*—Specifies a secondary interface (subinterface) number.
Step 2	Router(config-subif)# **encapsulation dot1q** *vlan-id*	Defines the encapsulation format as IEEE 802.1Q ("dot1q"), where *vlan-id* is the number of the VLAN (1–4095).
Step 3	Router(config-if)# **ip address** *ip-address mask* [**secondary**]	Sets a primary or secondary IP address for an interface, where: • *ip-address*—Specifies the IP address for the interface. • *mask*—Specifies the mask for the associated IP subnet. • **secondary**—(Optional) Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.

Verifying Subinterface Configuration on a VLAN

To verify the configuration of a subinterface and its status on the VLAN, use the **show vlans** privileged EXEC command.

The following example shows the status of subinterface number 1 on port 0 on the SPA in VLAN number 200:

```
Router# show vlans
VLAN ID:200 (IEEE 802.1Q Encapsulation)

Protocols Configured:         Received:       Transmitted:
          IP                  0                  14

VLAN trunk interfaces for VLAN ID 200:

GigabitEthernet4/1/0.1 (200)

      IP:12.200.21.21

      Total 0 packets, 0 bytes input
      Total 2 packets, 120 bytes output
```

Configuring Layer 2 Switching Features

The Catalyst 6500 Series switch supports simultaneous, parallel connections between Layer 2 Ethernet segments. After you review the SPA-specific guidelines described in this document, then refer to the "Configuring Layer 2 Ethernet Interfaces" section of the *Cisco IOS Software Configuration Guide, Release 12.2(33)SXH and Later Releases* for more information about configuring the Layer 2 Switching features.

Configuring MPLSoGRE and mVPNoGRE

The MPLS over generic routing encapsulation (MPLSoGRE) and multicast virtual private network over generic routing encapsulation (mVPNoGRE) provides a mechanism to send unicast and multicast packets across a non-MPLS network. This is accomplished by creating a GRE tunnel across the non-MPLS network. When MPLS (unicast VRF) or mVPN (multicast VRF) packets are sent across the non-MPLS network, they are encapsulated within a GRE packet and transverse the non-MPLS network through the GRE tunnel. Upon receiving the GRE packet at the other side of the non-MPLS network, it removes the GRE header and forwards the inner MPLS or unicast VRF or mVPN packet to its final destination.

Note For mVPNoGRE, there is one outer packet and two inner packets. The outer packet is unicast GRE. The first inner packet is multicast GRE (mVPN). The second inner packet is normal (customer) multicast.

Note MPLSoGRE and mVPNoGRE are not supported on Fast Ethernet SPAs on the Cisco 7600 SIP-200.

MPLSoGRE Support

MPLSoGRE supports the following features:

- PE-to-PE tunneling of VRF unicast and multicast packets.
- IPv4 on CE-facing interfaces.
- IPv4 on core-facing interfaces.
- GRE 4-byte headers (no option fields).
- Non-dedicated physical interface supporting both tunneled and non-tunneled traffic.
- Only a single route for the tunnel between the Cisco 7600 SIP-400 physical interface or subinterface and the IP cloud may exist.
- No software imposed limit on the maximum number of tunnels. The SIP-400 supports a maximum number of 128 tunnels. Tunnel traffic can be routed through SIP-400 main interfaces or subinterfaces.

MPLSoGRE Restrictions

The following are not supported with MPLSoGRE:

- Ingress/egress features are not supported on the tunnel interface; they are supported on the physical interface or subinterface.
- GRE options: sequencing, checksum, key, source route.

- Some tunnel options: carry security options of client packet, Unidirectional Link Routing, Mobile IP path MTU discovery.
- The Cisco 7600 SIP-400 physical interface or subinterface used for the tunnel endpoint may not be used to transport native MPLS or its variations (for example, AToMoMPLS, EoMPLS, FRoMPLS, and PPPoMPLS).
- IPv6.
- Advanced features such as Carrier Supporting Carrier (CSC) and Inter-Autonomous Systems (Inter-AS).
- Multiple tunnels on the same Cisco 7600 SIP-400 interface or subinterfaces.

PE-to-PE Tunneling

MPLSoGRE and mVPNoGRE use the provider edge to provider edge (PE-to-PE) tunneling variation. This feature provides a scalable way to connect multiple customer networks across a non-MPLS network.by multiplexing traffic destined to multiple customer networks through a single GRE tunnel.

On each side of the non-MPLS network, each customer edge (CE) switch is assigned a VPN routing and forwarding (VRF) number by the PE switch. The IP networks behind the CE switches are learned by the PE switch through a routing protocol such as BGP, OSPF or RIP. Routes to these networks are then stored in the VRF routing table for that CE switch.

The PE switch on one side of the non-MPLS network is learned by the PE switch on the other side of the non-MPLS network though a routing protocol running within the non-MPLS network. Routes between the PE switches are stored in the main or default routing table.

Routes of the customer networks behind the PE switch are learned by the other PE switch through BGP and are not known to the non-MPLS network. This is accomplished by defining a static route to BGP neighbor (the other PE switch) through a GRE tunnel across the non-MPLS network. When routes are learned from the BGP neighbor, they will have the next-hop of the GRE tunnel and all customer network traffic will be sent using the GRE tunnel.

GRE Tunnel Attached to a Cisco 7600 SIP-400 Interface or Subinterface

For the Catalyst 6500 Series switch to perform the MPLS and mVPN processing and have the Cisco 7600 SIP-400 perform the GRE processing, a GRE tunnel must be attached to a Cisco 7600 SIP-400 MPLS and PIM (multicast) enabled interface or subinterface. The Catalyst 6500 Series switch views the Cisco 7600 SIP-400 main interface or subinterface as an MPLS or PIM interface so MPLS and mVPN processing is performed, and provides the Cisco 7600 SIP-400 with the correlation information needed to perform GRE processing.

Tunnel Interface Configuration

The **ip pim sparse-mode** command is not configured on the tunnel interface. It is automatically configured on the Cisco 7600 SIP-400 interface or subinterface when a tunnel is attached to the interface or subinterface.

The tunnel source IP address is the IP address of the Cisco 7600 SIP-400 interface or subinterface. The following example illustrates the tunnel interface configuration on the Catalyst 6500 Series switch:

```
Router(config)# interface Tunnel1
Router(config-if)# ip address 8.0.0.1 255.0.0.0
Router(config-if)# mpls label protocol ldp
Router(config-if)# tag-switching ip
Router(config-if)# tunnel source 6.0.0.1
Router(config-if)# tunnel destination 7.0.0.1
```

Configuring AToM over GRE

MPLS over generic routing encapsulation (MPLSoGRE) encapsulates MPLS packets inside IP tunnels, creating a virtual point-to-point link across non-MPLS networks. This allows users of primarily MPLS networks to continue to use existing non-MPLS legacy networks until migration to MPLS is possible. Any Transport over MPLS over GRE (AToMoGRE) includes support for the following transports:

- ATM over MPLS
- Frame Relay over MPLS (FRoMPLS)
- High-Level Data Link Control (HDLC) over MPLS
- Scalable Ethernet over MPLS (EoMPLS)
- Circuit Emulation over Packet (CEoP)
- Hardware-based EoMPLS

AToMoGRE is supported in Cisco IOS Release 12.2(33)SXI or later releases, and is supported only on the following hardware:

- Cisco 7600 SIP-400, 5-Port Gigabit Ethernet SPA, 2-Port Gigabit Ethernet SPA (core facing)
- ATM SPA (such as 2-Port OC-3c/STM-1 ATM SPA, 4-Port OC-3c/STM-1 ATM SPA, 1-Port OC-12c/STM-4 ATM SPA, 1-Port OC-48c/STM-16 ATM SPA), CEoP SPA (such as 24-Port Channelized T1/E1/J1 CEoP SPA) with inverse multiplexing (IMA) support, and all Ethernet interfaces
- Supervisor 32, Supervisor 720, or RSP720

AToMoGRE supports the following features:

- Provider edge (PE)-to-PE, provider (P)-to-PE, and P-to-P tunneling of MPLS packets (see Figure 10-1, Figure 10-2, and Figure 10-3).

Figure 10-1 PE-to-PE GRE Tunnel

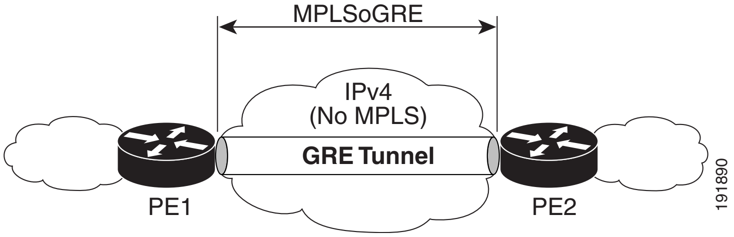


Figure 10-2 P-to-PE GRE Tunnel

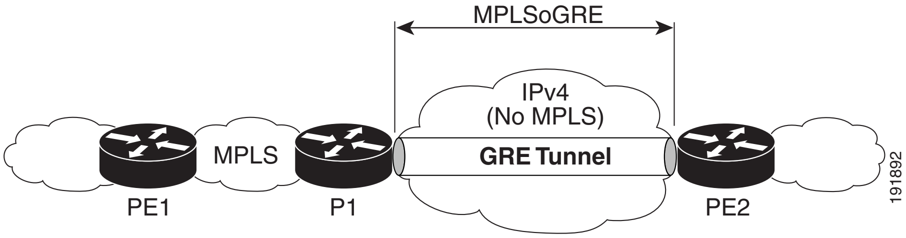
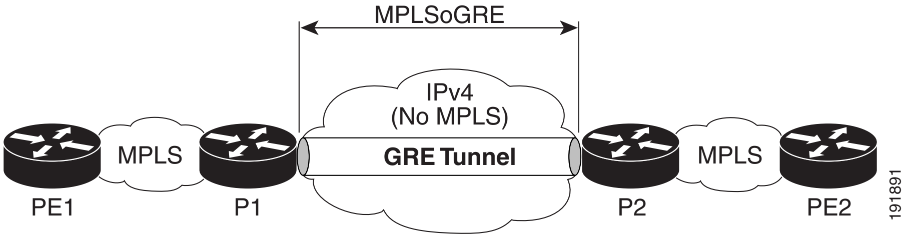


Figure 10-3 P-to-P GRE Tunnel



- IPv4 on customer edge (CE) facing interfaces.
- IPv4 on core facing interfaces.
- GRE 4-byte headers (no option fields).
- Nondedicated physical interface supporting both tunneled and nontunneled traffic.
- Multiple routes for the tunnel between the Cisco 7600 SIP-400 physical interface or subinterface and the IP cloud may exist. The routing protocol will pick only one route for MPLSoGRE traffic.
- No software-imposed limit on the maximum number of tunnels. The Cisco 7600 SIP-400 supports a maximum number of 128 tunnels. Tunnel traffic can be routed through Cisco 7600 SIP-400 main interfaces or subinterfaces.
- The Cisco 7600 SIP-400 physical interface or subinterface used for the tunnel endpoint can be used to carry native MPLS and AToMoMPLS and its variations: hardware-based EoMPLS, FRoMPLS, PPPoMPLS, HDLCoMPLS, Scalable EoMPLS, and CEoP.

AToMoGRE Configuration Guidelines

The following guidelines apply to AToMoGRE:

- Ingress and egress features are not supported on the tunnel interface; they are supported on the physical interface or subinterface.
- Unsupported GRE options are sequencing, checksum, key, and source route.
- Unsupported tunnel options are Carry Security Options of Client Packet, Unidirectional Link Routing, and Mobile IP Path MTU Discovery.
- The Cisco 7600 SIP-400 physical interface or subinterface used for the tunnel endpoint cannot be used to carry software-based EoMPLS and VPLS. Advanced features such as Carrier Supporting Carrier (CSC) and Inter-Autonomous Systems (Inter-AS) are not supported.
- AToM over GRE cannot be combined with the AToM Tunnel Select feature.

Configuring the Cisco 7600 SIP-400 Interface or Subinterface

Two configuration commands are required for configuring a Cisco 7600 SIP-400 interface or subinterface. The keywords of the commands may change based on input from the parser policy, but their placement and parameters remain the same. To configure the interface or subinterface, perform this task:

	Command	Purpose
Step 1	Router(config-subif)# **tunnel-interface** *tunnel-name*	Attaches a GRE tunnel to a Cisco 7600 SIP-400 subinterface. If the Cisco 7600 SIP-400 interface supports subinterfaces, the command will be available in subinterface configuration mode. If the Cisco 7600 SIP-400 interface does *not* support subinterfaces, the command is only available in interface configuration mode.
Step 2	Router(config-subif)# **ip route** *a.b.c.d e.f.g.h* [*i.j.k.l*]	Defines IP traffic that should be tunneled. This will normally be the IP address of the BGP neighbor. This command is only available in a submode of the **tunnel-interface** command. *a.b.c.d* is the IP address. *e.f.g.h* is the IP mask. *i.j.k.l* is the IP address of the next-hop switch.

The following example shows the commands to configure the MPLSoGRE and mVPNoGRE feature on a Cisco 7600 SIP-400 interface or subinterface. However, this example uses a Cisco 7600 SIP-400 interface that does *not* support subinterfaces:

```
Router# configure terminal
Router(config)# int pos2/0/0
Router(config-if)# tunnel-interface tu1
Router(config-if-ti)# ip route 4.0.0.1 255.255.255.255
Router(config-if-ti)# exit
Router(config-if)# end
Router#
```

When **tunnel-interface** is configured on the Cisco 7600 SIP-400 interface or subinterface, **ip pim sparse-mode** and **tag-switching ip** are automatically added to the interface. A static route to IP address contained on the **ip route** command is internally created. The following example shows the output of a **show running interface** after adding or configuring **tunnel-interface**. However, this example uses a Cisco 7600 SIP-400 interface that does *not* support subinterfaces:

```
Router# show run int pos2/0/0
!
interface POS2/0/0
 ip address 6.0.0.1 255.0.0.0
 ip pim sparse-mode
 no keepalive
 tunnel-interface Tunnel1
    ip route 4.0.0.1 255.255.255.255
    exit-tunnel-interface
 tag-switching ip
 clock source internal
end
```

Note You do not need to configure a static route (globally or on the tunnel) to the BGP neighbor on the Catalyst 6500 Series switch. This is automatically done by the **ip route** command under the **tunnel-interface** command on the Cisco 7600 SIP-400 interface or subinterface.

Displaying Unicast Routes

The display of unicast routes (Main Routing Table) shows the next hop for the BGP neighbor to be the Cisco 7600 SIP-400 interface or subinterface. On a switch that natively supports this feature, the next hop for the BGP neighbor is the tunnel interface.

The following example shows the output from the **show ip route** command:

```
Router# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     17.0.0.0/32 is subnetted, 1 subnets
O       17.0.0.2 [110/3] via 6.0.0.2, 00:09:55, POS2/0/0
     2.0.0.0/32 is subnetted, 1 subnets
C       2.0.0.1 is directly connected, Loopback0
     3.0.0.0/32 is subnetted, 1 subnets
O       3.0.0.1 [110/2] via 6.0.0.2, 00:09:55, POS2/0/0
S    64.0.0.0/8 [1/0] via 172.18.20.1
     4.0.0.0/32 is subnetted, 2 subnets
S       4.0.0.1 is directly connected, POS2/0/0
O       4.0.0.3 [110/3] via 6.0.0.2, 00:09:55, POS2/0/0
C    6.0.0.0/8 is directly connected, POS2/0/0
```

Displaying Multicast Routes

The display of multicast routes (groups) shows the output interface for the 239.0.0.0/8 group to be the Cisco 7600 SIP-400 interface or subinterface. On a switch that natively supports this feature, the output interface is the tunnel interface.

The following example shows the output from the **show ip mroute** command:

```
Router# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(2.0.0.1, 239.1.1.1), 00:02:02/00:03:02, flags: sTZ
  Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD
  Outgoing interface list:
    POS2/0/0, Forward/Sparse, 00:00:58/00:03:02, H

(4.0.0.1, 239.1.1.1), 00:00:58/00:02:02, flags: sTIZ
  Incoming interface: POS2/0/0, RPF nbr 8.0.0.2, RPF-MFD
  Outgoing interface list:
    MVRF vpn1, Forward/Sparse, 00:00:58/00:02:02, H
```

Displaying Tunnel-to-Interface Mappings

The **show cwan mplsogre** command displays the tunnel-to-interface mappings. The following example illustrates the output of the **show cwan mplsogre** command, which displays the tunnel-to-interface mappings:

```
Router# show cwan mplsogre
POS2/0/0
  Tunnel1 is attached
    Interface
      VLAN: 1022, STATE: UP
      IP Address: 6.0.0.1          IP Mask: 255.0.0.0
    Tunnel
      VLAN: 1017, STATE: UP
      IP Address: 8.0.0.1          IP Mask: 255.0.0.0
      Src Address: 6.0.0.1, Dst Address: 7.0.0.1
      Static Routes to Tunnel: 1
        IP Address: 4.0.0.1          IP Mask: 255.255.255.255
```

Scalable EoMPLS

As of the 12.2(33)SXH release, scalable EoMPLS now allows a Cisco 7600 SIP-400-based linecard to face the CE. This configuration allows the platform to scale the number of EoMPLS VCs that it can support from 4 K to 12 K. When AToM **xconnect** commands are placed on Cisco 7600 SIP-400 subinterfaces, the linecard performs AToM imposition and disposition. The supervisor engine performs only MPLS switching on traffic from these interfaces. Additionally, configuring **xconnect** commands on Cisco 7600 SIP-400 subinterfaces will not consume globally significant VLANs on a per xconnect basis. This change also provides the ability to support FRR on EoMPLS VCs with the same model as other CEF/MFI-based AToM configurations.

To achieve this scalability, Cisco 7600 SIP-400 must be the CE-facing linecard as opposed to the current model of a LAN linecard facing the CE. With Cisco 7600 SIP-400 configured for scalable EoMPLS, any linecard capable of switching MPLS packets may be core facing.

On a Supervisor Engine 720, configuring EoMPLS under a non-VLAN interface is considered hardware-based EoMPLS. Configuring EoMPLS on a VLAN interface is considered to be software-based MPLS. Configuring EoMPLS on Cisco 7600 SIP-400 subinterfaces is considered to be Scalable EoMPLS.

Configuring Flow Control Support on the Link

Flow control is turned on or off based on the result of the autonegotiation on the Cisco 7600 SIP-400. On the Cisco 7600 SIP-600, flow control can be configured independently of autonegotiation. For information on this process, see the "Configuring Autonegotiation on an Interface" section on page 10-8.

This section discusses the following topics:

- Verifying Flow Control Status for an Ethernet SPA Interface on a SIP-200, page 10-18
- Verifying Flow Control Status for an Ethernet SPA Interface on a SIP-400, page 10-19
- Configuring Flow Control for an Ethernet SPA Interface in a SIP-600, page 10-20

Verifying Flow Control Status for an Ethernet SPA Interface on a SIP-200

The following example shows how to verify that flow control pause frames are being transmitted and received for a Fast Ethernet SPA on the Cisco 7600 SIP-200:

```
Router# show hw sub 2 counter mac
    Show counters info for Subslot 2:
    port:0
    good_octets_received: 2046026640038
    bad_octets_received: 0
    good_frames_received: 31969140675
    bad_frames_received: 0
    broadcast_frames_received: 2
    multicast_frames_received: 3562
    good_octets_sent: 1373554315151
    good_frames_sent: 22892514199
    broadcast_frames_sent: 0
    multicast_frames_sent: 0
    mac_transfer_error: 0
    excessive_collision: 0
    unrecog_mac_control_received: 0
    fc_sent: 11232431
    good_fc_received: 0
    rx_over_flow_events: 234082101
    undersize: 0
    fragments: 0
    oversize: 0
    jabber: 0
    mac_rcv_error: 0
    bad_crc: 0
    collisions: 0
    late_collision: 0
    rate_limit_dropped: 0
    tx_fifo_full_packet_drops : 0
    spi4_rx_frames: 2814271686
    spi4_tx_frames: 1328805298
```

Verifying Flow Control Status for an Ethernet SPA Interface on a SIP-400

To verify flow control status on a Gigabit Ethernet interface on a SPA, use the **show interfaces gigabitethernet** privileged EXEC command and view the “output flow-control is” and “input flow-control is” output lines to see if input and output flow control is on or off. The “pause input” and “pause output” counters of the output of this command can be used to view the number of pause frames sent or received by the interface.

The following example shows that zero pause frames have been transmitted and received by the MAC device for interface port 1 (the second port) on the SPA located in subslot 0 of the SIP that is installed in slot 2 of the Catalyst 6500 Series switch:

```
Router# show interfaces gigabitethernet 2/0/1
GigabitEthernet2/0/1 is down, line protocol is down
  Hardware is GigEther SPA, address is 000a.f330.2e40 (bia 000a.f330.2e40)
  Internet address is 2.2.2.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  Full-duplex, 1000Mb/s, link type is force-up, media type is SX
  output flow-control is on, input flow-control is on
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 03:18:49, output 03:18:44, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

```
1703 packets input, 638959 bytes, 0 no buffer
Received 23 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 1670 multicast, 0 pause input
1715 packets output, 656528 bytes, 0 underruns
0 output errors, 0 collisions, 4 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 pause output
0 output buffer failures, 0 output buffers swapped out
```

Configuring Flow Control for an Ethernet SPA Interface in a SIP-600

On the Cisco 7600 SIP-600, flow control can be configured on Ethernet SPA interfaces by entering the **flowcontrol send** command to configure the interface to transmit pause frames or the **flowcontrol receive** command to configure the interface to receive pause frames.

To configure flow control on an Ethernet interface, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# flowcontrol send {desired | off | on}`	Enables transmission of outgoing pause frames. The following options can be configured with this command: • **desired**—Allows, but does not require, outgoing pause frames to leave the interface. • **off**—Disables transmission of outgoing pause frames. • **on**—Enables transmission of outgoing pause frames.
Step 2	`Router(config-if)# flowcontrol receive {desired | off | on}`	Enables the interface to receive incoming pause frames. The following options can be configured with this command: • **desired**—Allows, but does not require, the interface to receive incoming pause frames. • **off**—Does not allow incoming pause frames to enter the interface. • **on**—Allows incoming pause frames to enter the interface.

Note When a user configures flow control for either the transmit or receive direction, it is automatically enabled for both transmit and receive directions simultaneously.

Fast Ethernet SPAs have flow control enabled by default and it cannot be disabled.

Configuring EtherChannels

An EtherChannel bundles individual Ethernet links into a single logical link that provides the aggregate bandwidth of up to eight physical links.

Note EtherChannel is only supported on the 10-Port Gigabit Ethernet SPA and the 1-Port 10-Gigabit Ethernet SPA on the Cisco 7600 SIP-600. EtherChannel is not supported on the 2-Port Gigabit Ethernet SPA on the Cisco 7600 SIP-400 or on a Fast Ethernet SPA on the Cisco 7600 SIP-200.

For additional information on EtherChannels, see the "Configuring EtherChannels" section in the "Configuring Layer 3 and Layer 2 EtherChannel" chapter of the *Cisco IOS Software Configuration Guide, Release 12.2(33)SXH and Later Releases*.

Configuring H-VPLS

Hierarchal Virtual Private LAN Services (H-VPLS) use the provider core to join multiple attachment circuits together to simulate a virtual bridge that connects the multiple attachment circuits together. From a customer point of view, there is no topology for VPLS. All of the CE devices appear to connect to a logical bridge emulated by the provider core.

For more information on VPLS and H-VPLS feature, refer to the "Configuring Virtual Private LAN Service (VPLS)" section on page 4-23.

The H-VPLS feature works similarly on the Gigabit Ethernet SPAs as the OSM modules on the Cisco 7600 series router. For information about configuring VPLS and H-VPLS on the SIPs, refer to the "Virtual Private LAN Services on the Optical Services Modules" section of the *OSM Configuration Note* for the Cisco 7600 series router at the following URL:

http://www.cisco.com/en/US/docs/interfaces_modules/shared_port_adapters/configuration/7600series/SRC1/76cfgeth.html#wp11968087

Note H-VPLS is not available on Fast Ethernet SPAs.

H-VPLS Restrictions

In addition to the restrictions listed in the "Restrictions for VPLS" section in the *OSM Configuration Note* for the Cisco 7600 series router, the following restrictions apply to all transport types under H-VPLS:

- Split Horizon can be disabled, but should only be used for hub-and-spoke configurations.
- Hub-and-spoke and H-VPLS are supported.
- The Catalyst 6500 Series switch supports a maximum of 60 peer PEs and a maximum of 32,000 VCs.

Configuring Ethernet Operations, Administration, and Maintenance

In Cisco IOS Release 12.2(33)SXI and later releases, the Gigabit Ethernet SPAs support Operations, Administration, and Maintenance (OAM) as defined by IEEE 802.3ah, *Ethernet in the First Mile*. IEEE 802.3ah operates on a single point-to-point link between two devices using slow protocol packets called OAM protocol data units (OAMPDUs) that are never forwarded.

IEEE 802.3ah defines five functional areas, of which the Gigabit Ethernet SPAs on the Catalyst 6500 Series switch support the following three:

- OAM discovery—Supports identification of OAM support and capabilities on a peer device.
- Link monitoring—Provides event notification and link information. It also supports polling and response (but not writing) of the 802.3ah MIB.
- Remote failure indication—Supports informing a peer device that the receive path is down. This requires support of unidirectional operation on the link.

Ethernet OAM Configuration Guidelines

When configuring Ethernet OAM on the SPAs, consider the following guidelines:

- On Gigabit Ethernet links, the unidirectional fault signaling support in OAM and the autonegotiation capabilities of Gigabit Ethernet (IEEE 802.3z) are mutually exclusive. You must disable autonegotiation for OAM fault signaling to be sent over unidirectional links.
- Ethernet OAM requires point-to-point links where OAMPDUs are created and terminated.
- When configuring Ethernet OAM interface modes, consider the following guidelines:
  - At least one of the peer interfaces must be in active mode.
  - The peer interfaces either can be both in active mode, or one can be in active mode and the other in passive mode.
  - You can change Ethernet OAM modes without disabling OAM.
- When using templates to configure Ethernet OAM interfaces, consider the following guidelines:
  - If you use a template to configure common or global OAM characteristics and apply it to an interface, you can override any of the configuration statements in the template by configuring the same command at the interface with a different value.
  - You can define multiple templates to create different sets of link-monitoring characteristics.
  - You can only apply one template to any single Ethernet OAM interface.
- Table 10-1 provides information about where the OAM features for SPA interfaces are supported.

Table 10-1 Ethernet OAM Feature Compatibility by SIP and SPA Combination

Feature	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600
• OAM discovery • Link monitoring • Remote failure indication (Dying Gasp only)	Not supported.	In Cisco IOS Release 12.2(33)SXI: • 2-Port Gigabit Ethernet SPA	In Cisco IOS Release 12.2(33)SXI: • 1-Port 10-Gigabit Ethernet SPA • 5-Port Gigabit Ethernet SPA • 10-Port Gigabit Ethernet SPA
Remote loopback	Not supported.	Not supported.	Not supported.
MIB variable retrieval	Not supported.	Not supported.	Not supported.

Ethernet OAM Configuration Tasks

The following sections describe the Ethernet OAM configuration tasks:

- Enabling OAM on an Interface, page 10-23 (required)
- Enabling and Disabling a Link-Monitoring Session, page 10-25 (optional)
- Starting and Stopping Link-Monitoring Operation, page 10-25 (optional)
- Configuring Link-Monitoring Options, page 10-26 (optional)
- Configuring Remote Failure Indication Actions, page 10-33 (optional)
- Configuring Global Ethernet OAM Options Using a Template, page 10-34 (optional)
- Verifying Ethernet OAM Configuration, page 10-35

Enabling OAM on an Interface

OAM is disabled on an interface by default. When you enable OAM on an interface, the interface automatically advertises to the remote peer that it supports link-monitoring during OAM discovery. Link-monitoring support must be agreed upon by the peer interfaces for monitoring to operate across the link.

Once link-monitoring support is achieved between the peer interfaces, the interface will start the link-monitoring operation, send event OAMPDUs when errors occur locally, and interpret event OAM PDUs received by the remote peer.

You do not need to explicitly configure link-monitoring support, or start the link-monitoring operation on the link unless you have previously disabled monitoring support or operation on the interface.

To enable OAM features on a Gigabit Ethernet interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface** *type* *slot/subslot/port*	Specifies the Ethernet SPA interface, where: • *type*—Specifies the type of Ethernet interface, such as **gigabitethernet** or **tengigabitethernet**. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 10-4. **Note** Ethernet OAM can be defined on a main Gigabit Ethernet interface only, not on subinterfaces.
Step 2	Router(config-if)# **ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds* | **mode** {**active** | **passive**} | **timeout** *seconds*]	Enables OAM on a Gigabit Ethernet interface, where: • **max-rate** *oampdus*—(Optional) Specifies the maximum number of OAMPDUs that can be sent per second as an integer in the range of 1 to 10. The default is 10. • **min-rate** *num-seconds*—(Optional) Specifies the number of seconds (in the range 1–10) during which at least one OAMPDU must be sent. The default is 1 second. • **mode** {**active | passive**}—(Optional) Specifies the client mode for OAM discovery and link negotiation, where: – **active**— Specifies that the interface initiates OAMPDUs for protocol negotiation as soon as the interface becomes active. This is the default. At least one of the OAM peers must be configured in active mode. – **passive**—Specifies that the interface waits in a listening mode to receive an incoming OAMPDU for protocol negotiation from a peer. The passive interface begins sending OAMPDUs once it receives OAMPDUs from the peer.

Command	Purpose
	Note If you configure an interface in passive mode, then you must be sure that the peer is in active mode for successful OAM operation. • **timeout** *seconds*—Specifies the amount of time, in seconds (in the range 2–30), after which a device declares its OAM peer to be nonoperational and resets its state machine. The default is 5 seconds.

Enabling and Disabling a Link-Monitoring Session

The OAM peer interfaces must establish a link-monitoring session before the actual operation of link-monitoring can begin. If you have enabled OAM on the interface, and have not explicitly disabled link-monitoring support on the interface, then you do not need to explicitly configure link-monitoring support on the interface to establish a session.

The **ethernet oam link-monitor supported** command automatically runs in the background when you configure the **ethernet oam** interface configuration command. Be sure that at least one of the Ethernet OAM peers is configured for active mode so that a session can be established.

To explicitly configure and enable a link-monitoring session on an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ethernet oam link-monitor supported`	Enables link-monitoring support on an Ethernet OAM interface.

To disable a link-monitoring session on an interface, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# no ethernet oam link-monitor supported`	Disables link-monitoring support on an Ethernet OAM interface.

Starting and Stopping Link-Monitoring Operation

If a link-monitoring session is established among the Ethernet OAM peer interfaces, then sending and receiving of Event Notification OAMPDUs can begin between the peers. This link-monitoring operation across the link automatically starts when you enable OAM on the interface.

The **ethernet oam link-monitor on** command automatically runs in the background when you configure the **ethernet oam** interface configuration command.

You can stop and restart the operation of link-monitoring (or the sending and receiving of Event Notification OAMPDUs on a link). Stopping a link-monitoring operation is not the same thing as disabling link-monitoring support. When you stop a link-monitoring operation, the interface is still configured to support link-monitoring with its peer, but just is not actively sending and receiving Event Notification OAMPDUs.

To explicitly configure and start a link-monitoring operation on an interface, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)#` **`ethernet oam link-monitor on`**	Starts link-monitoring on an Ethernet OAM interface.

To stop a link-monitoring operation on an interface, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)#` **`no ethernet oam link-monitor on`**	Stops link-monitoring on an Ethernet OAM interface.

Configuring Link-Monitoring Options

When OAM link-monitoring is active, Event Notification OAMPDUs are sent to a remote OAM client when errors are detected locally. You can configure certain windows and thresholds to define when these error event notifications are triggered. If you do not modify the link-monitoring options, default values are used for the window periods and low thresholds.

The Gigabit Ethernet SPAs support the following types of error events as defined by IEEE 802.3ah:

- Errored Symbol Period (errored symbols per second)—This event occurs when the number of symbol errors during a specified period exceeds a threshold. These are coding symbol errors (for example, a violation of 4B/5B coding).
- Errored Frame (errored frames per second)—This event occurs when the number of frame errors during a specified period exceeds a threshold.
- Errored Frame Period (errored frames per N frames)—This event occurs when the number of frame errors within the last N frames exceeds a threshold.
- Errored Frame Seconds Summary (errored seconds per M seconds)—This event occurs when the number of errored seconds (one second intervals with at least one frame error) within the last M seconds exceeds a threshold.

Cisco Systems adds the following types of vendor-specific error events:

- Receive CRC (errored frames per second)—This event occurs when the number of frames received with CRC errors during a specified period exceeds a threshold.
- Transmit CRC (errored frames per second)—This event occurs when the number of frames transmitted with CRC errors during a specified period exceeds a threshold.

The link-monitoring options can be configured in a global template that can be applied to one or more interfaces, and also can be explicitly configured at the interface.

Specifying Errored Symbol Period Link-Monitoring Options

The errored symbol period link-monitoring options include the ability to specify the number of symbols to be tracked or counted for errors, and the high and low thresholds for triggering the Errored Symbol Period Link Event.

To specify errored symbol period link-monitoring options, perform this task in interface configuration or template configuration mode:

Command	Purpose
`Router(config-if)#` **`ethernet oam link-monitor symbol-period window`** *`million-symbol-units`*	(Optional) Specifies the number of symbols (in the range 1–65535, as a multiple of 1 million symbols) to be included in the error counting according to the specified thresholds. The default window unit is 100, or 100 million symbols.
`Router(config-if)#` **`ethernet oam link-monitor symbol-period threshold low`** *`low-symbols`*	(Optional) Specifies the low errored symbol threshold as a number of symbol errors (in the range 0–65535). If the number of error symbols in the window period is equal to or greater than *low-symbols*, then the Errored Symbol Period Link Event will be generated. The default low threshold is 0 symbols.
`Router(config-if)#` **`ethernet oam link-monitor symbol-period threshold high`** `{`**`none`** `|` *`high-symbols`*`}`	(Optional) Specifies the high errored symbol threshold as a number of error symbols (in the range 1–65535). If the number of error symbols in the window period is equal to or greater than *high-symbols*, then a user defined action will be triggered. There is no default for the high threshold, so you must explicitly configure a value to enable it. For more information about configuring a user-defined action, see "Specifying a High-Threshold Action" section on page 10-32.

Specifying Errored Frame Link-Monitoring Options

The errored frame link-monitoring options include the ability to specify a period of time during which frame errors are tracked or counted, and the high and low thresholds for triggering the Errored Frame Link Event. The Gigabit Ethernet SPAs on the Catalyst 6500 Series switch count general frame errors, such as CRC errors and corrupted packets, as errored frames.

To specify errored frame link-monitoring options, perform this task in interface configuration or template configuration mode:

Command	Purpose
`Router(config-if)#` **`ethernet oam link-monitor frame window`** *`100-millisecond-units`*	(Optional) Specifies a period of time (in the range 10–600, as a multiple of 100 milliseconds) during which error-counting occurs according to the specified thresholds. The default window unit is 10, or 1000 milliseconds.

Command	Purpose
`Router(config-if)# **ethernet oam link-monitor frame threshold low** *low-frames*`	(Optional) Specifies the low error frame threshold as a number of frames (in the range 0–65535). If the number of error frames in the window period is equal to or greater than *low-frames*, then the Errored Frame Link Event will be generated. The default low threshold is 0 frame errors.
`Router(config-if)# **ethernet oam link-monitor frame threshold high** {**none** | *high-frames*}`	(Optional) Specifies the high error frame threshold as a number of error frames (in the range 1–65535). If the number of error frames in the window period is equal to or greater than *high-frames*, then a user-defined action will be triggered. There is no default for the high threshold, so you must explicitly configure a value to enable it. Use the **none** keyword to disable the high threshold. For more information about configuring a user-defined action, see "Specifying a High-Threshold Action" section on page 10-32.

Specifying Errored Frame Period Link-Monitoring Options

The errored frame period link-monitoring options include the ability to specify the number of error frames to be tracked or counted for errors, and the high and low thresholds for triggering the Errored Frame Period Link Event. The Gigabit Ethernet SPAs on the Catalyst 6500 Series switch count general frame errors, such as CRC errors and corrupted packets, as errored frames.

To specify errored frame period link-monitoring options, perform this task in interface configuration or template configuration mode:

Command	Purpose
`Router(config-if)# **ethernet oam link-monitor frame-period window** *10000-frame-units*`	(Optional) Specifies the number of frames (in the range 1000–65535, as a multiple of 10000 frames) to be included in the error counting according to the specified thresholds. The default window unit is 1000, or 10000000 frames.

Command	Purpose
Router(config-if)# **ethernet oam link-monitor frame-period threshold low** *low-frames*	(Optional) Specifies the low error frame threshold as a number of frames (in the range 0–65535). If the number of error frames in the window period is equal to or greater than *low-frames*, then the Errored Frame Period Link Event will be generated. The default low threshold is 0 frame errors.
Router(config-if)# **ethernet oam link-monitor frame-period threshold high** {**none** | *high-frames*}	(Optional) Specifies the high error frame threshold as a number of frames (in the range 1–65535). If the number of error frames in the window period is equal to or greater than *high-frames*, a user defined action will be triggered. There is no default for the high threshold, so you must explicitly configure a value to enable it. Use the **none** keyword to disable the high threshold. For more information about configuring a user-defined action, see "Specifying a High-Threshold Action" section on page 10-32.

Specifying Errored Frame Seconds Summary Link-Monitoring Options

The errored frame seconds summary link-monitoring options include the ability to specify a period of time during which tracking of a number of errored-seconds periods (one-second intervals with at least one frame error) occurs, and the high and low thresholds for triggering the Errored Frames Seconds Summary Link Event.

To specify errored frame seconds summary link-monitoring options, perform this task in interface configuration or template configuration mode:

Command	Purpose
Router(config-if)# **ethernet oam link-monitor frame-seconds window** *100-millisecond-units*	(Optional) Specifies a period of time (in the range 100–9000, as a multiple of 100 milliseconds) during which tracking of an errored-seconds period occurs according to the specified thresholds. The default window unit is 100, or 10000 milliseconds.

Command	Purpose
`Router(config-if)# ethernet oam link-monitor frame-seconds threshold low low-errored-seconds`	(Optional) Specifies the low errored seconds threshold as a number of errored seconds (in the range 0–900). If the number of errored seconds in the window period is equal to or greater than *low-errored-seconds*, then the Errored Frame Seconds Summary Link Event will be generated. The default low threshold is 0 error seconds.
`Router(config-if)# ethernet oam link-monitor frame-seconds threshold high {none | high-errored-seconds}`	(Optional) Specifies the high errored seconds threshold as a number of errored seconds (in the range 1–900). If the number of errored seconds in the window period is equal to or greater than *high-errored-seconds*, then a user defined action will be triggered. There is no default for the high threshold, so you must explicitly configure a value to enable it. Use the **none** keyword to disable the high threshold. For more information about configuring a user-defined action, see "Specifying a High-Threshold Action" section on page 10-32.

Specifying Receive CRC Link-Monitoring Options

The receive CRC link-monitoring options include the ability to specify a period of time during which tracking of frames received with CRC occurs, and the high and low thresholds for triggering the error. Receive CRC link-monitoring is a Cisco-specific implementation and is only locally significant to the Ethernet OAM interface on the Catalyst 6500 Series switch.

To specify receive CRC link-monitoring options, perform this task in interface configuration or template configuration mode:

Command	Purpose
`Router(config-if)# ethernet oam link-monitor receive-crc window 100-millisecond-units`	(Optional) Specifies a period of time (in the range 10–1800, as a multiple of 100 milliseconds) during which tracking of frames received with CRC errors occurs according to the specified thresholds. The default window unit is 10, or 1000 milliseconds.

Command	Purpose
`Router(config-if)#` **`ethernet oam link-monitor receive-crc threshold low`** *`low-frames`*	(Optional) Specifies the low CRC threshold as a number of frames (in the range 0–65535). If the number of frames received with CRC errors in the window period is equal to or greater than *low-frames*, then the Receive CRC error will be generated. The default low threshold is 1 frame.
`Router(config-if)#` **`ethernet oam link-monitor receive-crc threshold high`** `{`**`none`** `|` *`high-frames`*`}`	(Optional) Specifies the high CRC threshold as a number of frames (in the range 1–65535). If the number of frames received with CRC errors in the window period is equal to or greater than *high-frames*, a user-defined action will be triggered. There is no default for the high threshold, so you must explicitly configure a value to enable it. Use the **none** keyword to disable the high threshold. For more information about configuring a user-defined action, see "Specifying a High-Threshold Action" section on page 10-32.

Specifying Transmit CRC Link-Monitoring Options

The transmit CRC link-monitoring options include the ability to specify a period of time during which tracking of frames transmitted with CRC occurs, and the high and low thresholds for triggering the error. Transmit CRC link-monitoring is a Cisco-specific error event and is only locally significant to the Ethernet OAM interface on the Catalyst 6500 Series switch.

To specify transmit CRC link-monitoring options, perform this task in interface configuration or template configuration mode:

Command	Purpose
`Router(config-if)#` **`ethernet oam link-monitor transmit-crc window`** *`100-millisecond-units`*	(Optional) Specifies a period of time (in the range 10–1800, as a multiple of 100 milliseconds) during which tracking of frames received with CRC errors occurs according to the specified thresholds. The default window unit is 10, or 1000 milliseconds.

Command	Purpose
`Router(config-if)# ethernet oam link-monitor transmit-crc threshold low low-frames`	(Optional) Specifies the low CRC threshold as a number of frames (in the range 0–65535). If the number of frames transmitted with CRC errors in the window period is equal to or greater than *low-frames*, then the Receive CRC error will be generated. The default low threshold is 1 frame.
`Router(config-if)# ethernet oam link-monitor transmit-crc threshold high {none | high-frames}`	(Optional) Specifies the high CRC threshold as a number of frames (in the range 1–65535). If the number of frames transmitted with CRC errors in the window period is equal to or greater than *high-frames*, a user defined action will be triggered. There is no default for the high threshold, so you must explicitly configure a value to enable it. Use the **none** keyword to disable the high threshold. For more information about configuring a user-defined action, see "Specifying a High-Threshold Action" section on page 10-32.

Specifying a High-Threshold Action

When you configure high thresholds for OAM link-monitoring, you can specify an action to be taken when the high threshold is exceeded.

When configuring high-threshold actions, consider the following guidelines:

- There is no default action.
- If you configure a high threshold but do not configure any corresponding action, only a message appears on the syslog and no other action is taken on the interface.
- If you want to associate different high-threshold actions for different kinds of link-monitoring functions, you can use configuration templates. However, only one configuration template can be applied to any Ethernet OAM interface.
- Only one high-threshold action can be configured for any Ethernet OAM interface.

To configure an action when a high threshold for an error is exceeded on an Ethernet OAM interface, use the following command in interface configuration or template configuration mode:

Command	Purpose
`Router(config-if)# ethernet oam link-monitor high-threshold action {error-disable-interface | failover}`	(Optional) Configures the action when a high-threshold error is exceeded, where: • **error-disable-interface**—Shuts down the Ethernet OAM interface. • **failover**—(EtherChannel interface only) Configures the interface for an automatic failover of traffic from one port in an EtherChannel to another port in the same EtherChannel when one of the ports in the channel exceeds the high error threshold within the specified interval. The port failover only occurs if there is at least one operational port available in the EtherChannel. The failed port will be put into an error disable state. If the failed port is the last port in the EtherChannel, the port will not be put into an error disable state and continues to pass traffic regardless of the type of errors being received. Single, nonchanneling ports go into the error disable state when the error threshold is exceeded within the specified interval.

Configuring Remote Failure Indication Actions

When an RFI event occurs locally, the local client sends an Information OAMPDU to its peer with a bit selected that indicates the type of failure. The Gigabit Ethernet SPAs on the Catalyst 6500 Series switch process all of the following types of Remote Failure Indication (RFI) conditions as defined by IEEE 802.3ah:

- Critical Event—This type of RFI is sent when an unspecified critical event has occurred. These events are vendor specific, and the failure indication might be sent immediately and continuously.
- Dying Gasp—This type of RFI is sent when an unrecoverable condition (for example, a power failure) has occurred. The conditions for a Dying Gasp RFI are vendor specific, and the failure indication might be sent immediately and continuously. The Gigabit Ethernet SPAs on the Catalyst 6500 Series switch generate a Dying Gasp RFI when an interface is error-disabled or administratively shut down. This is the only type of RFI that the Gigabit Ethernet SPAs on the Catalyst 6500 Series switch generate.
- Link Fault—This type of RFI is sent when a loss of signal is detected by the receiver (for example, a peer's laser is malfunctioning). A link fault is sent once per second in the Information OAMPDU. The link fault RFI applies only when the physical sublayer is capable of independent transmit and receive.

When the Gigabit Ethernet SPAs receive an OAMPDU with an RFI bit selected, a syslog message is created providing the failure reason, as shown in the following example:

```
%ETHERNET_OAM-SP-6-RFI: The client on interface Gi1/1 has received a remote failure
indication from its remote peer (failure reason = remote client administratively turned
off)
```

You can configure a response, or action, by the local client to shut down the OAM interface when it receives Information OAMPDUs with a Dying Gasp RFI bit selected.

To configure an error disable action for the local Ethernet OAM interface, use the following command in interface configuration or template configuration mode:

Command	Purpose
`Router(config-if)#` **`ethernet oam remote-failure dying-gasp action error-disable-interface`**	(Optional) Specifies that the local Ethernet OAM interface is shut down upon receipt of an Information OAMPDU from its peer that indicates a Dying Gasp.

Configuring Global Ethernet OAM Options Using a Template

Create configuration templates when you have a common set of link-monitoring or remote-failure characteristics that you want to apply to multiple Ethernet OAM interfaces. Templates simplify Ethernet OAM interface configuration.

Although you can configure multiple configuration templates, only one template can be associated with any single Ethernet OAM interface. You can override any commands defined within a template by explicitly configuring the same command (that is predefined by the template) in interface configuration mode.

To configure global Ethernet OAM interface options using a template, use the following command beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`template`** *`template-name`*	Creates or selects a template and enters template configuration mode, where *template-name* is an up to 32-character string defining the name of the template.

Step 2	`Router(config-template)#` **`ethernet oam link-monitor`** *`command`*
	`or`
	`Router(config-template)#` **`ethernet oam remote-failure`** *`command`*
		Specify one or more **ethernet oam** configuration commands. Repeat this step for the number of commands that you want to configure. For information about link-monitoring commands, see the "Configuring Link-Monitoring Options" section on page 10-26.

Step 3	`Router(config-template)#` **`exit`**	Exit template configuration mode and return to global configuration mode.

	Command	Purpose
Step 4	`Router(config)# interface type slot/subslot/port`	Specifies the Ethernet SPA interface, where: • *type*—Specifies the type of Ethernet interface, such as **gigabitethernet** or **tengigabitethernet**. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 10-4. **Note** Ethernet OAM only can be defined on a main Gigabit Ethernet interface, not on subinterfaces.
Step 5	`Router(config-if)# source template template-name`	Attaches the template called *template-name* and applies the set of configuration commands defined by the named template to the specified interface.

Verifying Ethernet OAM Configuration

To verify the Ethernet OAM configuration, perform this task in privileged EXEC configuration mode:

Command	Purpose
`Router# show ethernet oam discovery [interface type slot/subslot/port]`	Displays information about OAM functions negotiated during the OAM discovery phase of establishing an OAM session, where: • *type*—Specifies the type of Ethernet interface, such as **gigabitethernet** or **tengigabitethernet**. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 10-4.
`Router# show ethernet oam statistics [interface type slot/subslot/port]`	Displays statistics for information OAMPDUs and local and remote faults, where: • *type*—Specifies the type of Ethernet interface, such as **gigabitethernet** or **tengigabitethernet**. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 10-4.

Command	Purpose
`Router# show ethernet oam status [interface type slot/subslot/port]`	Displays information about the link-monitoring configuration and status on the local OAM client, where: • *type*—Specifies the type of Ethernet interface, such as **gigabitethernet** or **tengigabitethernet**. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 10-4.
`Router# show ethernet oam summary`	Displays information about the OAM session with the remote OAM client, where: • *type*—Specifies the type of Ethernet interface, such as **gigabitethernet** or **tengigabitethernet**. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 10-4.

This section includes the following topics:

- Verifying an OAM Session, page 10-36
- Verifying OAM Discovery Status, page 10-36
- Verifying Information OAMPDU and Fault Statistics, page 10-37
- Verifying Link-Monitoring Configuration and Status, page 10-38

Verifying an OAM Session

To verify an OAM session, use the **show ethernet oam summary** command.

The following example shows that the local OAM client is established on the second Gigabit Ethernet SPA interface (1) located in subslot 1 of the SIP installed in chassis slot 6 of the Catalyst 6500 Series switch (Gi6/1/1).

The local client interface is in session with a remote client with MAC address 0012.7fa6.a700 and organizationally unique identifier (OUI) 00000C, which is the OUI for Cisco Systems. The remote client is in active mode, and has established capabilities for link-monitoring and remote loopback for the OAM session.

```
Router# show ethernet oam summary
Symbols:          * - Master Loopback State,  # - Slave Loopback State
Capability codes: L - Link Monitor,  R - Remote Loopback
                  U - Unidirection,  V - Variable Retrieval

  Local                       Remote
Interface       MAC Address    OUI    Mode    Capability

  Gi6/1/1       0012.7fa6.a700 00000C active   L R
```

Verifying OAM Discovery Status

To verify OAM Discovery status on the local client and remote peer, use the **show ethernet oam discovery** command as shown in the following example:

```
Router# show ethernet oam discovery interface gigabitethernet6/1/1

GigabitEthernet6/1/1
```

```
Local client
------------
  Administrative configurations:
    Mode:              active
    Unidirection:      not supported
    Link monitor:      supported (on)
    Remote loopback:   not supported
    MIB retrieval:     not supported
    Mtu size:          1500

  Operational status:
Port status:       operational
    Loopback status:   no loopback
    PDU permission:    any
    PDU revision:      1

Remote client
-------------
  MAC address: 0030.96fd.6bfa
  Vendor(oui): 0x00 0x00 0x0C (cisco)

  Administrative configurations:
    Mode:              active
    Unidirection:      not supported
    Link monitor:      supported
    Remote loopback:   not supported
    MIB retrieval:     not supported
    Mtu size:          1500
```

Verifying Information OAMPDU and Fault Statistics

To verify statistics for information OAMPDUs and local and remote faults, use the **show ethernet oam statistics** command as shown in the following example:

```
Router# show ethernet oam statistics interface gigabitethernet6/1/1

GigabitEthernet6/1/1
Counters:
---------
Information OAMPDU Tx                   : 588806
  Information OAMPDU Rx                   : 988
  Unique Event Notification OAMPDU Tx     : 0
  Unique Event Notification OAMPDU Rx     : 0
  Duplicate Event Notification OAMPDU TX  : 0
  Duplicate Event Notification OAMPDU RX  : 0
  Loopback Control OAMPDU Tx              : 1
  Loopback Control OAMPDU Rx              : 0
  Variable Request OAMPDU Tx              : 0
  Variable Request OAMPDU Rx              : 0
  Variable Response OAMPDU Tx             : 0
  Variable Response OAMPDU Rx             : 0
  Cisco OAMPDU Tx                         : 4
  Cisco OAMPDU Rx                         : 0
  Unsupported OAMPDU Tx                   : 0
  Unsupported OAMPDU Rx                   : 0
  Frames Lost due to OAM                  : 0

Local Faults:
-------------
  0 Link Fault records
  2 Dying Gasp records
    Total dying gasps       : 4
    Time stamp              : 00:30:39
    Total dying gasps       : 3
    Time stamp              : 00:32:39

  0 Critical Event records

Remote Faults:
--------------
  0 Link Fault records
  0 Dying Gasp records
  0 Critical Event records

Local event logs:
-----------------
  0 Errored Symbol Period records
  0 Errored Frame records
  0 Errored Frame Period records
  0 Errored Frame Second records

Remote event logs:
------------------
  0 Errored Symbol Period records
  0 Errored Frame records
  0 Errored Frame Period records
  0 Errored Frame Second records
```

Verifying Link-Monitoring Configuration and Status

To verify link-monitoring configuration and status on the local client, use the **show ethernet oam status** command. The highlighted “Status” field in the following example shows that link-monitoring status is supported and enabled (on).

```
Router# show ethernet oam status interface gigabitethernet6/1/1

GigabitEthernet6/1/1
General
-------
  Mode:                  active
  PDU max rate:          10 packets per second
  PDU min rate:          1 packet per 1 second
  Link timeout:          5 seconds
  High threshold action: no action

Link Monitoring
---------------
  Status: supported (on)

  Symbol Period Error
    Window:              1 million symbols
    Low threshold:       1 error symbol(s)
    High threshold:      none

  Frame Error
    Window:              10 x 100 milliseconds
    Low threshold:       1 error frame(s)
    High threshold:      none
Frame Period Error
    Window:              1 x 100,000 frames
    Low threshold:       1 error frame(s)
    High threshold:      none

  Frame Seconds Error
    Window:              600 x 100 milliseconds
    Low threshold:       1 error second(s)
    High threshold:      none
```

Verifying Status of the Remote OAM Client

To verify the status of a remote OAM client, use the **show ethernet oam summary** and **show ethernet oam status** commands.

To verify the remote client mode and capabilities for the OAM session, use the **show ethernet oam summary** command and observe the values in the Mode and Capability fields. The following example shows that the local client (local interface Gi6/1/1) is connected to the remote client:

```
Router# show ethernet oam summary
Symbols:          * - Master Loopback State,  # - Slave Loopback State
Capability codes: L - Link Monitor,  R - Remote Loopback
                  U - Unidirection,  V - Variable Retrieval

  Local                       Remote
Interface       MAC Address    OUI    Mode     Capability

  Gi6/1/1       0012.7fa6.a700 00000C active    L R
```

Configuring QoS Features on Ethernet SPAs

For information about the QoS features supported by the Ethernet SPAs, see the "Configuring QoS Features on a SIP" section on page 4-33 of Chapter 4, "Configuring the SIPs and SSC."

QoS Configuration Guidelines for the Ethernet SPA

For Fast Ethernet SPAs and the 2-Port Gigabit Ethernet SPA, the following QoS behavior applies:

- In both the ingress and egress directions, all QoS features calculate packet size similarly to how packet size calculation is performed by the FlexWAN and Enhanced FlexWAN modules on the Catalyst 6500 Series switch.
- Specifically, all features consider the IEEE 802.3 Layer-2 headers and the Layer-3 protocol payload. The CRC, interframe gap, and preamble are not included in the packet size calculations.

Note For Fast Ethernet SPAs, QoS cannot change the speed of an interface (for example, Fast Ethernet SPAs cannot change QoS settings whenever an interface speed is changed between 100 Mbps to 10 Mbps). When the speed is changed, the user must also adjust the QoS setting accordingly.

Saving the Configuration

To save your running configuration to nonvolatile random-access memory (NVRAM), perform this task in privileged EXEC configuration mode:

Command	Purpose
Router# **copy running-config startup-config**	Writes the new configuration to NVRAM.

For information about managing your system image and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

Shutting Down and Restarting an Interface on a SPA

You can shut down and restart any of the interface ports on a SPA independently of each other. Shutting down an interface stops traffic and enters the interface into an administratively down state.

There are no restrictions for online insertion and removal (OIR) on Fast Ethernet or Gigabit Ethernet SPAs. Fast Ethernet and Gigabit Ethernet SPAs can be removed from a SIP at any time. SIPs populated with any type of SPAs can be removed from the switch at any time.

If you are preparing for an OIR of a SPA, you do not need to independently shut down each of the interfaces prior to deactivation of the SPA. The **hw-module subslot shutdown** command automatically stops traffic on the interfaces and deactivates them along with the SPA in preparation for OIR.

You also do not need to independently restart any interfaces on a SPA after OIR of a SPA or SIP.

To shut down an interface on a SPA, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# shutdown`	Disables an interface.

To restart an interface on a SPA, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# no shutdown`	Restarts a disabled interface.

Verifying the Interface Configuration

In addition to using the **show running-configuration** command to display your switch configuration settings, you can use the **show interfaces gigabitethernet** command to get detailed information on a per-port basis for your Gigabit Ethernet SPAs, and use the **show interfaces fastethernet** command to get detailed information on a per-port basis for your Fast Ethernet SPAs.

Verifying Per-Port Interface Status

To find detailed interface information on a per-port basis for the Fast Ethernet and Gigabit Ethernet SPAs, use the **show interfaces fastethernet** and **show interfaces gigabitethernet** commands, respectively. For a description of the command output, see the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX*.

The following example provides sample output for interface port 1 on the SPA located in the top subslot (0) of the SIP that is installed in slot 2 of the Catalyst 6500 Series switch:

```
Router# show interfaces gigabitethernet 2/0/1
GigabitEthernet2/0/1 is down, line protocol is down
  Hardware is GigEther SPA, address is 000a.f330.2e40 (bia 000a.f330.2e40)
  Internet address is 2.2.2.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  Full-duplex, 1000Mb/s, link type is force-up, media type is SX
  output flow-control is on, input flow-control is on
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 03:18:49, output 03:18:44, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1703 packets input, 638959 bytes, 0 no buffer
     Received 23 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 1670 multicast, 0 pause input
     1715 packets output, 656528 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

Configuration Examples

This section includes the following configuration examples:

Basic Interface Configuration Example

The following example shows how to enter global configuration mode to specify the interface that you want to configure, configure an IP address for the interface, and save the configuration. This example configures interface port 1 on the SPA that is located in subslot 0 of the SIP, that is installed in slot 3 of the Catalyst 6500 Series switch:

```
!Enter global configuration mode
!
Router# configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
!
! Specify the interface address
!
Router(config)# interface gigabitethernet 3/0/1
!
! Configure an IP address
!
Router(config-if)# ip address 192.168.50.1 255.255.255.0
!
! Start the interface
!
Router(config-if)# no shutdown
!
! Save the configuration to NVRAM
!
Router(config-if)# exit
Router# copy running-config startup-config
```

MAC Address Configuration Example

The following example changes the default MAC address on the interface to 1111.2222.3333:

```
!Enter global configuration mode
!
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
!
! Specify the interface address
!
Router(config)# interface gigabitethernet 3/0/1
!
! Modify the MAC address
!
Router(config-if)# mac-address 1111.2222.3333
```

MTU Configuration Example

The following example sets the interface MTU to 9216 bytes:

Note The SPA automatically adds an additional 38 bytes to the configured interface MTU size.

```
!Enter global configuration mode
!
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
!
! Specify the interface address
!
Router(config)# interface gigabitethernet 3/0/1
!
! Configure the interface MTU
!
Router(config-if)# mtu 9216
```

VLAN Configuration Example

The following example creates subinterface number 268 on SPA interface port 2 (the third port), and configures the subinterface on the VLAN with ID number 268 using IEEE 802.1Q encapsulation:

Note The SPA does not support ISL encapsulation.

```
!Enter global configuration mode
!
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
!
! Specify the interface address
!
Router(config)# interface gigabitethernet 3/0/1.268
!
! Configure dot1q encapsulation and specify the VLAN ID
!
Router(config-subif)# encapsulation dot1q 268
```

MPLSoGRE and mVPNoGRE Configuration Example

The following example shows how to configure the MPLSoGRE and mVPNoGRE feature on a Cisco 7600 SIP-400 interface or subinterface; however, this example uses a Cisco 7600 SIP-400 interface that does *not* support subinterfaces:

```
! Enter global configuration mode.
!
Router# configure terminal
! Enter configuration commands, one per line. End with CNTL/Z.
!
! Specify the Gigabit Ethernet interface to configure.
!
Router(config)# interface gigabitethernet 2/0/0
! Attach a GRE Tunnel to a Cisco 7600 SIP-400 subinterface.
!
Router(config-if)# tunnel-interface tu1
! Define the IP traffic that should be tunneled.
!
Router(config-if-ti)# ip route 10.0.0.1 255.255.255.0
Router(config-if-ti)# exit
```

When **tunnel-interface** is configured on the Cisco 7600 SIP-400 interface or subinterface, **ip pim sparse-mode** and **tag-switching ip** are automatically added to the interface. A static route to IP address contained on the **ip route** command is internally created. The following example shows the output of the **show running interface** command after adding or configuring **tunnel-interface**; however, this example uses a Cisco 7600 SIP-400 interface that does *not* support subinterfaces:

```
Router# show running interface gigabitethernet 2/0/0
!
interface gigabitethernet2/0/0
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-mode
 no keepalive
 tunnel-interface Tunnel1
    ip route 10.11.0.1 255.255.255.0
    exit-tunnel-interface
 tag-switching ip
 clock source internal
end
```

Note You do not need to configure a static route (globally or on the tunnel) to the BGP neighbor on the Catalyst 6500 Series switch. This is automatically done by the **ip route** command under the **tunnel-interface** command on the Cisco 7600 SIP-400 interface or subinterface.

The following example illustrates the tunnel interface configuration on the Catalyst 6500 Series switch:

```
interface Tunnel0
 ip address 10.0.0.1 255.255.255.0
 ip pim sparse-dense-mode
 mpls ip
 tunnel source 22.22.22.22
 tunnel destination 44.44.44.44
```

EoMPLS Configuration Example

The following example shows how to configure software-based EoMPLS:

```
! Enter global configuration mode.
!
Router# configure terminal
! Enter configuration commands, one per line. End with CNTL/Z.
Router# vlan 101
!
Router(config)# interface VLAN101
Router(config-if)# xconnect 7.7.7.7 73829 encapsulation MPLS
!
Router(config)# interface gigabitethernet 4/1/0.1
Router(config-subif)# encapsulation dot1Q 100
```

The following example shows the commands to configure Scalable EoMPLS (only for a Cisco 7600 SIP-400 Ethernet interface):

```
Router(config)# interface GigabitEthernet 1/2/1
Router(config-if)# no ip address
Router(config-if)# no cdp enable
!
Router(config-if)# interface GigabitEthernet 1/2/1.2
Router(config-subif)# encapsulation dot1Q 2
Router(config-subif)# xconnect 5.5.5.5 20002 encapsulation mpls
!
[Snip ...]
!
Router(config-if)# interface GigabitEthernet 1/2/1.4095
Router(config-subif)# encapsulation dot1Q 4095
Router(config-subif)# xconnect 5.5.5.5 24095 encapsulation mpls
```

The following example shows how to configure hardware EoMPLS (other Ethernet interfaces):

```
Router(config)# interface GigabitEthernet 1/1
Router(config-if)# no ip address
Router(config-if)# no cdp enable
!
Router(config-subif)# interface GigabitEthernet 1/1.2
Router(config-subif)# encapsulation dot1Q 2
Router(config-subif)# xconnect 5.5.5.5 10002 encapsulation mpls
!
[Snip ...]
Router(config)# interface GigabitEthernet 1/1.3095
Router(config-subif)# encapsulation dot1Q 3095
Router(config-subif)# xconnect 5.5.5.5 13095 encapsulation mpls
!
```

Changing the Speed of a Fast Ethernet SPA Configuration Example

The following example shows how to change the speed of a Fast Ethernet SPA:

Note In order to change the speed of a Fast Ethernet SPA, you must disable autonegotiation.

```
Router# show run interface fastethernet 5/0/1
Building configuration...
Current configuration : 86 bytes
!
! Disable Autonegotiation
!
interface FastEthernet5/0/1
ip address 10.1.0.2 255.255.0.0
negotiation auto
end
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/0/1
Router(config-if)# no negotiation auto
Router(config-if)# speed 10
Router(config-if)# end
Router# show run interface fastethernet 5/01
Building configuration...
Current configuration : 112 bytes
!
interface FastEthernet 5/0/1
ip address 10.1.0.2 255.255.0.0
speed 10
duplex full
no negotiation auto
end
Router# show interface fastethernet 5/0/1
FastEthernet5/0/1 is up, line protocol is up
Hardware is FastEthernet SPA, address is 000a.8b3e.cc00 (bia 000a.8b3e.cc00)
Internet address is 10.1.0.2/16
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive not supported
Full-duplex, 10Mb/s
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:04, output 00:00:04, output hang never
Last clearing of "show interface" counters 1d00h
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
```

```
1608 packets input, 547102 bytes, 0 no buffer
Received 1 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog
0 input packets with dribble condition detected
1606 packets output, 548403 bytes, 0 underruns

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/0/1
Router(config-if)# speed 100
Router(config-if)# end
Router#
*Apr 25 21:10:36: %SYS-5-CONFIG_I: Configured from console by console
Router# show interface fastethernet 5/0/1
FastEthernet5/0/1 is down, line protocol is down
Hardware is FastEthernet SPA, address is 000a.8b3e.cc00 (bia 000a.8b3e.cc00)
Internet address is 10.1.0.2/16
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive not supported
Full-duplex, 100Mb/s
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:23, output 00:00:22, output hang never
Last clearing of "show interface" counters 1d00h
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
1608 packets input, 547102 bytes, 0 no buffer
Received 1 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
```

CHAPTER 11

Troubleshooting the Fast Ethernet and Gigabit Ethernet SPAs

This chapter describes techniques that you can use to troubleshoot the operation of your Fast Ethernet or Gigabit Ethernet SPAs.

It includes the following sections:

- General Troubleshooting Information, page 11-1
- Performing Basic Interface Troubleshooting, page 11-2
- Understanding SPA Automatic Recovery, page 11-6
- Configuring the Interface for Internal and External Loopback, page 11-7
- Using the Cisco IOS Event Tracer to Troubleshoot Problems, page 11-8
- Preparing for Online Insertion and Removal of a SPA, page 11-9

The first section provides information about basic interface troubleshooting. If you are having a problem with your SPA, use the steps in the "Performing Basic Interface Troubleshooting" section on page 11-2 to begin your investigation of a possible interface configuration problem.

To perform more advanced troubleshooting, see the other sections in this chapter.

General Troubleshooting Information

This section describes general information for troubleshooting SIPs and SPAs. It includes the following sections:

- Using Debug Commands, page 11-1
- Using show Commands, page 11-2

Using Debug Commands

Along with the other **debug** commands supported on the Catalyst 6500 Series switch, you can obtain specific debug information for SPAs on the Catalyst 6500 Series switch using the **debug hw-module subslot** privileged exec command.

The **debug hw-module subslot** command is intended for use by Cisco Systems technical support personnel. For more information about the **debug hw-module subslot** command and other **debug** commands, see the *Cisco IOS Debug Command Reference, Release 12.2*.

Caution Because the debugging output is assigned high priority in the CPU process, it can cause the system to become unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the possibility that increased **debug** command processing overhead will affect system use.

For more information about other commands that can be used on a Catalyst 6500 Series switch, refer to the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

Using show Commands

There are several **show** commands that you can use to monitor and troubleshoot the SIPs and SPAs on the Catalyst 6500 Series switch. This chapter describes using the **show interfaces** command to perform troubleshooting of your SPA.

Also see Chapter 10, "Configuring the Fast Ethernet and Gigabit Ethernet SPAs" for additional information about these **show** commands.

Performing Basic Interface Troubleshooting

You can perform most of the basic interface troubleshooting using the **show interfaces gigabitethernet** or **show interfaces tengigabitethernet** command and examining several areas of the output to determine how the interface is operating.

The following example shows output from both the **show interfaces gigabitethernet** and **show interfaces tengigabitethernet** commands with some of the significant areas of the output to observe shown in bold:

```
Router# show interfaces gigabitethernet 2/0/1
GigabitEthernet2/0/1 is down, line protocol is down
  Hardware is GigEther SPA, address is 000a.f330.2e40 (bia 000a.f330.2e40)
  Internet address is 2.2.2.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  Full-duplex, 1000Mb/s, link type is force-up, media type is SX
  output flow-control is on, input flow-control is on
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 03:18:49, output 03:18:44, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1703 packets input, 638959 bytes, 0 no buffer
     Received 23 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 1670 multicast, 0 pause input
     1715 packets output, 656528 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out

Router# show interfaces tengigabitethernet7/0/0
TenGigabitEthernet7/0/0 is up, line protocol is up (connected)
  Hardware is TenGigEther SPA, address is 0000.0c00.0102 (bia 000f.342f.c340)
  Internet address is 15.1.1.2/24
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  Full-duplex, 10Gb/s
  input flow-control is on, output flow-control is on
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:10, output hang never
  Last clearing of "show interface" counters 20:24:30
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  L2 Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes
  L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes mcast
  L3 out Switched: ucast: 0 pkt, 0 bytes mcast: 0 pkt, 0 bytes
     237450882 packets input, 15340005588 bytes, 0 no buffer
     Received 25 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     1676 packets output, 198290 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

To verify that your interface is operating properly, perform the task in Table 11-1,

Table 11-1 Basic Interface Troubleshooting Steps

	Action	Example
Step 1	From global configuration mode, enter the **show interfaces gigabitethernet** or the **show interfaces tengigabitethernet** command.	Router# **show interfaces gigabitethernet 2/0/1** Router# **show interfaces tengigabitethernet7/0/0**
Step 2	Verify that the interface is up.	Router# **show interfaces gigabitethernet 2/0/1** **GigabitEthernet2/0/1 is up**, line protocol is up Router# **show interfaces tengigabitethernet7/0/0** **TenGigabitEthernet7/0/0 is up,** line protocol is up (connected)
Step 3	Verify that the line protocol is up.	Router# **show interfaces gigabitethernet 2/0/1** GigabitEthernet2/0/1 is up, **line protocol is up** Router# **show interfaces tengigabitethernet7/0/0** TenGigabitEthernet7/0/0 is up, **line protocol is up (connected)**

Table 11-1 Basic Interface Troubleshooting Steps

<table>
<tr><th></th><th>Action</th><th>Example</th></tr>
<tr><td>Step 4</td><td>Verify that the interface duplex mode matches the remote interface configuration.</td><td>The following example shows that the local interface is currently operating in full-duplex mode:
<pre>
Router# show interfaces gigabitethernet 2/0/1
[text omitted]

  Keepalive not supported
  Full-duplex, 1000Mb/s, link type is force-up, media
type is SX

Router# show interfaces tengigabitethernet7/0/0
[text omitted]

Keepalive not supported
  Full-duplex, 10Gb/s
</pre></td></tr>
<tr><td>Step 5</td><td>Verify that the interface speed matches the speed on the remote interface.</td><td>The following example shows that the local interface is currently operating at 100Mbps (GigabitEthernet) or 10 Gbps (Ten GigabitEthernet):
<pre>
Router# show interfaces gigabitethernet 2/0/1
.
.
.
  Keepalive not supported
  Full-duplex, 1000Mb/s, link type is force-up, media
type is SX
.
.
Router# show interfaces tengigabitethernet7/0/0
[text omitted]

  Full-duplex, 10Gb/s
</pre></td></tr>
<tr><td>Step 6</td><td>Observe the output hang status on the interface.</td><td><pre>
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 03:18:49, output 03:18:44, output hang never
</pre></td></tr>
<tr><td>Step 7</td><td>Observe the CRC counter.</td><td><pre>
Router# show interfaces gigabitethernet 2/0/1
</pre></td></tr>
<tr><td>Step 8</td><td>Observe the late collision counter.</td><td><pre>
     0 output errors, 0 collisions, 4 interface resets
     0 babbles, 0 late collision, 0 deferred
</pre></td></tr>
<tr><td>Step 9</td><td>Observe the carrier signal counters.</td><td><pre>
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped
     out
</pre></td></tr>
</table>

For more information about the verification steps in and possible responses to correct detected problems, see the following sections:

- Verifying the Interface is Up, page 11-5
- Verifying the Line Protocol is Up, page 11-5
- Verifying Output Hang Status, page 11-5
- Verifying the CRC Counter, page 11-5
- Verifying Late Collisions, page 11-6
- Verifying the Carrier Signal, page 11-6

Verifying the Interface is Up

In the output from the **show interfaces gigabitethernet** or **show interfaces tengigabitethernet** command, verify that the interface is up. If the interface is down, perform the following corrective actions:

- If the interface is administratively down, use the **no shutdown** interface configuration command to enable the interface.
- Be sure that the cable is fully connected.
- Verify that the cable is not bent or damaged. If the cable is bent or damaged, the signal will be degraded.
- Verify that a hardware failure has not occurred. Observe the LEDs to confirm the failure. See the other troubleshooting sections of this chapter, and refer to the *Cisco 7600 Series Router SIP, SSC, and SPA Hardware Installation Guide*. If the hardware has failed, replace the SPA as necessary.

Verifying the Line Protocol is Up

In the output from the **show interfaces gigabitethernet** or **show interfaces tengigabitethernet** command, verify that the line protocol is up. If the line protocol is down, the line protocol software processes have determined that the line is unusable.

Perform the following corrective actions:

- Swap the cable.
- Check the local and remote interface for misconfiguration.
- Verify that a hardware failure has not occurred. Observe the LEDs to confirm the failure. See the other troubleshooting sections of this chapter, and refer to the *Cisco 7600 Series Router SIP, SSC, and SPA Hardware Installation Guide*. If the hardware has failed, replace the SPA as necessary.

Verifying Output Hang Status

In the output from the **show interfaces gigabitethernet** or **show interfaces tengigabitethernet** command, observe the value of the output hang field.

The output hang provides the number of hours, minutes, and seconds since the last reset caused by a lengthy transmission. When the number of hours the field exceeds 24 hours, the number of days and hours is shown. If the field overflows, asterisks are printed. The field shows a value of never if no output suspensions have occurred.

Verifying the CRC Counter

In the output from the **show interfaces gigabitethernet** or **show interfaces tengigabitethernet** command, observe the value of the CRC counter. Excessive noise will cause high CRC errors accompanied by a low number of collisions.

Perform the following corrective actions if you encounter high CRC errors:

- Check the cables for damage.
- Verify that the correct cables are being used for the SPA interface.

Verifying Late Collisions

In the output from the **show interfaces gigabitethernet** or **show interfaces tengigabitethernet** command, observe the value of the late collision counter.

Perform the following corrective actions if you encounter late collisions on the interface:

- Verify that the duplex mode on the local and remote interface match. Late collisions occur when there is a duplex mode mismatch.
- Verify the length of the Ethernet cables. Late collisions result from cables that are too long.

Verifying the Carrier Signal

In the output from the **show interfaces gigabitethernet** or **show interfaces tengigabitethernet** command, observe the value of the carrier signal counters. The lost carrier counter shows the number of times that the carrier was lost during transmission. The no carrier counter shows the number of times that the carrier was not present during transmission.

Carrier signal resets can occur when an interface is in loopback mode or shut down.

Perform the following corrective actions if you observe the carrier signal counter incrementing outside of these conditions:

- Check the interface for a malfunction.
- Check for a cable problem.

Understanding SPA Automatic Recovery

When the Gigabit Ethernet SPAs encounter thresholds for certain types of errors and identifies a fatal error, the SPA initiates an automatic recovery process.

You do not need to take any action unless the error counters reach a certain threshold, and multiple attempts for automatic recovery by the SPA fail.

The GigabitEthernet SPAs might perform automatic recovery for the following types of errors:

- SPI4 TX/RX out of frame
- SPI4 TX train valid
- SPI4 TX DIP4
- SPI4 RX DIP2

When Automatic Recovery Occurs

If the SPI4 errors occur more than 25 times within 10 milliseconds, the SPA automatically deactivates and reactivates itself. Error messages are logged on the console indicating the source of the error and the status of the recovery.

If Automatic Recovery Fails

If the SPA attempts automatic recovery more than five times in an hour, then the SPA deactivates itself and remains deactivated.

To troubleshoot automatic recovery failure for a SPA, perform the following steps:

Step 1 Use the **show hw-module subslot** *slot/subslot* **oir** command to verify the status of the SPA.
The status is shown as failed if the SPA has been powered off due to five consecutive failures.

Step 2 If you verify that automatic recovery has failed, perform OIR of the SPA.
For information about performing an OIR, see the "Preparing for Online Insertion and Removal of a SPA" section on page 11-9.

Step 3 If reseating the SPA after OIR does not resolve the problem, replace the SPA hardware.

Configuring the Interface for Internal and External Loopback

Loopback support is useful for testing the interface without connectivity to the network, or for diagnosing equipment malfunctions between the interface and a device. The Gigabit Ethernet SPAs support both an internal and an external loopback mode. The external loopback mode requires the use of a loopback cable and implements a loopback through the transceiver on the SPA.

You can also configure an internal loopback without the use of a loopback cable that implements a loopback at the PHY device internally on a Gigabit Ethernet interface port, or at the MAC device internally on a Gigabit Ethernet interface port. By default, loopback is disabled.

Configuring the Interface for Internal Loopback

Different Gigabit Ethernet interfaces use different loopback commands.

To enable internal loopback at the PHY device for an interface on a SPA, perform one of these tasks beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# loopback`	Enables an interface for internal loopback on the Gigabit Ethernet SPA.
`Router(config-if)# loopback internal`	Enables an interface for internal loopback on the Gigabit Ethernet SPA.

Configuring the Interface for External Loopback

Before beginning external loopback testing, remember that the external loopback mode requires the use of a loopback cable.

To enable external loopback, use the following command beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# **loopback external**`	Enables an interface for external loopback on the Gigabit Ethernet SPA.

Verifying Loopback Status

To verify whether loopback is enabled on an interface port on a SPA, use the **show interfaces gigabitethernet** or **show interfaces tengigabitethernet** privileged EXEC command and observe the value shown in the loopback field.

The following example shows that loopback is disabled for interface port 0 (the first port) on the SPA installed in the top (0) subslot of the SIP that is located in slot 3 of the Catalyst 6500 Series switch:

```
Router# show interfaces gigabitethernet 3/0/0
GigabitEthernet3/0/0 is up, line protocol is up
  Hardware is GigMac 1 Port 10 GigabitEthernet, address is 0008.7db3.8dfe (bia )
  Internet address is 10.0.0.2/24
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set
.
.
.
Router# show interfaces tengigabitethernet7/0/0
TenGigabitEthernet7/0/0 is up, line protocol is up (connected)
  Hardware is TenGigEther SPA, address is 0000.0c00.0102 (bia 000f.342f.c340)
  Internet address is 15.1.1.2/24
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
```

Using the Cisco IOS Event Tracer to Troubleshoot Problems

Note This feature is intended for use as a software diagnostic tool and should be configured only under the direction of a Cisco Technical Assistance Center (TAC) representative.

The Event Tracer feature provides a binary trace facility for troubleshooting Cisco IOS software. This feature gives Cisco service representatives additional insight into the operation of the Cisco IOS software and can be useful in helping to diagnose problems in the unlikely event of an operating system malfunction or, in the case of redundant systems, Route Processor switchover.

Event tracing works by reading informational messages from specific Cisco IOS software subsystem components that have been preprogrammed to work with event tracing, and by logging messages from those components into system memory. Trace messages stored in memory can be displayed on the screen or saved to a file for later analysis.

The SPAs currently support the “spa” component to trace SPA OIR-related events.

For more information about using the Event Tracer feature, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/evnttrcr.html

Preparing for Online Insertion and Removal of a SPA

The Catalyst 6500 Series switch supports online insertion and removal (OIR) of the SIP, in addition to each of the SPAs. Therefore, you can remove a SIP with its SPAs still intact, or you can remove a SPA independently from the SIP, leaving the SIP installed in the switch.

This means that a SIP can remain installed in the switch with one SPA remaining active, while you remove another SPA from one of the SIP subslots. If you are not planning to immediately replace a SPA into the SIP, then be sure to install a blank filler plate in the subslot. The SIP should always be fully installed with either functional SPAs or blank filler plates.

For more information about activating and deactivating SPAs in preparation for OIR, see the “Preparing for Online Insertion and Removal of SIPs and SPAs” topic in the “Troubleshooting a SIP” chapter in this guide.

PART 5

Packet over SONET Shared Port Adapters

CHAPTER 12

Overview of the POS SPAs

This chapter provides an overview of the release history, and feature and Management Information Base (MIB) support for the Packet over SONET (POS) SPAs on the Catalyst 6500 Series switch.

This chapter includes the following sections:

Release History

Release	Modification
Cisco IOS Release 12.2(33)SXI	Support for the 2-Port OC-48c/STM-16 POS SPA was introduced on the Cisco 7600 SIP-600 on the Catalyst 6500 series switch.
Cisco IOS Release 12.2(33)SXH	Support for the 1-Port OC-48 POS/RPR SPA with SFP Optics was introduced on the Cisco 7600 SIP-400 on the Catalyst 6500 series switch.
Cisco IOS Release 12.2(18)SXF10	Support for the 1-Port OC-48c/STM-16 POS SPA was introduced on the Cisco 7600 SIP-400 on the Catalyst 6500 series switch.
Cisco IOS Release 12.2(18)SXF2	Support for the 1-Port OC-192c/STM-64 POS/RPR VSR Optics SPA was introduced on the Cisco 7600 SIP-600 on the Cisco 7600 series router and Catalyst 6500 series switch.

Cisco IOS Release 12.2(18)SXF	Support for the following hardware was introduced on the Cisco 7600 series router and Catalyst 6500 series switch: • 1-Port OC-192c/STM-64 POS/RPR SPA • 1-Port OC-192c/STM-64 POS/RPR XFP SPA
Cisco IOS Release 12.2(18)SXE	Support for the following hardware was introduced on the Cisco 7600 series router and Catalyst 6500 series switch: • 2-Port OC-3c/STM-1 POS SPA • 4-Port OC-3c/STM-1 POS SPA • 1-Port OC-12c/STM-4 POS SPA

POS Technology Overview

Packet-over-SONET is a high-speed method of transporting IP traffic between two points. This technology combines the Point-to-Point Protocol (PPP) with Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) interfaces.

SONET is an octet-synchronous multiplex scheme defined by the American National Standards Institute (ANSI) standard (T1.1051988) for optical digital transmission at hierarchical rates from 51.840 Mbps to 2.5 Gbps (Synchronous Transport Signal, STS-1 to STS-48) and greater. SDH is an equivalent international standard for optical digital transmission at hierarchical rates from 155.520 Mbps (Synchronous Transfer Mode-1 [STM-1]) to 2.5 Gbps (STM-16) and greater.

SONET specifications have been defined for single-mode fiber and multimode fiber. The POS SPAs on the Catalyst 6500 Series switch allow transmission over both single-mode and multimode fiber at various optical carrier rates.

SONET/SDH transmission rates are integral multiples of 51.840 Mbps. The following transmission multiples are currently specified and used on the POS SPAs on the Catalyst 6500 Series switch:

- OC-3c/STM-1—155.520 Mbps
- OC-12c/STM-4—622.080 Mbps
- OC-48c/STM-16—2.488 Gbps
- OC-192c/STM-64—9.953 Gbps

Supported Features

This section provides a list of some of the primary features supported by the POS SPA hardware and software:

- Jumbo frames (up to 9216 bytes)
- Online insertion and removal (OIR) from the SIP, or OIR of the SIP with the SPA inserted
- Small form-factor pluggable (SFP) optics module OIR
- Field-programmable gate array (FPGA) upgrade support

The POS SPAs also support the following groups of features:

SONET/SDH Compliance Features

This section lists the SONET/SDH compliance features supported by the POS SPAs on the Catalyst 6500 Series switch:

- 1+1 SONET Automatic Protection Switching (APS) as per G.783 Annex A
- 1+1 SDH Multiplex Section Protection (MSP) as per G.783 Annex A
- American National Standards Institute (ANSI) T1.105
- ITU-T G.707, G.783, G.957, G.958
- Telcordia GR-253-CORE: SONET Transport Systems: Common Generic Criteria
- Telcordia GR-1244: Clocks for the Synchronized Network: Common Generic Criteria

SONET/SDH Error, Alarm, and Performance Monitoring Features

This section lists the SONET/SDH error, alarm, and performance monitoring features supported by the POS SPAs on the Catalyst 6500 Series switch:

- Signal failure bit error rate (SF-BER)
- Signal degrade bit error rate (SD-BER)
- Signal label payload construction (C2)
- Path trace byte (J1)
- Section:
  - Loss of signal (LOS)
  - Loss of frame (LOF)
  - Error counts for B1
  - Threshold crossing alarms (TCA) for B1
- Line:
  - Line alarm indication signal (LAIS)
  - Line remote defect indication (LRDI)
  - Line remote error indication (LREI)
  - Error counts for B2
  - Threshold crossing alarms (TCA) for B2
- Path:
  - Path alarm indication signal (PAIS)
  - Path remote defect indication (PRDI)

- Path remote error indication (PREI)
- Error counts for B3
- Threshold crossing alarms (TCA) for B3
- Loss of pointer (LOP)
- New pointer events (NEWPTR)
- Positive stuffing event (PSE)
- Negative stuffing event (NSE)

SONET/SDH Synchronization Features

This section lists the SONET/SDH synchronization features supported by the POS SPAs on the Catalyst 6500 Series switch:

- Local (internal) timing (for inter-router connections over dark fiber or Wavelength Division Multiplex [WDM] equipment)
- Loop (line) timing (for connecting to SONET/SDH equipment)
- +/– 20 ppm clock accuracy over full operating temperature

WAN Protocol Features

This section lists the WAN protocols supported by the POS SPAs on the Catalyst 6500 Series switch:

- RFC 1661, *The Point-to-Point Protocol (PPP)*
- RFC 1662, *PPP in HDLC framing*
- RFC 2615, *PPP over SONET/SDH* (with 1+x43 self-synchronous payload scrambling)
- RFC 3518, *Point-to-Point Protocol (PPP) Bridging Control Protocol (BCP)*—See Table 12-1for BCP feature restrictions on the Catalyst 6500 series switch
- Cisco Protect Group Protocol over UDP/IP (Port 1972) for APS and MSP
- Multiprotocol Label Switching (MPLS)

Network Management Features

This section lists the network management features supported by the POS SPAs on the Catalyst 6500 Series switch:

- Simple Network Management Protocol (SNMP) Management Information Base (MIB) counters
- Local (diagnostic) loopback
- Network loopback
- NetFlow Data Export
- IP over the Section Data Communications Channel (SDCC) —See Table 12-1 for SDCC feature restrictions on the Catalyst 6500 series switch
- RFC 3592 performance statistics for timed intervals (current, 15-minute, multiple 15-minute, and 1-day intervals):

- Regenerator section
- Multiplex section
- Path errored seconds
- Severely errored seconds
- Severely errored framed seconds

Restrictions

Note For other SIP-specific features and restrictions see also Chapter 3, "Overview of the SIPs and SSC."

Table 12-1 provides information about POS feature compatibility and restrictions by SIP and SPA combination.

Table 12-1 POS Feature Compatibility and Restrictions by SIP and SPA Combination

Feature	Cisco 7600 SIP-200	Cisco 7600 SIP-400	Cisco 7600 SIP-600
Bridge Control Protocol (BCP)	2-Port and 4-Port OC-3c/STM-1 POS SPA—Supported.	• 1-Port OC-12c/STM-4 POS SPA—Supported. • 2-Port and 4-Port OC-3c/STM-1 POS SPA—Supported. • 1-Port OC-48c/STM-16 POS SPA—Supported.	Not supported on any POS SPAs.
Dynamic Packet Transport (DPT), which includes RPR/SRP	Not supported on any POS SPAs.	Not supported on any POS SPAs.	Not supported on any POS SPAs.
Frame Relay	Supported on all POS SPAs.	Supported on all POS SPAs.	Not supported on any POS SPAs.
Multilink PPP	Not supported on any OC-3 POS SPAs.	Not supported on any OC-3 POS SPAs.	Not supported on any OC-3 POS SPAs.
Section Data Communications Channel (SDCC)	• 2-Port OC-3c/STM-1 POS SPA—Supported. • 4-Port OC-3c/STM-1 POS SPA—SDCC is supported on up to two ports.	• 2-Port OC-3c/STM-1 POS SPA—Supported. • 4-Port OC-3c/STM-1 POS SPA—SDCC is supported on up to two ports. • 1-Port OC-12c/STM-4 POS SPA—Supported. • 1-Port OC-48c/STM-16 POS SPA—Not supported.	Not supported on any POS SPAs.

Supported MIBs

The following MIBs are supported in Cisco IOS Release 12.2(18)SXF2 for the 2-Port and 4-Port OC-3c/STM-1 POS SPA, 1-Port OC-12c/STM-4 POS SPA, 1-Port OC-192c/STM-64 POS/RPR SPA, 1-Port OC-192c/STM-64 POS/RPR XFP SPA, and 1-Port OC-192c/STM-64 POS/RPR VSR Optics SPA on the Catalyst 6500 Series switch:

- CISCO-APS-MIB
- CISCO-ENTITY-ASSET-MIB
- CISCO-ENTITY-FRU-CONTROL-MIB
- CISCO-ENVMON-MIB (For NPEs, NSEs, line cards, and MSCs only)
- CISCO-EXTENDED-ENTITY-MIB
- CISCO-OPTICAL-MIB
- ENTITY-MIB
- OLD-CISCO-CHASSIS-MIB
- IF-MIB
- SONET-MIB (RFC 2558, *Definitions of Managed Objects for SONET/SDH Interface Type*)

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

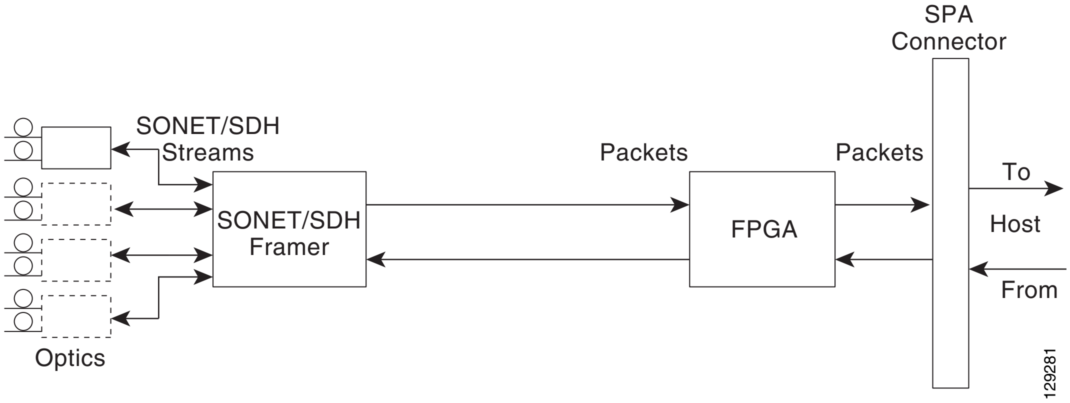
SPA Architecture

This section provides an overview of the architecture of the POS SPAs and describes the path of a packet in the ingress and egress directions. Some of these areas of the architecture are referenced in the SPA software and can be helpful to understand when troubleshooting or interpreting some of the SPA CLI and **show** command output.

4-Port OC-3c/STM-1 POS SPA Architecture

Figure 12-1 identifies some of the hardware devices that are part of the POS SPA architecture. The figure shows the four ports that are supported by the 4-Port OC-3c/STM-1 POS SPA only.

Figure 12-1 4-Port OC-3c/STM-1 POS SPA Architecture



Every incoming and outgoing packet on the 4-Port OC-3c/STM-1 POS SPA goes through the SONET/SDH framer and field-programmable gate array (FPGA) devices.

Path of a Packet in the Ingress Direction

The following steps describe the path of an ingress packet through the 4-Port OC-3c/STM-1 POS SPA:

1. The framer receives SONET/SDH streams from the SFP optics, extracts clocking and data, and processes the section, line, and path overhead.
2. The framer extracts the POS frame payload and verifies the frame size and frame check sequence (FCS).
3. The framer passes valid frames to the field-programmable gate array (FPGA) on the SPA.
4. The FPGA on the SPA transfers frames to the host through the SPI4.2 bus for further processing and switching.

Path of a Packet in the Egress Direction

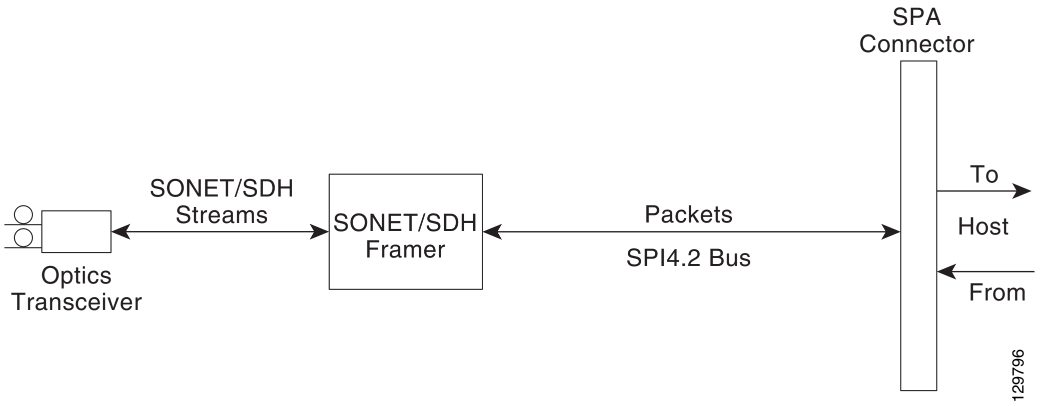
The following steps describe the path of an egress packet through the 4-Port OC-3c/STM-1 POS SPA:

1. The host sends packets to the FPGA on the SPA using the SPI4.2 bus.
2. The FPGA on the SPA stores the data in the appropriate channel's first-in first-out (FIFO) queue.
3. The FPGA on the SPA passes the packet to the framer.
4. The framer accepts the data and stores it in the appropriate channel queue.
5. The framer adds the FCS and SONET/SDH overhead.
6. The framer sends the data to the SFP optics for transmission onto the network.

1-Port OC-192c/STM-64 POS/RPR XFP SPA Architecture

Figure 12-2 identifies the primary hardware devices that are part of the POS SPA architecture. The figure shows a single optics transceiver supported by both of the POS SPAs. However, the 1-Port OC-192c/STM-64 POS/RPR SPA and 1-Port OC-192c/STM-64 POS/RPR VSR Optics SPA support fixed optics, while the 1-Port OC-192c/STM-64 POS/RPR XFP SPA supports XFP optics. The path of a packet remains the same except for where the optic transceiver support resides.

Figure 12-2 1-Port OC-192c/STM-64 POS/RPR XFP SPA Architecture



In POS mode, every incoming and outgoing packet on the OC-192 POS SPAs goes through the SONET/SDH framer and SPI4.2 interface.

Path of a Packet in the Ingress Direction

The following steps describe the path of an ingress packet through the 1-Port OC-192c/STM-64 POS/RPR XFP SPA:

1. The framer receives SONET/SDH streams from the XFP optics, extracts clocking and data, and processes the section, line, and path overhead.
2. The framer extracts the POS frame payload and verifies the frame size and frame check sequence (FCS).
3. The framer passes valid frames to the System Packet Level Interface 4.2 (SPI4.2) interface on the SPA.
4. The SPI4.2 interface transfers frames to the host through the SPI4.2 bus for further processing and switching.

Path of a Packet in the Egress Direction

The following steps describe the path of an egress packet through the 1-Port OC-192c/STM-64 POS/RPR XFP SPA:

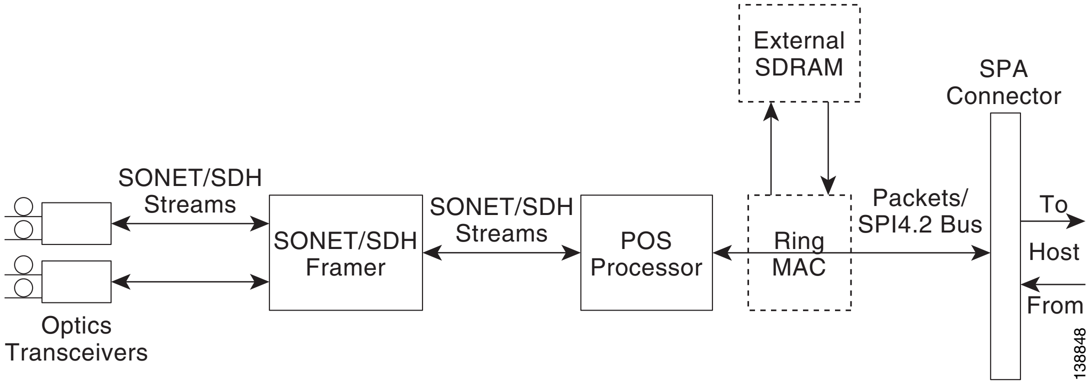
1. The host sends packets to the SPA using the SPI4.2 bus.
2. The SPA stores the data in the appropriate channel’s first-in first-out (FIFO) queue.
3. The SPA passes the packet to the framer.
4. The framer accepts the data and stores it in the appropriate channel queue.

5. The framer adds the FCS and SONET/SDH overhead.
6. The framer sends the data to the XFP optics for transmission onto the network.

2-Port OC-48c/STM-16 POS SPA Architecture

Figure 12-3 identifies the primary hardware devices that are part of the 2-Port OC-48c/STM-16 POS SPA architecture.

Figure 12-3 2-Port OC-48c/STM-16 POS SPA Architecture



Path of a Packet in the Ingress Direction

The following steps describe the path of an ingress packet through the 2-Port OC-48c/STM-16 POS SPA:

1. The framer receives SONET/SDH streams from the SFP optics, extracts clocking and data, and processes the section, line, and path overhead.
2. The framer detects Loss of Signal (LOS), Loss of Frame (LOF), Severely Errored Frame (SEF), Line Alarm Indication Signal (AIS-L), Loss of Pointer (LOP), Line Remote Defect Indication Signal (Enhanced RDI-L), Path Alarm Indication Signal (AIS-P), Standard and Enhanced Path Remote Defect Indication Signal (RDI-P), Path Remote Error Indication (Enhanced REI-P). The framer extracts or inserts DCC bytes.
3. The framer processes the S1 synchronization status byte, the pointer action bytes (per Telcordia GR-253-CORE), and extracts or inserts DCC bytes.
4. The POS processor extracts the POS frame payload and verifies the frame size and frame check sequence (FCS).
5. The POS processor supports PPP, Frame Relay, or HDLC modes and optionally performs payload scrambling.
6. The POS processor passes valid frames to the System Packet Level Interface 4.2 (SPI4.2) interface on the SPA.
7. The SPI4.2 interface transfers frames to the host through the SPI4.2 bus for further processing and switching.

Path of a Packet in the Egress Direction

The following steps describe the path of an egress packet through the 2-Port OC-48c/STM-16 POS SPA:

1. The host sends packets to the SPA using the SPI4.2 bus.
2. The SPA stores the data in the appropriate SPI4 channel's first-in first-out (FIFO) queue.
3. The SPA passes the packet from the SPI4 interface to the POS processor where it is encapsulated in a POS frame and FCS is added.
4. The POS frame is sent to the SONET/SDH framer where it is placed into the SONET payload.
5. The framer adds the FCS and SONET/SDH overhead.
6. The framer sends the data to the SFP optics for transmission onto the network.

Displaying the SPA Hardware Type

To verify the SPA hardware type that is installed in your Catalyst 6500 Series switch, you can use the **show idprom** command. For other hardware information, you can also use the **show interfaces** or **show controllers** commands. There are several other commands on the Catalyst 6500 Series switch that also provide SPA hardware information. For more information about these commands, see the "Command Summary for POS SPAs" and the "SIP and SPA Commands" chapters in this guide.

Table 12-2 shows the hardware description that appears in the **show** command output for each type of SPA that is supported on the Catalyst 6500 Series switch.

Table 12-2 SPA Hardware Descriptions in show Commands

SPA	Description in show interfaces Command	Description in show idprom Command
2-Port OC-3c/STM-1 POS SPA	Hardware is Packet over Sonet	2-port OC3/STM1 POS Shared Port Adapter / SPA-2XOC3-POS
4-Port OC-3c/STM-1 POS SPA	Hardware is Packet over Sonet	4-port OC3/STM1 POS Shared Port Adapter / SPA-4XOC3-POS
1-Port OC-12c/STM-4 POS SPA	Hardware is Packet over Sonet	1-port OC12/STM4 POS Shared Port Adapter / SPA-1XOC12-POS
1-Port OC-48c/STM-16 POS SPA	Hardware is Packet over Sonet	1-port OC48/STM16 POS/RPR Shared Port Adapter / SPA-1XOC48POS/RPR
2-Port OC-48c/STM-16 POS SPA	Hardware is Packet over Sonet	2-port OC48/STM16 POS/RPR Shared Port Adapter / SPA-2XOC48POS/RPR
4-Port OC-48c/STM-16 POS SPA	Hardware is Packet over Sonet	4-port OC48/STM16 POS/RPR Shared Port Adapter / SPA-4XOC48POS/RPR

Table 12-2 SPA Hardware Descriptions in show Commands (continued)

SPA	Description in show interfaces Command	Description in show idprom Command
1-Port OC-192c/STM-64 POS/RPR SPA	Hardware is Packet over Sonet	1-port OC192/STM64 POS/RPR Shared Port Adapter / SPA-OC192POS-VSR / SPA-OC192POS-LR
1-Port OC-192c/STM-64 POS/RPR XFP SPA	Hardware is Packet over Sonet	1-port OC192/STM64 POS/RPR XFP Optics Shared Port Adapter / SPA-OC192POS-XFP

Example of the show idprom Command

The following example shows sample output for the **show idprom module detail** command for a 4-Port OC-3c/STM-1 POS SPA installed in subslot 3 of the SIP installed in slot 2 of the router:

```
Router# show idprom module 2/3 detail
IDPROM for SPA module #2/3
        (FRU is '4-port OC3/STM1 POS Shared Port Adapter')
        EEPROM version            : 4
        Compatible Type           : 0xFF
        Controller Type           : 1088
        Hardware Revision         : 0.230
        Boot Timeout              : 0 msecs
        PCB Serial Number         : PRTA0304155
        Part Number               : 73-9313-02
        73/68 Board Revision      : 04
        Fab Version               : 02
        RMA Test History          : 00
        RMA Number                : 0-0-0-0
        RMA History               : 00
        Deviation Number          : 0
        Product Identifier (PID)  : SPA-4XOC3-POS
        Version Identifier (VID)  : V01
.
.
.
```

Example of the show interfaces Command

The following example shows output from the **show interfaces pos** command on a Catalyst 6500 Series switch with a 4-Port OC-3c/STM-1 POS SPA installed in slot 5:

```
Router# show interfaces pos 5/0/1
POS5/0/1 is up, line protocol is up
  Hardware is Packet over Sonet
  Internet address is 10.5.5.5/8
  MTU 4470 bytes, BW 155000 Kbit, DLY 100 usec,
     reliability 96/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive not set
  Scramble disabled
  Last input 00:00:11, output 00:00:11, output hang never
  Last clearing of ''show interface'' counters 00:00:23
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  5 packets input, 520 bytes
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  5 packets output, 520 bytes, 0 underruns
     0 output errors, 0 applique, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Example of the show controllers Command

The following example shows output from the **show controllers pos** command on a Catalyst 6500 Series switch for the first interface (0) of a POS SPA installed in subslot 2 of a SIP installed in chassis slot 3:

```
Router# show controllers pos 3/2/0
POS3/2/0
SECTION
LOF = 0 LOS = 0 BIP(B1) = 0
LINE
AIS = 0 RDI = 0 FEBE = 0 BIP(B2) = 0
PATH
AIS = 0 RDI = 0 FEBE = 0 BIP(B3) = 0
PLM = 0 UNEQ = 0 TIM = 0 TIU = 0
LOP = 0 NEWPTR = 0 PSE = 0 NSE = 0

Active Defects: None
Active Alarms: None
Alarm reporting enabled for: SF SLOS SLOF B1-TCA B2-TCA PLOP B3-TCA

Framing: SONET
APS

COAPS = 0 PSBF = 0
State: PSBF_state = False
Rx(K1/K2): 00/00 Tx(K1/K2): 00/00
Rx Synchronization Status S1 = 00
S1S0 = 00, C2 = CF
Remote aps status (none); Reflected local aps status (none)
CLOCK RECOVERY
RDOOL = 0
State: RDOOL_state = False
PATH TRACE BUFFER: STABLE
Remote hostname : sip-sw-7600-2
Remote interface: POS3/2/1
Remote IP addr : 0.0.0.0
Remote Rx(K1/K2): 00/00 Tx(K1/K2): 00/00

BER thresholds: SF = 10e-3 SD = 10e-6
TCA thresholds: B1 = 10e-6 B2 = 10e-6 B3 = 10e-6

Clock source: internal
```

CHAPTER 13

Configuring the POS SPAs

This chapter provides information about configuring the Packet over SONET (POS) shared port adapters (SPAs) on the Catalyst 6500 Series switch. This chapter includes the following sections:

- Configuration Tasks, page 13-1
- Verifying the Interface Configuration, page 13-18
- Configuration Examples, page 13-18

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Configuration Tasks

This section describes how to configure POS SPAs and includes information about verifying the configuration.

It includes the following topics:

- Required Configuration Tasks, page 13-2
- Specifying the Interface Address on a SPA, page 13-3
- Modifying the Interface MTU Size, page 13-4
- Modifying the POS Framing, page 13-5
- Modifying the Keepalive Interval, page 13-7
- Modifying the CRC Size, page 13-7
- Modifying the Clock Source, page 13-8
- Modifying SONET Payload Scrambling, page 13-9
- Configuring the Encapsulation Type, page 13-10
- Configuring APS, page 13-11
- Configuring POS Alarm Trigger Delays, page 13-12
- Configuring SDCC, page 13-15

- Configuring Layer 2 Internetworking Features on POS SPAs, page 13-16
- Saving the Configuration, page 13-17
- Shutting Down and Restarting an Interface on a SPA, page 13-18

Required Configuration Tasks

This section lists the required configuration steps to configure the POS SPAs. Some of the required configuration commands implement default values that might be appropriate for your network. If the default value is correct for your network, then you do not need to configure the command. These commands are indicated by "(As Required)" in the Purpose column.

To configure the POS SPAs, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface pos slot/subslot/port`	Specifies the POS interface to configure and enters interface configuration mode, where: • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 13-3.
Step 3	`Router(config-if)# ip address ip-address mask [secondary]`	Sets a primary or secondary IP address for an interface, where: • *ip-address*—Specifies the IP address for the interface. • *mask*—Specifies the mask for the associated IP subnet. • **secondary**—(Optional) Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 4	`Router(config-if)# pos framing {sonet | sdh}`	(As Required) Specifies the POS framing type, where: • **sonet**—Enables Synchronous Optical Network Framing for optical carrier (OC) rates. This is the default. • **sdh**—Enables Synchronous Digital Hierarchy framing for synchronous transfer mode (STM) rates. The POS framing type must be configured to be the same on both ends of the POS link.
Step 5	`Router(config-if)# mtu bytes`	(As Required) Configures the maximum transmission unit (or packet size) for an interface, where: • *bytes*—Specifies the maximum number of bytes for a packet. The default is 4470 bytes.

	Command	Purpose
Step 6	Router(config-if)# **keepalive** [*period* [*retries*]]	(As Required) Specifies the frequency at which the Cisco IOS software sends messages to the other end of the link, to ensure that a network interface is alive, where: • *period*—Specifies the time interval in seconds for sending keepalive packets. The default is 10 seconds. • *retries*—Specifies the number of times that the device will continue to send keepalive packets without response before bringing the interface down. The default is 5 retries. The keepalive must be configured to be the same on both ends of the POS link.
Step 7	Router(config-if)# **crc** [**16** | **32**]	(As Required) Specifies the length of the cyclic redundancy check (CRC), where: • **16**—Specifies a 16-bit length CRC. This is the default. • **32**—Specifies a 32-bit length CRC. The CRC size must be configured to be the same on both ends of the POS link.
Step 8	Router(config-if)# **clock source** {**line** | **internal**}	(As Required) Specifies the clock source for the POS link, where: • **line**—The link uses the recovered clock from the line. This is the default. • **internal**—The link uses the internal clock source.
Step 9	Router(config-if)# **encapsulation** *encapsulation-type*	(As Required) Specifies the encapsulation method used by the interface, where: • *encapsulation-type*—Can be HDLC, PPP, or Frame Relay. The default encapsulation is HDLC. The encapsulation must be configured to be the same on both ends of the POS link. Note: The POS SPAs on the Cisco 7600 SIP-600 do not support Frame Relay.
Step 10	Router(config-if)# **pos scramble-atm**	(As Required) Enables SONET payload scrambling. The default configuration is SONET payload scrambling disabled. The SONET payload scrambling must be configured to be the same on both ends of the POS link.
Step 11	Router(config-if)# **no shutdown**	Enables the interface.

Specifying the Interface Address on a SPA

SPA interface ports begin numbering with "0" from left to right. Single-port SPAs use only the port number 0. To configure or monitor SPA interfaces, you need to specify the physical location of the SIP, SPA, and interface in the CLI. The interface address format is *slot***/***subslot***/***port*, where:

- *slot*—Specifies the chassis slot number in the Catalyst 6500 Series switch where the SIP is installed.

- *subslot*—Specifies the secondary slot of the SIP where the SPA is installed.
- *port*—Specifies the number of the individual interface port on a SPA.

The following example shows how to specify the first interface (0) on a SPA installed in the first subslot of a SIP (0) installed in chassis slot 3:

```
Router(config)# interface serial 3/0/0
```

This command shows a serial SPA as a representative example, however the same *slot/subslot/port* format is similarly used for other SPAs (such as ATM and POS) and other non-channelized SPAs.

For more information about identifying slots and subslots, see the "Identifying Slots and Subslots for SIPs, SSCs, and SPAs" section on page 4-2.

Modifying the Interface MTU Size

The Cisco IOS software supports three different types of configurable maximum transmission unit (MTU) options at different levels of the protocol stack:

- Interface MTU—Checked by the SPA on traffic coming in from the network. Different interface types support different interface MTU sizes and defaults. The interface MTU defines the maximum packet size allowable (in bytes) for an interface before drops occur. If the frame is smaller than the interface MTU size, but is not smaller than three bytes of payload size, then the frame continues to process.
- IP MTU—Can be configured on a subinterface and is used by the Cisco IOS software to determine whether fragmentation of a packet takes place. If an IP packet exceeds the IP MTU size, then the packet is fragmented.
- Tag or Multiprotocol Label Switching (MPLS) MTU—Can be configured on a subinterface and allows up to six different labels, or tag headers, to be attached to a packet. The maximum number of labels is dependent on your Cisco IOS software release.

Different encapsulation methods and the number of MPLS MTU labels add additional overhead to a packet. For example, for an Ethernet packet, SNAP encapsulation adds an 8-byte header, dot1q encapsulation adds a 2-byte header, and each MPLS label adds a 4-byte header (*n* labels x 4 bytes).

Interface MTU Configuration Guidelines

When configuring the interface MTU size on the POS SPAs, consider the following guidelines:

- If you are also using MPLS, be sure that the **mpls mtu** command is configured for a value less than or equal to the interface MTU.
- If you change the interface MTU size, the giant counter increments when the interface receives a packet that exceeds the MTU size that you configured, plus an additional 88 bytes for overhead, and an additional 2 or 4 bytes for the configured cyclic redundancy check (CRC).

  For example, with a maximum MTU size of 9216 bytes, the giant counter increments:

  - For a 16-bit CRC (or FCS), when receiving packets larger than 9306 bytes (9216 + 88 + 2).
  - For a 32-bit CRC, when receiving packets larger than 9308 bytes (9216 + 88 + 4).

- The Frame Relay Local Management Interface (LMI) protocol requires that all permanent virtual circuit (PVC) status reports fit into a single packet. Using the default MTU of 4470 bytes, this limits the number of data-link connection identifiers (DLCIs) to 890. The following formula demonstrates how to determine the maximum DLCIs for a configured interface MTU:
  - Maximum DLCIs = (MTU bytes – 20)/(5 bytes per DLCI)
  - Maximum DLCIs for the default MTU = (4470 – 20)/5 = 890 DLCIs per interface

Interface MTU Configuration Task

To modify the MTU size on an interface, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# **mtu** *bytes*`	Configures the maximum packet size for an interface, where: • *bytes*—Specifies the maximum number of bytes for a packet. The default is 4470 bytes.

To return to the default MTU size, use the **no** form of the command.

Verifying the MTU Size

To verify the MTU size for an interface, use the **show interfaces pos** privileged EXEC command and observe the value shown in the MTU field.

The following example shows an MTU size of 4470 bytes for interface port 0 (the first port) on the SPA installed in subslot 1 of the SIP that is located in slot 2 of the Catalyst 6500 Series switch:

```
Router# show interfaces pos 2/1/0
POS2/1/0 is up, line protocol is up (APS working - active)
  Hardware is Packet over Sonet
  Internet address is 10.1.1.1/24
  MTU 4470 bytes, BW 155000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255.
.
.
.
```

Modifying the POS Framing

POS framing can be specified as SONET (Synchronous Optical Network) or SDH (Synchronous Digital Hierarchy). SONET and SDH are a set of related standards for synchronous data transmission over fiber-optic networks. SONET is the United States version of the standard published by the American National Standards Institute (ANSI). SDH is the international version of the standard published by the International Telecommunications Union (ITU).

To modify the POS framing, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# pos framing {sonet | sdh}`	Specifies the POS framing type, where: • **sonet**—Enables Synchronous Optical Network framing for optical carrier (OC) rates. This is the default. • **sdh**—Enables Synchronous Digital Hierarchy framing for synchronous transfer mode (STM) rates. The POS framing type must be configured to be the same on both ends of the POS link.

To return to the default, use the **no** form of the command.

Verifying the POS Framing

To verify the POS framing, use the **show controllers pos** privileged EXEC command and observe the value shown in the Framing field. The following example shows that POS framing mode is set to SONET for the first interface (0) on the POS SPA installed in subslot 2 of a SIP installed in chassis slot 3:

```
Router# show controllers pos 3/2/0
POS3/2/0
SECTION
LOF = 0 LOS = 0 BIP(B1) = 0
LINE
AIS = 0 RDI = 0 FEBE = 0 BIP(B2) = 0
PATH
AIS = 0 RDI = 0 FEBE = 0 BIP(B3) = 0
PLM = 0 UNEQ = 0 TIM = 0 TIU = 0
LOP = 0 NEWPTR = 0 PSE = 0 NSE = 0

Active Defects: None
Active Alarms: None
Alarm reporting enabled for: SF SLOS SLOF B1-TCA B2-TCA PLOP B3-TCA

Framing: SONET
APS

COAPS = 0 PSBF = 0
State: PSBF_state = False
Rx(K1/K2): 00/00 Tx(K1/K2): 00/00
Rx Synchronization Status S1 = 00
S1S0 = 00, C2 = CF
Remote aps status (none); Reflected local aps status (none)
CLOCK RECOVERY
RDOOL = 0
State: RDOOL_state = False
PATH TRACE BUFFER: STABLE
Remote hostname : sip-sw-7600-2
Remote interface: POS3/2/1
Remote IP addr : 0.0.0.0
Remote Rx(K1/K2): 00/00 Tx(K1/K2): 00/00

BER thresholds: SF = 10e-3 SD = 10e-6
TCA thresholds: B1 = 10e-6 B2 = 10e-6 B3 = 10e-6
```

```
Clock source: internal
```

Modifying the Keepalive Interval

When the keepalive feature is enabled, a keepalive packet is sent at the specified time interval to keep the interface active. The keepalive interval must be configured to be the same on both ends of the POS link.

To modify the keepalive interval, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# **keepalive** [*period* [*retries*]]`	Specifies the frequency at which the Cisco IOS software sends messages to the other end of the link, to ensure that a network interface is alive, where: - *period*—Specifies the time interval in seconds for sending keepalive packets. The default is 10 seconds. - *retries*—Specifies the number of times that the device will continue to send keepalive packets without response before bringing the interface down. The default is 5 retries.

To disable keepalive packets, use the **no** form of this command.

Verifying the Keepalive Interval

To verify the keepalive interval, use the **show interfaces pos** privileged EXEC command and observe the value shown in the Keepalive field.

The following example shows that keepalive is enabled for interface port 0 on the POS SPA installed in the SIP that is located in slot 2 of the Catalyst 6500 Series switch:

```
Router# show interfaces pos 2/0/0
    Hardware is Packet over Sonet
    Internet address is 10.1.1.1.2
    MTU 9216 bytes, BW 622000 Kbit, DLY 100 usec, reliability 255/255, txload 1/255,
    rxload 1/255
      Keepalive set (10 sec)
.
.
.
```

Modifying the CRC Size

CRC is an error-checking technique that uses a calculated numeric value to detect errors in transmitted data. The CRC size indicates the length in bits of the FCS.

The CRC size must be configured to be the same on both ends of the POS link.

To modify the CRC size, perform this task in interface configuration mode:

Command	Purpose
Router(config-if)# **crc** [**16** | **32**]	(As Required) Specifies the length of the cyclic redundancy check (CRC), where: • **16**—Specifies a 16-bit length CRC. This is the default. • **32**—Specifies a 32-bit length CRC. The CRC size must be configured to be the same on both ends of the POS link.

To return to the default CRC size, use the **no** form of the command.

Verifying the CRC Size

To verify the CRC size, use the **show interfaces pos** privileged EXEC command and observe the value shown in the CRC field.

The following example shows that the CRC size is 16 for interface port 0 on the POS SPA installed in the SIP that is located in slot 2 of the Catalyst 6500 Series switch:

```
Router# show interfaces pos 2/0/0
    Hardware is Packet over Sonet
    Internet address is 10.1.1.2.1
    MTU 9216 bytes, BW 622000 Kbit, DLY 100 usec reliability 255/255, txload 1/255, rxload
    1/255
        Encapsulation HDLC, crc 16, loopback not set
.
.
.
```

Modifying the Clock Source

A clock source of internal specifies that the interface clocks its transmitted data from its internal clock. A clock source of line specifies that the interface clocks its transmitted data from a clock recovered from the line’s receive data stream.

For information about the recommended clock source settings for POS switch interfaces, refer to *Configuring Clock Settings on POS Router Interfaces* at the following URL:

http://www.cisco.com/en/US/tech/tk482/tk607/technologies_tech_note09186a0080094bb9.shtml

To modify the clock source, perform this task in interface configuration mode:

Command	Purpose
Router(config-if)# **clock source** {**line** | **internal**}	Specifies the clock source for the POS link, where: • **line**—The link uses the recovered clock from the line. This is the default. • **internal**—The link uses the internal clock source.

To return to the default clock source, use the **no** form of this command.

Verifying the Clock Source

To verify the clock source, use the **show controllers pos** privileged EXEC command and observe the value shown in the Clock source field.

The following example shows that the clock source is internal for interface port 0 on the POS SPA installed in subslot 0 of the SIP that is located in slot 2 of the Catalyst 6500 Series switch:

```
Router# show controllers pos 2/0/0
POS2/0/0
SECTION
LOF = 0 LOS = 1 BIP(B1) = 7
LINE
AIS = 0 RDI = 1 FEBE = 20 BIP(B2) = 9
PATH
AIS = 0 RDI = 0 FEBE = 0 BIP(B3) = 5
PLM = 0 UNEQ = 0 TIM = 0 TIU = 0
LOP = 0 NEWPTR = 0 PSE = 0 NSE = 0


Active Defects: None
Active Alarms: None

Alarm reporting enabled for: SF SLOS SLOF B1-TCA LAIS LRDI B2-TCA PAIS PLOP PRDI PUNEQ
B3-TCA RDOOL

APS

COAPS = 2 PSBF = 0
State: PSBF_state = False
Rx(K1/K2): 00/00 Tx(K1/K2): 00/00
Rx Synchronization Status S1 = 00
S1S0 = 02, C2 = CF
CLOCK RECOVERY
RDOOL = 0
State: RDOOL_state = False
PATH TRACE BUFFER: STABLE
Remote hostname : RouterTester. Port 102/1
Remote interface:
Remote IP addr :
Remote Rx(K1/K2): / Tx(K1/K2): /

BER thresholds: SF = 10e-5 SD = 10e-6
TCA thresholds: B1 = 10e-6 B2 = 10e-6 B3 = 10e-6

Clock source: internal
.
.
.
```

Modifying SONET Payload Scrambling

SONET payload scrambling applies a self-synchronous scrambler (x43+1) to the Synchronous Payload Envelope (SPE) of the interface to ensure sufficient bit transition density.

The default configuration is SONET payload scrambling disabled.

SONET payload scrambling must be configured to be the same on both ends of the POS link.

To modify SONET payload scrambling, perform this task in interface configuration mode:

Command	Purpose
Router(config-if)# **pos scramble-atm**	Enables SONET payload scrambling.

To disable SONET payload scrambling, use the **no** form of this command.

Verifying SONET Payload Scrambling

To verify SONET payload scrambling, use the **show interfaces pos** privileged EXEC command and observe the value shown in the Scramble field.

The following example shows that SONET payload scrambling is disabled for interface port 0 on the POS SPA installed in subslot 0 of the SIP that is located in slot 2 of the Catalyst 6500 Series switch:

```
Router# show interfaces pos 2/0/0
    Hardware is Packet over Sonet
    Internet address is 10.0.0.1/24
    MTU 9216 bytes, BW 622000 Kbit, DLY 100 usec,
        reliability 255/255, txload 1/255, rxload 1/255
        Encapsulation HDLC, crc 16, loopback not set
    Keepalive not set
    Scramble disabled
.
.
.
```

Configuring the Encapsulation Type

By default, the POS interfaces support High-Level Data Link Control (HDLC) encapsulation. The encapsulation method can be specified as HDLC, Point-to-Point Protocol (PPP) or Frame Relay. The encapsulation type must be configured to be the same on both ends of the POS link.

To modify the encapsulation method, perform this task in interface configuration mode:

Command	Purpose
Router(config-if)# **encapsulation** *encapsulation-type*	Specifies the encapsulation method used by the interface, where: • *encapsulation-type*—Can be HDLC, PPP, or Frame Relay. The default is HDLC.

Verifying the Encapsulation Type

To verify the encapsulation type, use the **show interfaces pos** privileged EXEC command and observe the value shown in the Encapsulation field.

The following example shows the encapsulation type is HDLC for port 0 on the POS SPA installed in subslot 0 of the SIP that is located in slot 2 of the Catalyst 6500 Series switch:

```
Router# show interfaces pos 2/0/0
  Hardware is Packet over Sonet
  Internet address is 10.0.0.1/24
  MTU 9216 bytes, BW 622000 Kbit, DLY 100 usec,
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
    Keepalive not set
    Scramble disabled
.
.
.
```

Configuring APS

Automatic protection switching (APS) allows switchover of POS circuits in the event of circuit failure and is often required when connecting SONET equipment to telco equipment. APS refers to the method of using a protect POS interface in the SONET network as the backup for a working POS interface. When the working interface fails, the protect interface quickly assumes its traffic load. Depending on the configuration, the two circuits may be terminated in the same switch, or in different switches.

For more information about APS, refer to *A Brief Overview of Packet Over SONET APS* at the following URL:

http://www.cisco.com/en/US/tech/tk482/tk607/technologies_tech_note09186a0080093eb5.shtml

To configure the working POS interface, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# ` **`aps working`** *`circuit-number`*	Configures a POS interface as a working APS interface, where: • *circuit-number*—Specifies the circuit number associated with this working interface.

To remove the POS interface as a working interface, use the **no** form of this command.

To configure the protect POS interface, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# ` **`aps protect`** *`circuit-number ip-address`*	Configures a POS interface as a protect APS interface, where: • *circuit-number*—Specifies the number of the circuit to enable as a protect interface. • *ip-address*—Specifies the IP address of the switch that has the working POS interface.

To remove the POS interface as a protect interface, use the **no** form of this command.

Verifying the APS Configuration

To verify the APS configuration or to determine if a switchover has occurred, use the **show aps** command.

The following is an example of a switch configured with a working interface. In this example, POS interface 0/0/0 is configured as a working interface in group 1, and the interface is selected (that is, active).

```
Router# show aps
POS0/0/0 working group 1 channel 1 Enabled Selected
```

The following is an example of a switch configured with a protect interface. In this example, POS interface 2/1/1 is configured as a protect interface in group 1. The output also shows that the working channel is located on the switch with the IP address 10.0.0.1 and that the interface currently selected is enabled.

```
Router# show aps
POS2/1/1 APS Group 1: protect channel 0 (inactive)
Working channel 1 at 10.0.0.1 (Enabled)
  SONET framing; SONET APS signalling by default
  Remote APS configuration: (null)
.
.
.
```

Configuring POS Alarm Trigger Delays

A trigger is an alarm that when activated causes the line protocol to go down. The POS alarm trigger delay helps to ensure uptime of a POS interface by preventing intermittent problems from disabling the line protocol. The POS alarm trigger delay feature delays the setting of the line protocol to down when trigger alarms are received. If the trigger alarm was sent because of an intermittent problem, the POS alarm trigger delay can prevent the line protocol from going down when the line protocol is functional.

Line-Level and Section-Level Triggers

The **pos delay triggers line** command is used for POS switch interfaces connected to internally protected Dense Wavelength Division Multiplexing (DWDM) systems. This command is invalid for interfaces that are configured as working or protect APS. Normally, a few microseconds of line- or section-level alarms brings down the link until the alarm has been clear for ten seconds. If you configure holdoff, the link-down trigger is delayed for 100 milliseconds. If the alarm stays up for more than 100 milliseconds, the link is brought down. If the alarm clears before 100 milliseconds, the link remains up.

The following line- and section-level alarms are triggers, by default, for the line protocol to go down:

- Line alarm indication signal (LAIS)
- Section loss of signal (SLOS)
- Section loss of frame (SLOF)

You can use the **pos delay triggers line** command to delay a down trigger of the line protocol on the interface. You can set the delay from 50 to 10000 milliseconds. The default delay is 100 milliseconds.

To configure POS line- or section-level triggers, perform this task beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# pos delay triggers line ms`	Specifies a delay for setting the line protocol to down when a line-level trigger alarm is received, where: • *ms*—Specifies the delay in milliseconds. The default delay is 100 milliseconds.
Step 2	`Router(config-if)# pos threshold {b1-tca | b2-tca | b3-tca | sd-ber | sf-ber} rate`	Configures the POS bit error rate (BER) threshold values of the specified alarms, where: • **b1-tca** *rate*—Specifies the B1 BER threshold crossing alarm. The default is 6. • **b2-tca** *rate*—Specifies the B2 BER threshold crossing alarm. The default is 6. • **b3-tca** *rate*—Specifies the B3 BER threshold crossing alarm. The default is 6. • **sd-ber** *rate*—Specifies the signal degrade BER threshold. The default is 6. • **sf-ber** *rate*—Specifies the signal failure BER threshold. The default is 3. • *rate*—Specifies the bit error rate from 3 to 9 (10e-n). The default varies by the type of threshold that you configure.
Step 3	`Router(config-if)# pos ais-shut`	Sends a line alarm indication signal (AIS-L) to the other end of the link after a **shutdown** command has been issued to the specified POS interface. AIS-L is also known as LAIS when alarm-related output is generated using the **show controllers pos** command. By default, the AIS-L is not sent to the other end of the link. Stops transmitting the AIS-L by issuing either the **no shutdown** or the **no pos ais-shut** commands.

To disable alarm trigger delays, use the **no** form of the **pos delay triggers line** command.

To determine which alarms are reported on the POS interface, and to display the BER thresholds, use the **show controllers pos** command.

Path-Level Triggers

To configure various path alarms as triggers and to specify an activation delay between 50 and 10000 milliseconds, use the **pos delay triggers path** command. The default delay value is 100 milliseconds. The following path alarms are not triggers by default. You can configure these path alarms as triggers and also specify a delay:

- Path alarm indication signal (PAIS)
- Path remote defect indication (PRDI)
- Path loss of pointer (PLOP)

- sd-ber (signal degrade [SD] bit error rate [BER])
- sf-ber (signal failure [SF] BER)
- b1-tca (B1 BER threshold crossing alarm [TCA])
- b2-tca (B2 BER TCA)
- b3-tca (B3 BER TCA)

The **pos delay triggers path** command can also bring down the line protocol when the higher of the B2 and B3 error rates is compared with the signal failure (SF) threshold. If the SF threshold is crossed, the line protocol of the interface goes down.

To configure POS path-level triggers, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# pos delay triggers path ms`	Specifies that path-level alarms should act as triggers and specifies a delay for setting the line protocol to down when a path-level trigger alarm is received, where: • *ms*—Specifies the delay in milliseconds. The default delay is 100 milliseconds.

To disable path-level triggers, use the **no** form of this command.

Verifying POS Alarm Trigger Delays

To verify POS alarm trigger delays, use the **show controllers pos** privileged EXEC command and observe the values shown in the Line alarm trigger delay and Path alarm trigger delay fields.

The following example shows the POS alarm trigger delays for interface port 0 on the POS SPA installed in the SIP that is located in slot 2 of the Catalyst 6500 Series switch:

```
Router# show controllers pos 2/0/0 details
POS2/0/0
SECTION
LOF = 0 LOS = 1 BIP(B1) = 5
LINE
AIS = 0 RDI = 1 FEBE = 5790 BIP(B2) = 945
PATH
AIS = 0 RDI = 0 FEBE = 0 BIP(B3) = 5
PLM = 0 UNEQ = 0 TIM = 0 TIU = 0
LOP = 1 NEWPTR = 0 PSE = 0 NSE = 0

Active Defects: None
Active Alarms: None
Alarm reporting enabled for: SF SLOS SLOF B1-TCA B2-TCA PLOP B3-TCA

Line alarm trigger delay = 100 ms
Path alarm trigger delay = 100 ms
.
.
.
```

Configuring SDCC

Before any management traffic can traverse the section data communication channel (SDCC) links embedded in the POS SPA overhead, the SDCC interfaces must be configured and activated.

Note SDCC is not supported by the 1-Port OC-48c/STM-16 POS SPA on the Cisco 7600 SIP-400 and is not supported by any POS SPAs on the Cisco 7600 SIP-600.

SDCC Configuration Guidelines

When configuring SDCC on a POS SPA, consider the following guidelines:

- SDCC must be enabled on the main POS interfaces.
- SDCC can be configured on up to two interfaces of the 4-Port OC-3c/STM-1 POS SPA.
- SDCC supports only HDLC and PPP encapsulation, not Frame Relay.

SDCC Configuration Task

To configure the POS SPAs for SDCC, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface pos`** *`slot/subslot/port`*	Specifies the POS interface to configure and enters interface configuration mode, where: • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 13-3.
Step 3	`Router(config-if)#` **`sdcc enable`**	Enables SDCC on the interface.
Step 4	`Router(config-if)#` **`exit`**	Exits interface configuration mode and returns to global configuration mode.
Step 5	`Router(config)#` **`interface sdcc`** *`slot/subslot/port`*	Specifies the SDCC interface and enters interface configuration mode, where: • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 13-3.

	Command	Purpose
Step 6	`Router(config-if)# ip address ip-address mask [secondary]`	Sets a primary or secondary IP address for an interface, where: - *ip-address*—Specifies the IP address for the interface. - *mask*—Specifies the mask for the associated IP subnet. - **secondary**—(Optional) Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 7	`Router(config-if)# no shutdown`	Enables the interface.

Verifying the SDCC Interface Configuration

To verify the SDCC interface, use the **show interfaces sdcc** privileged EXEC command and observe the value shown in the Hardware is field.

The following example shows the SDCC interface port 1 on the POS SPA installed in subslot 0 of the SIP that is located in slot 5 of the Catalyst 6500 Series switch:

```
Router# show interfaces sdcc 5/0/1
SDCC5/0/1 is up, line protocol is up
  Hardware is SDCC
  Internet address is 10.14.14.14/8
  MTU 1500 bytes, BW 155000 Kbit, DLY 20000 usec,
     reliability 5/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive not set
  Last input 00:01:24, output never, output hang never
  Last clearing of ''show interface'' counters 00:01:30
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  5 packets input, 520 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  5 packets output, 520 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Configuring Layer 2 Internetworking Features on POS SPAs

This section provides information about the Layer 2 internetworking features that are supported by the POS SPAs on the Catalyst 6500 Series switch.

Configuring Multipoint Bridging

Multipoint bridging (MPB) enables the connection of multiple ATM PVCs, Frame Relay PVCs, BCP ports, and WAN Gigabit Ethernet subinterfaces into a single broadcast domain (virtual LAN), together with the LAN ports on that VLAN. This feature enables service providers to add support for Ethernet-based Layer 2 services to the proven technology of their existing ATM and Frame Relay legacy networks. Customers can then use their current VLAN-based networks over the ATM or Frame Relay cloud. This feature also allows service providers to gradually update their core networks to the latest Gigabit Ethernet optical technologies, while still supporting their existing customer base.

For MPB configuration guidelines and restrictions and feature compatibility tables, see the "Configuring Multipoint Bridging" section on page 4-17 of Chapter 4, "Configuring the SIPs and SSC."

Configuring the Bridging Control Protocol

The Bridging Control Protocol (BCP) enables forwarding of Ethernet frames over SONET networks and provides a high-speed extension of enterprise LAN backbone traffic through a metropolitan area. The implementation of BCP on the SPAs includes support for IEEE 802.1D, IEEE 802.1Q Virtual LAN (VLAN), and high-speed switched LANs.

For BCP configuration guidelines and restrictions and feature compatibility tables, see the "Configuring PPP Bridging Control Protocol Support" section on page 4-18 of Chapter 4, "Configuring the SIPs and SSC."

Configuring Virtual Private LAN Service (VPLS) and Hierarchical VPLS

VPLS enables geographically separate LAN segments to be interconnected as a single bridged domain over a packet switched network, such as IP, MPLS or hybrid of both bridging techniques.

VPLS with EoMPLS uses an MPLS-based provider core, where the PE routers have to cooperate to forward customer Ethernet traffic for a given VPLS instance in the core. VPLS uses the provider core to join multiple attachment circuits together to simulate a virtual bridge that connects the multiple attachment circuits together. From a customer point of view, there is no topology for VPLS. All of the CE devices appear to connect to a logical bridge emulated by the provider core.

For VPLS and H-VPLS configuration guidelines and restrictions and feature compatibility tables, see the Configuring Virtual Private LAN Service (VPLS), page 4-23 of Chapter 4, "Configuring the SIPs and SSC."

Saving the Configuration

To save your running configuration to nonvolatile random-access memory (NVRAM), perform this task in privileged EXEC configuration mode:

Command	Purpose
`Router# copy running-config startup-config`	Writes the new configuration to NVRAM.

For more information about managing configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

Shutting Down and Restarting an Interface on a SPA

You can shut down and restart any of the interface ports on a SPA independently of each other. Shutting down an interface stops traffic and then enters the interface into an administratively down state.

If you are preparing for an OIR of a SPA, it is not necessary to independently shut down each of the interfaces prior to deactivation of the SPA. You do not need to independently restart any interfaces on a SPA after OIR of a SPA or SIP. For more information about performing an OIR for a SPA, see the "Preparing for Online Insertion and Removal of SIPs, SSCs, and SPAs" section on page 5-3.

To shut down an interface on a SPA, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# shutdown`	Disables an interface.

To restart an interface on a SPA, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# no shutdown`	Restarts a disabled interface.

Verifying the Interface Configuration

Besides using the **show running-configuration** command to display your Catalyst 6500 Series switch configuration settings, you can use the **show interfaces pos** and **show controllers pos** commands to get detailed information on a per-port basis for your POS SPAs.

Verifying Per-Port Interface Status

To find detailed interface information on a per-port basis for the POS SPAs, use the **show interfaces pos** command. For a description of the command output, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

Monitoring Per-Port Interface Statistics

To find detailed alarm and error information on a per-port basis for the POS SPAs, use the **show controllers pos** command. For a description of the command output, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

Configuration Examples

This section includes the following examples for configuring a POS SPA installed in a Catalyst 6500 Series switch:

- Basic Interface Configuration Example, page 13-19
- MTU Configuration Example, page 13-19
- POS Framing Configuration Example, page 13-20

- Keepalive Configuration Example, page 13-20
- CRC Configuration Example, page 13-20
- Clock Source Configuration Example, page 13-21
- SONET Payload Scrambling Configuration Example, page 13-21
- Encapsulation Configuration Example, page 13-21
- APS Configuration Example, page 13-21
- POS Alarm Trigger Delays Configuration Example, page 13-23
- SDCC Configuration Example, page 13-23

Basic Interface Configuration Example

The following example shows how to enter global configuration mode to enter global configuration mode to specify the interface that you want to configure, configure an IP address for the interface, enable the interface, and save the configuration. This example configures interface port 0 (the first port) of the SPA located in subslot 0 of the SIP that is installed in slot 2 of the Catalyst 6500 Series switch:

```
!Enter global configuration mode
!
Router# configure terminal
!
! Specify the interface address
!
Router(config)# interface pos 2/0/0
!
! Configure an IP address
!
Router(config-if)# ip address 192.168.50.1 192.255.255.0
!
! Enable the interface
!
Router(config-if)# no shutdown
!
! Save the configuration to NVRAM
!
Router(config-if)# exit
Router# copy running-config startup-config
```

MTU Configuration Example

The following example sets the MTU to 4470 bytes on interface port 1 (the second port) of the SPA located in the bottom subslot (1) of the SIP that is installed in slot 2 of the Catalyst 6500 Series switch:

```
!Enter global configuration mode
!
Router# configure terminal
!
! Specify the interface address
!
Router(config)# interface pos 2/1/1
!
! Configure MTU
!
Router(config-if)# mtu 4470
```

POS Framing Configuration Example

The following example shows how to change from the default POS framing of SONET to SDH:

```
!Enter global configuration mode
!
Router# configure terminal
!
! Specify the interface address
!
Router(config)# interface pos 2/1/1
! (The default pos framing is sonet)
!
!Modify the framing type
!
Router(config-if)# pos framing sdh
```

Keepalive Configuration Example

The following example shows how to change from the default keepalive period of 10 seconds to 20 seconds:

```
!Enter global configuration mode
!
Router# configure terminal
!
! Specify the interface address
!
Router(config)# interface pos 2/1/1
!
! Configure keepalive 20
!
Router(config-if)# keepalive 20
```

CRC Configuration Example

The following example shows how to change the CRC size from 32 bits to the default 16 bits for POS SPAs:

```
!Enter global configuration mode
!
Router# configure terminal
!
! Specify the interface address
!
Router(config)# interface pos 2/1/1
!
! Configure crc 16
!
Router(config-if)# crc 16
```

Clock Source Configuration Example

The following example shows how to change from the default clock source of internal to line:

```
!Enter global configuration mode
!
Router# configure terminal
!
! Specify the interface address
!
Router(config)# interface pos 2/1/1
!
! Configure the clock source
!
Router(config-if)# clock source line
```

SONET Payload Scrambling Configuration Example

The following example shows how to change from a default SONET payload scrambling of disabled to enabled:

```
!Enter global configuration mode
!
Router# configure terminal
!
! Specify the interface address
!
Router(config)# interface pos 2/1/1
!
! Configure the SONET payload scrambling
!
Router(config-if)# pos scramble-atm
```

Encapsulation Configuration Example

The following example shows how to change from the default encapsulation method of HDLC to PPP:

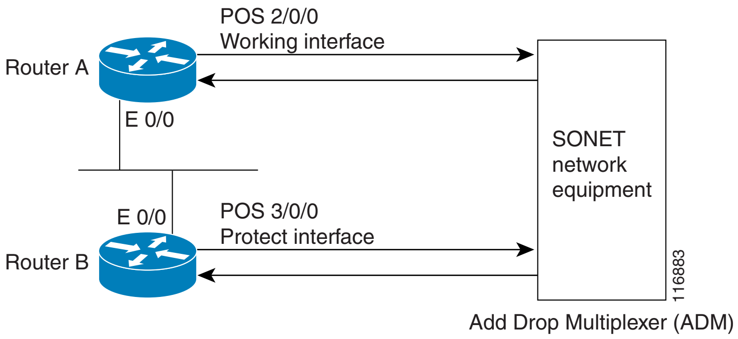
```
!Enter global configuration mode
!
Router# configure terminal

! Specify the interface address

Router(config)# interface pos 2/1/1
!
! Configure ppp
!
Router(config-if)# encapsulation ppp
```

APS Configuration Example

The following example shows the configuration of APS on router A and router B (Figure 13-1), and how to configure more than one protect or working interface on a router by using the **aps group** command.

Note In the following figure, the devices with the ATM SPAs are shown as Cisco 7600 series routers, but they can also be Catalyst 6500 series switches.

Figure 13-1 Basic APS Configuration



In this example, router A is configured with the working interface and router B is configured with the protect interface. If the working interface on router A becomes unavailable, the connection will automatically switch over to the protect interface on router B. The loopback interface is used as the interconnect. The **aps group** command is used even when a single protect group is configured.

The following example shows how to configure Router A for this scenario:

```
!Enter global configuration mode
!
Router# configure terminal
!
! Configure a loopback interface as the protect interconnect path
!
Router(config)# interface loopback 1
Router(config-if)# ip address 10.10.10.10 255.0.0.0

! Configure the POS interface address for the APS working interface
!
Router(config)# interface pos 2/0/0
!
! Configure the POS interface IP address and other interface parameters
!
Router(config-if)# ip address 172.16.1.8 255.255.0.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# no keepalive
Router(config-if)# crc 32
!
! Configure the APS group number by which to associate APS interfaces
!
Router(config-if)# aps group 1
!
! Configure a circuit number for the APS working interface
!
Router(config-if)# aps working 1
```

The following example shows how to configure Router B for this scenario:

```
!Enter global configuration mode
!
Router# configure terminal
!
```

```
! Configure the POS interface address for the APS protect interface
!
Router(config)# interface pos 3/0/0
!
! Configure the POS interface IP address and other interface parameters
!
Router(config-if)# ip address 172.16.1.9 255.255.0.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# no keepalive
Router(config-if)# crc 32
!
! Configure the APS group number by which to associate APS interfaces
!
Router(config-if)# aps group 1
!
! Configure a circuit number for the protect interface and an IP address for the router
! that has the APS working interface. In this case, the loopback interface address is
! used.
!
Router(config-if)# aps protect 1 10.10.10.10
```

POS Alarm Trigger Delays Configuration Example

The following example shows how to change POS line-level and path-level alarm trigger delays from the default of 100 milliseconds to 200 milliseconds:

```
!Enter global configuration mode
!
Router# configure terminal
!
! Specify the interface address
!
Router(config)# interface pos 2/1/1
!
Router(config-if)# pos delay triggers line 200
Router(config-if)# pos delay triggers path 200
```

SDCC Configuration Example

Note SDCC is not supported by the 1-Port OC-48c/STM-16 POS SPA on the Cisco 7600 SIP-400 and is not supported by any POS SPAs on the Cisco 7600 SIP-600.

The following example shows how to configure an SDCC interface:

```
!Enter global configuration mode
!
Router# configure terminal
!
! Specify the POS interface
!
Router(config)# interface pos 5/0/1
!
! Enable SDCC on the POS interface
!
Router(config-if)# sdcc enable
!
! Exit interface configuration mode and return to
! global configuration mode
!
Router(config-if# exit
!
! Specify the SDCC interface
!
Router(config)# interface sdcc 5/0/1
!
! Specify the IP address
!
Router(config-if)# ip address 10.14.14.14. 255.0.0.0
!
! Enable the interface
!
Router(config-if)# no shutdown
```

PART 6

Serial Shared Port Adapters

CHAPTER 14

Overview of the Serial SPAs

This chapter provides an overview of the release history, and feature and MIB support for the Cisco 7600 SIP-200 with the 2 and 4-Port T3/E3 SPAs, the 8-Port Channelized T1/E1 SPA, and the 2 or 4-Port CT3 SPA.

This chapter includes the following sections:

Release History

Release	Modification
Cisco IOS Release 12.2(33)SXI	Support for the following hardware was introduced on the Cisco 7600 SIP-400 on the Catalyst 6500 Series switch: • 2-Port and 4-Port Clear Channel T3/E3 SPA • 2-Port and 4-Port Channelized T3 SPA • 8-Port Channelized T1/E1 SPA

Cisco IOS Release 12.2(33)SXH	Support for the following hardware was introduced on the Cisco 7600 SIP-200 on the Catalyst 6500 Series switch: • 1-Port Channelized OC-3/STM-1 SPA Support for the following hardware was introduced on the Cisco 7600 SIP-400 on the Catalyst 6500 Series switch: • 2-Port Channelized T3 SPA
Cisco IOS Release 12.2(18)SXE	Support for the following hardware was introduced on the Cisco 7600 SIP-200 on the Cisco 7600 series router and Catalyst 6500 series switch: • 2-Port T3/E3 SPA (SPA-2XT3/E3) • 4-Port T3/E3 SPA (SPA-4XT3/E3) • 8-Port T1/E1 SPA (SPA-8XCHT1/E1) • 2-Port CT3 SPA (SPA-2XCT3/DS0) • 4-Port CT3 SPA (SPA-4XCT3/DS0)

Supported Features

This section provides a list of some of the primary features supported by the SIP and SPA hardware and software.

Cisco 7600 SIP-200 and Cisco 7600 SIP-400 Features

The SIPs are carrier cards designed to process packets between different SPAs and the Catalyst 6500 Series switch switching fabric.

- Online insertion and removal (OIR)
- Supports up to four single-height or two double-height Shared Port Adaptors (SPAs).
- Field Programmable Gate Array (FPGA) upgrade support

  The SIPs support the standard FPD upgrade methods for the Catalyst 6500 Series switch. For more information about FPD support, see Chapter 31, “Upgrading Field-Programmable Devices.”

Restrictions

Note For other SIP-specific features and restrictions see also Chapter 3, “Overview of the SIPs and SSC”in this guide.

- On a 2-port or 4-port Channelized T3 SPA, when one of the T3 ports is configured as a DS3 Clear Channel interface and the other T3s are configured with a large number (greater than or equal to 400) of low-bandwidth channels (NxDS0, N=1,2,3 or 4), the DS3 Clear Channel interface is not able to run at 100 percent DS3 line rate when those low-bandwidth channels are idle (not transmitting or receiving packets). This issue does not occur if those low-bandwidth channels are not idle.
- On a 2-Port and 4-Port Channelized T3 SPA or 1-Port Channelized OC-3/STM-1 SPA, the maximum number of channels is limited to 1023 per SPA.

- On a 2-Port and 4-Port Channelized T3 SPA or 1-Port Channelized OC-3/STM-1 SPA, the maximum number of FIFO buffers is 4096. The FIFO buffers are shared among the interfaces; how they are shared is determined by speed. If all the FIFO buffers have been assigned to existing interfaces, a new interface cannot be created, and the "%Insufficient FIFOs to create channel group" error message is seen. Table 14-1 provides FIFO allocation information.

  To find the number of available FIFO buffers, use the **show controller t3** command:

```
Router# show controller t3 3/0/0

T3 3/0/0 is up.
  Hardware is SPA-4XCT3/DS0
  IO FPGA version: 2.6, HDLC Framer version: 0
  T3/T1 Framer(1) version: 2, T3/T1 Framer(2) version: 2
  SUBRATE FPGA version: 1.4
  HDLC controller available FIFO buffers 3112
```

Table 14-1 FIFO Allocation

Number of Timeslots	Number of FIFO Buffers
1-6 DS0	4
7-8 DS0	6
9 DS0	6
10-12 DS0	8
13-23 DS0	12
1-6 E1 TS	4
7-9 E1 TS	6
11-16 E1 TS	8
17-31E1 TS	16
T1	12
E1	16
DS3	336

SPA Features

The following is a list of some of the significant software features supported by the 2 and 4-Port T3/E3 SPA, the 8-Port Channelized T1/E1 SPA, the 1-Port Channelized OC-3/STM-1 SPA, and the 2 and 4-Port CT3 SPAs.

- Software selectable between T1, E1, E3, or T3 framing on each card (ports are configured as all T1, E1, T3, or E3). Applies to the 2 and 4-Port T3/E3 SPA and 8-Port Channelized T1/E1 SPA.
- Layer 2 encapsulation support:
  - Point-to-Point Protocol (PPP)
  - High-level Data Link Control (HDLC)
  - Frame Relay
- Internal or network clock (selectable per port)
- Online insertion and removal (OIR)

- Hot standby router protocol (HSRP)
- Alarm reporting-24-hour history maintained, 15-minute intervals on all errors
- 16- and 32-bit cyclic redundancy checks (CRC) supported (16-bit default)
- Local and remote loopback
- Bit error rate testing (BERT) pattern generation and detection per port

Note BERT is not supported on the 8-Port Channelized T1/E1 SPA.

- Dynamic provisioning— Dynamic provisioning allows for the addition of new customer circuits within a channelized interface without affecting other customers.
- FPD (field programmable device upgrades)
- End-to-end FRF.12 fragmentation support
- Link Fragmentation and Interleaving (LFI) support
- Compressed Real-Time Protocol (cRTP)—Supported on the Cisco 7600 SIP-200 with the 8-Port Channelized T1/E1 SPA, 2-Port and 4-Port Channelized T3 SPA, 2-Port and 4-Port Clear Channel T3/E3 SPA, and 1-Port Channelized OC-3/STM-1 SPA. For more information about configuring cRTP, see the "Configuring Compressed Real-Time Protocol" section on page 4-4.
- T1 Features
  - All ports can be fully channelized down to DS0
  - Data rates in multiples of 56 Kbps or 64 Kbps per channel
  - Maximum 1.536 Mbps for each T1 port
  - D4 (SF) and ESF support for each T1 port
  - ANSI T1.403 and AT&T TR54016 CI FDL support
  - Internal and receiver recovered clocking modes
  - Short haul and long haul CSU support
  - B8ZS and AMI line encoding

  Note B8ZS and AMI line encoding are not configurable for TW on the 2-Port and 4-Port Channelized T3 SPA.

  - Support for Multilink Point-to-Point Protocol (MLPPP) for full T1s on the same SPA (hardware based) and across SPAs (software based).
  - Support for Multilink Frame Relay (MLFR)
- E1 Features
  - Maximum 1.984 Mbps for each E1 port in framed mode and a 2.048 Mbps in unframed E1 mode
  - All ports can be fully channelized down to DS0
  - Compliant with ITU G7.03, G.704, ETSI and ETS300156
  - Internal and receiver recovered clocking modes
  - HDB3 and AMI line encoding

  - Support for Multilink Point-to-Point Protocol (MLPPP) for full E1s on the same SPA (hardware based) and across SPAs (software based).
  - Support for Multilink Frame Relay (MLFR)
- E3 Features
  - Full duplex connectivity at E3 rate (34.368 MHz)
  - Supports G.751 or G.832 framing (software selectable)
  - High-density bipolar with three zones (HD3B) line coding
  - Compliant with E3 pulse mask
  - Line build-out: configured for up to 450 feet (135 m) of type 728A or equivalent coaxial cable
  - Loopback modes: DTE, local, dual, and network
  - E3 alarm/event detection (once per second polling)

    - Alarm indication signal (AIS)

    - Loss of frame (LOF)

    - Remote alarm indication (RAI)
  - Subrate and scrambling features for these DSU vendors:

    - Digital Link

    - ADC Kentrox
- T3 Features
  - Binary 3-zero substitution (B3ZS) line coding
  - Compliant with DS3 pulse mask per ANSI T1.102-1993
  - DS3 far-end alarm and control (FEAC) channel support
  - Full-duplex connectivity at DS-3 rate (44.736 MHz)
  - 672 DS0s per T3
  - Loopback modes: DTE, local, dual, and network
  - C-bit or M23 framing (software selectable)
  - Line build-out: configured for up to 450 feet (135 m) of type 734A or equivalent coaxial cable
  - DS-3 alarm/event detection (once per second polling)

    - Alarm indication signal (AIS)

    - Out of frame (OOF)

    - Far-end receive failure (FERF)
  - Generation and termination of DS3 Maintenance Data Link (MDL) in C-bit framing
  - Full FDL support and FDL performance monitoring
  - Subrate and scrambling features for these DSU vendors:

    - Digital Link

    - ADC Kentrox

    - Adtran

    - Verilink

    - Larscom

Note On a 2-port or 4-port Channelized T3 SPA, when one of the T3 ports is configured as a DS3 Clear Channel interface and the other T3s are configured with a large number (greater than or equal to 400) of low-bandwidth channels (NxDS0, N=1,2,3 or 4), the DS3 Clear Channel interface is not able to run at 100 percent DS3 line rate when those low-bandwidth channels are idle (not transmitting or receiving packets). This issue does not occur if those low-bandwidth channels are not idle.

Supported MIBs

The following MIBs are supported in Cisco IOS Release 12.2S for the serial SPAs on the Catalyst 6500 Series switch:

All serial SPAs:

- CISCO-ENTITY-ALARM-MIB
- CISCO-CLASS-BASED-QOS-MIB
- CISCO-ENVMON-MIB (For NPEs, NSEs, line cards, and SIPs only)
- CISCO-ENTITY-ASSET-MIB
- CISCO-ENTITY-FRU-CONTROL-MIB
- CISCO-ENTITY-SENSOR-MIB
- ENTITY-MIB
- IF-MIB
- RMON-MIB
- MPLS-LDP-MIB
- MPLS-LSR-MIB
- MPLS-TE-MIB
- MPLS-VPN-MIB

2 and 4-Port T3/E3 SPA:

- DS3/E3 MIB

8-Port Channelized T1/E1 SPA:

- DS1/E1 MIB

2 or 4-Port CT3 SPA:

- DS1-MIB
- DS3-MIB
- CISCO-FRAME-RELAY-MIB
- IANAifType-MIB
- RFC1381-MIB

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Displaying the SPA Hardware Type

To verify the SPA hardware type that is installed in your Catalyst 6500 Series switch, you can use the **show diagbus** command or the **show interface** command (once the interface has been configured). There are several other commands on the Catalyst 6500 Series switch that also provide SPA hardware information.

Table 14-2 shows the hardware description that appears in the **show** command output for each type of SPA that is supported on the Catalyst 6500 Series switch.

Table 14-2 SPA Hardware Descriptions in show Commands

SPA	Description in show interfaces and show controllers commands
4-Port T3/E3 SPA	Hardware is SPA-4XT3/E3
2-Port T3/E3 SPA	Hardware is SPA-2XT3/E3
8-Port Channelized T1/E1 SPA	Hardware is SPA-T1E1
2-Port CT3 SPA	Hardware is 2 ports CT3 SPA
4-Port CT3 SPA	Hardware is 4 ports CT3 SPA

Example of the show interface Command

The following example shows output from the **show interface serial 5/0/0** command on a Catalyst 6500 Series switch with a 4-Port T3/E3 SPA installed in slot 5:

```
Serial5/0/0 is up, line protocol is up
Hardware is SPA-4XT3/E3[3/0]
MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
reliability 248/255, txload 1/255, rxload 1/255
Encapsulation HDLC, crc 16, loopback not set
Keepalive set (10 sec)
Last input 00:00:06, output 00:00:07, output hang never
Last clearing of ''show interface'' counters 00:00:01
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicast)
0 runts, 0 giants, 0 throttles
```

```
0 parity
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 applique, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
```

The following example shows output from the **show interface serial 6/0/1** command on a Catalyst 6500 Series switch with a 8-Port Channelized T1/E1 SPA installed in slot 6:

```
Serial6/0/1:0 is up, line protocol is up
  Hardware is SPA-T1E1
  MTU 1500 bytes, BW 1536 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, crc 16, loopback not set
  Keepalive set (10 sec)
  LCP Open, multilink Open
  Last input 00:00:03, output 00:00:03, output hang never
  Last clearing of "show interface" counters 5d17h
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 3194905708
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     74223 packets input, 1187584 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     74227 packets output, 1187751 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     4 carrier transitions no alarm present
  Timeslot(s) Used:1-24, subrate: 64Kb/s, transmit delay is 0 flags
```

Example of the show controllers Command

The following example shows output from the **show controller serial** command on a Catalyst 6500 Series switch with a 4-Port T3/E3 SPA installed in slot 5:

```
Router# show controllers serial 5/0/2
Serial5/0/2 -
   Framing is c-bit, Clock Source is Line
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 0, since reset 0
   Data in current interval (807 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 306 Unavailable Secs
     500 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     564 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 2:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     564 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
```

```
[output omitted]
```

The following example shows output from the **show controllers** command on a Catalyst 6500 Series switch with a 8-Port Channelized T1/E1 SPA installed in slot 6:

```
Router# show controllers t1
T1 6/0/0 is up.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  No alarms detected.
  alarm-trigger is not set
  Framing is ESF, Line Code is B8ZS, Clock Source is Line.
  Data in current interval (394 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Total Data (last 24 hours)
     0 Line Code Violations, 0 Path Code Violations,
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
T1 6/0/1 is up.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  No alarms detected.
  alarm-trigger is not set
  Framing is ESF, Line Code is B8ZS, Clock Source is Line.
  Data in current interval (395 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Total Data (last 24 hours)
     0 Line Code Violations, 0 Path Code Violations,
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

The following example shows output from the **show controllers** command on a Catalyst 6500 Series switch with a 4-Port CT3 SPA installed in slot 3:

```
Router# show controllers t3
T3 3/1/2 is up.  Hardware is 4 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Channelized T3
  No alarms detected.
  Framing is M23, Line Code is B3ZS, Clock Source is Internal
  Equipment customer loopback
  Data in current interval (146 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     0 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 0 LOS Defect Secs

  T1 1 is up
  timeslots: 1-24
  FDL per AT&T 54016 spec.
```

```
No alarms detected.
Framing is ESF, Clock Source is Internal
Data in current interval (104 seconds elapsed):
   0 Line Code Violations, 0 Path Code Violations
   0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
   0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs
   0 Unavail Secs, 0 Stuffed Secs
   0 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs
Total Data (last 2 15 minute intervals):
   0 Line Code Violations,0 Path Code Violations,
   0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
   0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs
   0 Unavail Secs, 0 Stuffed Secs
   0 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs
```

CHAPTER 15

Configuring the 8-Port Channelized T1/E1 SPA

This chapter provides information about configuring the 8-Port Clear Channel T1/E1 SPA on the Catalyst 6500 Series switch. It includes the following sections:

- Configuration Tasks, page 15-1
- Verifying the Interface Configuration, page 15-19
- Configuration Examples, page 15-20

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Configuration Tasks

This section describes how to configure the 8-Port Clear Channel T1/E1 SPA for the Catalyst 6500 Series switch and includes information about verifying the configuration.

It includes the following topics:

- Required Configuration Tasks, page 15-1
- Specifying the Interface Address on a SPA, page 15-6
- Optional Configurations, page 15-6
- Configuring QoS Features on Serial SPAs, page 15-19

Required Configuration Tasks

This section lists the required configuration steps to configure the 8-Port Clear Channel T1/E1 SPA. Some of the required configuration commands implement default values that might be appropriate for your network. If the default value is correct for your network, then you do not need to configure the command.

- Setting the Card Type, page 15-2
- Enabling the Interfaces on the Controller, page 15-3

- Verifying Controller Configuration, page 15-4
- Setting the IP Address, page 15-5
- Verifying Interface Configuration, page 15-5

Note To better understand the address format used to specify the physical location of the SIP, SPA, and interfaces, see the "Specifying the Interface Address on a SPA" section on page 15-6.

Setting the Card Type

The SPA is not functional until the card type is set. Information about the SPA is not indicated in the output of any **show** commands until the card type has been set. There is no default card type.

Note Mixing of interface types is not supported. All ports on a SPA must be of the same type.

To set the card type for the 8-Port Clear Channel T1/E1 SPA, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# card type {e1 | t1} slot subslot`	Sets the serial mode for the SPA: • **t1**—Specifies T1 connectivity of 1.536 Mbps. B8ZS is the default line code for T1. • **e1**—Specifies a wide-area digital transmission scheme used predominantly in Europe that carries data at a rate of 1.984 Mbps in framed mode and a 2.048 Mbps in unframed E1 mode. • *slot subslot*—Specifies the location of the SPA. See the "Specifying the Interface Address on a SPA" section on page 15-6.
Step 3	`Router(config)# exit`	Exits configuration mode and returns to the EXEC command interpreter prompt.

Enabling the Interfaces on the Controller

To create the interfaces for the 8-Port Clear Channel T1/E1 SPA, perform this task:

	Command	Purpose
Step 1	Router(config)# **controller** {**t1** | **e1**} *slot/subslot/port*	Select the controller to configure and enter controller configuration mode. • **t1**—Specifies the T1 controller. • **e1**—Specifies the E1 controller. • *slot/subslot/port*—Specifies the location of the interface. See the: "Specifying the Interface Address on a SPA" section on page 15-6
Step 2	Router(config-controller)# **clock source** {**internal** | **line**}	Sets the clock source. **Note** The clock source is set to internal if the opposite end of the connection is set to line and the clock source is set to line if the opposite end of the connection is set to internal. • **internal**—Specifies that the internal clock source is used. • **line**—Specifies that the network clock source is used. This is the default for T1 and E1.
Step 3	Router(config-controller)# **linecode** {**ami** | **b8zs** | **hdb3**}	Selects the linecode type. • **ami**—Specifies Alternate Mark Inversion (AMI) as the linecode type. Valid for T1 and E1 controllers. • **b8zs**—Specifies binary 8-zero substitution (B8ZS) as the linecode type. Valid for T1 controller only. This is the default for T1 lines. • **hdb3**—Specifies high-density binary 3 (hdb3) as the linecode type. Valid for E1 controller only. This is the default for E1 lines.

Step 4	For T1 controllers
	Router(config-controller)# **framing** {**sf** | **esf**}
	For E1 controllers
	Router(config-controller)# **framing** {**crc4** | **no-crc4**}
		Selects the framing type.
		• **sf**—Specifies Super Frame as the T1 frame type.
		• **esf**—Specifies Extended Super Frame as the T1 frame type. This is the default for E1.
		• **crc4**—Specifies CRC4 as the E1 frame type. This is the default for E1.
		• **no-crc4**—Specifies no CRC4 as the E1 frame type.

	Command	Purpose
Step 5	`Router(config-controller)#` **`channel-group`** *`t1 t1-number`* `{`**`timeslots`** *`range`* `|` **`unframed`**`}` `[`**`speed`** `{`**`56`** `|` **`64`**`}]`	Define the time slots that belong to each T1 or E1 circuit. • *t1 t1-number*— Channel-group number. When configuring a T1 data line, channel-group numbers can be values from 1 to 28. When configuring an E1 data line, channel-group numbers can be values from 0 to 30. • **timeslots** *range*— One or more time slots or ranges of time slots belonging to the channel group. The first time slot is numbered 1. For a T1 controller, the time slot range is from 1 to 24. For an E1 controller, the time slot range is from 1 to 31. • **unframed**—Unframed mode (G.703) uses all 32 time slots for data. None of the 32 time slots are used for framing signals. • **speed**—(Optional) Speed of the underlying DS0s. – **56**—56 kbps – **64**—64 kbps **Note** The default is 64 is speed is not mentioned in the configuration. **Note** Each channel group is presented to the system as a serial interface that can be configured individually. **Note** Once a channel group has been created with the **channel-group** command, the channel group cannot be changed without removing the channel group. To remove a channel group, see the section: Changing a Channel Group Configuration, page 15-16.
Step 6	`Router(config)#` **`exit`**	Exits configuration mode and returns to the EXEC command interpreter prompt.

Verifying Controller Configuration

Use the **show controllers** command to verify the controller configuration:

```
Router(config)# show controllers t1
T1 6/0/1 is up.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  No alarms detected.
  alarm-trigger is not set
  Framing is ESF, Line Code is B8ZS, Clock Source is Line.
  Data in current interval (395 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Total Data (last 24 hours)
```

```
0 Line Code Violations, 0 Path Code Violations,
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Setting the IP Address

To set the IP address for the 8-Port Clear Channel T1/E1 SPA, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface serial slot/subslot/port:channel-group`	Selects the interface to configure from global configuration mode. • *slot/subslot/port:channel-group*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 15-6.
Step 2	`Router(config-if)# ip address address mask`	Sets the IP address and subnet mask. • *address*—IP address. • *mask*—Subnet mask.
Step 3	`Router(config)# exit`	Exits configuration mode and returns to the EXEC command interpreter prompt.

Verifying Interface Configuration

Use the **show interfaces** command to verify the interface configuration:

```
Router(config)# show interfaces
.
.
.
Serial6/0/1:0 is up, line protocol is up
  Hardware is SPA-T1E1
  MTU 1500 bytes, BW 1536 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, crc 16, loopback not set
  Keepalive set (10 sec)
  LCP Open, multilink Open
  Last input 00:00:03, output 00:00:03, output hang never
  Last clearing of "show interface" counters 5d17h
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 3194905708
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     74223 packets input, 1187584 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     74227 packets output, 1187751 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     4 carrier transitions no alarm present
  Timeslot(s) Used:1-24, subrate: 64Kb/s, transmit delay is 0 flags
.
.
```

Specifying the Interface Address on a SPA

SPA interface ports begin numbering with "0" from left to right. Single-port SPAs use only the port number 0. To configure or monitor SPA interfaces, you need to specify the physical location of the SIP, SPA, and interface in the CLI. The interface address format is *slot/subslot/port*, where:

- *slot*—Specifies the chassis slot number in the Catalyst 6500 Series switch where the SIP is installed.
- *subslot*—Specifies the secondary slot of the SIP where the SPA is installed.
- *port*—Specifies the number of the individual interface port on a SPA.

The following example shows how to specify the first interface (0) on a SPA installed in the first subslot of a SIP (0) installed in chassis slot 3:

```
Router(config)# interface serial 3/0/0
```

This command shows a serial SPA as a representative example, however the same *slot/subslot/port* format is similarly used for other SPAs (such as ATM and POS) and other non-channelized SPAs.

For the 8-Port Channelized T1/E1 SPA, the interface address format is *slot/subslot/port*:*channel-group*, where:

- *channel-group*—Specifies the logical channel group assigned to the timeslots within the T1 link.

For more information about identifying slots and subslots, see the "Identifying Slots and Subslots for SIPs, SSCs, and SPAs" section on page 4-2.

Optional Configurations

There are several standard, but optional, configurations that might be necessary to complete the configuration of your serial SPA.

Note For additional command output details, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

- Configuring Framing, page 15-7
- Configuring Encapsulation, page 15-8
- Configuring the CRC Size for T1, page 15-9
- Configuring FDL, page 15-10
- Configuring Multilink Point-to-Point Protocol (Hardware-based), page 15-11
- Configuring MLFR for T1/E1, page 15-13
- Invert Data on the T1/E1 Interface, page 15-15
- Changing a Channel Group Configuration, page 15-16
- Configuring Multipoint Bridging, page 15-16
- Configuring Bridging Control Protocol Support, page 15-16
- FRF.12 Guidelines, page 15-18
- LFI Guidelines, page 15-18
- Hardware MLPPP LFI Guidelines, page 15-18
- FRF.12 LFI Guidelines, page 15-19

- Configuring QoS Features on Serial SPAs, page 15-19

Configuring Framing

Framing is used to synchronize data transmission on the line. Framing allows the hardware to determine when each packet starts and ends. To configure framing, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **controller** {**t1** | **e1**} *slot/subslot/port*	Selects the controller to configure. • **t1**—Specifies the T1 controller. • **e1**—Specifies the E1 controller. • *slot/subslot/port*—Specifies the location of the controller. See "Specifying the Interface Address on a SPA" section on page 15-6.

Step 3	For T1 controllers:
	Router(config-controller)# **framing** {**sf** | **esf**}
	For E1 controllers:
	Router(config-controller)# **framing** {**crc4** | **no-crc4** | **unframed**}
		Set the framing on the interface.
		• **sf**—Specifies Super Frame as the T1 frame type.
		• **esf**—Specifies extended Super Frame as the T1 frame type. This is the default for T1.
		• **crc4**—Specifies CRC4 frame as the E1 frame type. This is the default for E1.
		• **no-crc4**—Specifies no CRC4 frame as the E1 frame type.
		• **unframed**—Unframed mode (G.703) uses all 32 time slots for data.

Verifying Framing Configuration

Use the **show controllers** command to verify the framing configuration:

```
Router# show controllers t1
T1 6/0/0 is down.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  Receiver has loss of frame.
  alarm-trigger is not set
  Framing is ESF, Line Code is B8ZS, Clock Source is Line.
  Data in current interval (717 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 717 Unavail Secs
  Total Data (last 24 hours)
     0 Line Code Violations, 0 Path Code Violations,
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 86400 Unavail Secs
```

Configuring Encapsulation

When traffic crosses a WAN link, the connection needs a Layer 2 protocol to encapsulate traffic. To set the encapsulation method, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface serial`** *`slot/subslot/port:channel-group`*	Selects the interface to configure. • *slot/subslot/port:channel-group*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 15-6.
Step 3	`Router(config-if)#` **`encapsulation`** `{`**`hdlc`** `|` **`ppp`** `|` **`frame-relay`**`}`	Set the encapsulation method on the interface. • **hdlc**—High-Level Data Link Control (HDLC) protocol for serial interface. This encapsulation method provides the synchronous framing and error detection functions of HDLC without windowing or retransmission. This is the default for synchronous serial interfaces. • **ppp**—PPP (for serial interface). • **frame-relay**—Frame Relay (for serial interface).

Verifying Encapsulation

Use the **show interfaces serial** command to verify encapsulation on the interface:

```
Router# show interfaces serial 6/0/0:0
Serial6/0/0:0 is down, line protocol is down
  Hardware is SPA-T1E1
  MTU 1500 bytes, BW 1536 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, crc 32, loopback not set
  Keepalive set (10 sec)
  LCP Closed, multilink Closed
  Last input 1w0d, output 1w0d, output hang never
  Last clearing of "show interface" counters 6d23h
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1152 kilobits/sec
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions alarm present
  Timeslot(s) Used:1-24, subrate: 64Kb/s, transmit delay is 0 flags
```

Configuring the CRC Size for T1

All 8-Port Clear Channel T1/E1 SPA interfaces use a 16-bit cyclic redundancy check (CRC) by default, but also support a 32-bit CRC. CRC is an error-checking technique that uses a calculated numeric value to detect errors in transmitted data. The designators 16 and 32 indicate the length (in bits) of the frame check sequence (FCS). A CRC of 32 bits provides more powerful error detection, but adds overhead. Both the sender and receiver must use the same setting.

CRC-16, the most widely used CRC throughout the United States and Europe, is used extensively with WANs. CRC-32 is specified by IEEE 802 and as an option by some point-to-point transmission standards. It is often used on Switched Multimegabit Data Service (SMDS) networks and LANs.

To set the length of the CRC on a T1 interface, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface serial`** *`slot/subslot/port:channel-group`*	Selects the interface to configure. • *slot/subslot/port:channel-group*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 15-6.
Step 3	`Router(config-if)#` **`crc`** `{`**`16`** `|` **`32`**`}`	Selects the CRC size in bits. • **16**—16-bit CRC. This is the default • **32**—32-bit CRC.

Verifying the CRC Size

Use the **show interfaces serial** command to verify the CRC size set on the interface:

```
Router# show interfaces serial 6/0/0:0
Serial6/0/0:0 is up, line protocol is up
  Hardware is SPA-T1E1
  MTU 1500 bytes, BW 1536 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, crc 32, loopback not set
  Keepalive set (10 sec)
  LCP Open, multilink Open
  Last input 00:00:38, output 00:00:00, output hang never
  Last clearing of "show interface" counters 01:46:16
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     1272 packets input, 20396 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     6 input errors, 3 CRC, 0 frame, 0 overrun, 0 ignored, 3 abort
     1276 packets output, 20460 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions no alarm present
  Timeslot(s) Used:1-24, subrate: 64Kb/s, transmit delay is 0 flags
```

Configuring FDL

Facility Data Link (FDL) is a 4-kbps channel provided by the Extended Super Frame (ESF) T1 framing format. The FDL performs outside the payload capacity and allows you to check error statistics on terminating equipment without intrusion.

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# controller t1 slot/subslot/port`	Selects the controller to configure. • *slot/subslot/port*—Specifies the location of the controller. See the "Specifying the Interface Address on a SPA" section on page 15-6.
Step 3	`Router(config-controller)# fdl [ansi | att]`	If the framing format was configured for **esf**, configures the format used for Facility Data Link (FDL). • **ansi**—Use the ANSI T1.403 standard. • **att**—Use the AT&T TR54016 standard.

Verifying FDL

Use the **show controllers t1** command to verify the FDL setting:

```
Router# show controllers t1
```

```
T1 6/0/1 is up.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  No alarms detected.
  alarm-trigger is not set
  Framing is ESF, FDL is ansi, Line Code is B8ZS, Clock Source is Line.
  Data in current interval (742 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Total Data (last 73 15 minute intervals):
     1278491 Line Code Violations, 3 Path Code Violations,
     0 Slip Secs, 1 Fr Loss Secs, 177 Line Err Secs, 0 Degraded Mins,
     3 Errored Secs, 0 Bursty Err Secs, 1 Severely Err Secs, 227 Unavail Secs
.
.
.
```

Configuring Multilink Point-to-Point Protocol (Hardware-based)

Multilink Point-to-Point Protocol (MLPPP) allows you to combine T1 or E1 lines into a bundle that has the combined bandwidth of multiple T1/E1 lines. You choose the number of bundles and the number of T1 or E1 lines in each bundle.

MLPPP for T1/E1 Configuration Guidelines

The required conditions are as follows:

- Only T1 or E1 links in a bundle.
- All links on the same SPA.
- Maximum of 12 links in a bundle.

Consider these guidelines about hardware-based MLPPP:

- Only three fragmentation sizes are supported: 128, 256, and 512 bytes.
- Fragmentation is enabled by default, with a default size of 512 bytes.
- Fragmentation size is configured using the **ppp multilink fragment-delay** command after using the **interface multilink** command. Among the three possible fragmentationsizes, the least size satisfying the delay criteria is configured. For example, a 192 byte packet causes a delay of 1 millisecond on a T1 link, so the nearest fragmentation size is 128 bytes.

  Use the **show ppp multilink** command to indicate the MLPPP type and the fragmentation size:

  ```
  Router# show ppp multilink
  Multilink1, bundle name is Patriot2
  Bundle up for 00:00:13
  Bundle is Distributed
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received, 206/255 load
  0x0 received sequence, 0x0 sent sequence
  Member links: 2 active, 0 inactive (max not set, min not set)
  Se4/2/0/1:0, since 00:00:13, no frags rcvd
  Se4/2/0/2:0, since 00:00:10, no frags rcvd
  Distributed fragmentation on. Fragment size 512.  Multilink in Hardware.
  ```

- Fragmentation is disabled explicitly by using the **no ppp multilink fragmentation** command after using the **interface multilink** command.

Creating a Multilink Bundle

To create a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface multilink group-number`	Creates a multilink interface and enter multilink interface mode. • *group-number*—The group number for the multilink bundle.
Step 3	`Router(config-if)# ip address address mask`	Sets the IP address for the multilink group. • *address*—The IP address. • *mask*—The IP netmask.

Assigning an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# **configure terminal**`	Enters global configuration mode.
Step 2	`Router(config)# **interface serial** *slot/subslot/port/t1-number:channel-group*`	Selects the interface to configure and enters interface configuration mode. See the "Specifying the Interface Address on a SPA" section on page 15-6. • *slot/subslot/port/t1-number:channel-group*—Select the interface to configure.
Step 3	`Router(config-if)# **encapsulation ppp**`	Enables PPP encapsulation.
Step 4	`Router(config-if)# **multilink-group** *group-number*`	Assigns the interface to a multilink bundle. • *group-number*—The multilink group number for the T1 or E1 bundle.
Step 5	`Router(config-if)# **ppp multilink**`	Enables multilink PPP on the interface.
	Repeat these commands for each interface you want to assign to the multilink bundle.	

Configuring Fragmentation Size on an MLPPP Bundle (optional)

To configure the fragmentation size on a multilink ppp bundle, perform this task:

	Command	Purpose
Step 1	`Router# **configure terminal**`	Enters global configuration mode.
Step 2	`Router(config)# **interface multilink** *slot/subslot/port/t1-number:channel-group*`	Creates a multilink interface and enters multilink interface mode. • *group-number*—The group number for the multilink bundle. The range is 1 to 2147483647.
Step 3	`Router(config-if)# **ppp multilink fragment-delay** *delay*`	Sets the fragmentation size satisfying the configured delay on the multilink bundle. • *delay*—Delay in milliseconds.

Disabling the Fragmentation on an MLPPP Bundle (optional)

To assign an interface to a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# **configure terminal**`	Enters global configuration mode.
Step 2	`Router(config)# **interface multilink** *group-number*`	Creates a multilink interface and enters multilink interface mode. • *group-number*—The group number for the multilink bundle. The range is 1 to 2147483647.
Step 3	`Router(config-if)# **no ppp multilink fragmentation**`	Disables the fragmentation on the multilink bundle.

Verifying Multilink PPP

Use the **show ppp multilink** command to verify the PPP multilinks:

```
Router# show ppp multilink
Multilink1, bundle name is mybundle
    Bundle up for 01:40:50
    Bundle is Distributed
    0 lost fragments, 0 reordered, 0 unassigned
    0 discarded, 0 lost received, 1/255 load
    0x0 received sequence, 0x0 sent sequence
Member links: 5 active, 0 inactive (max not set, min not set)
    Se6/0/0/1:0, since 01:40:50, no frags rcvd
    Se6/0/1/1:0, since 01:40:09, no frags rcvd
    Se6/0/3/1:0, since 01:15:44, no frags rcvd
    Se6/0/4/1:0, since 01:03:17, no frags rcvd
    Se6/0/6/1:0, since 01:01:06, no frags rcvd
    Se6/0/6:0, since 01:01:06, no frags rcvd
```

Configuring MLFR for T1/E1

Multilink Frame Relay (MLFR) allows you to combine T1/E1 lines into a bundle that has the combined bandwidth of multiple T1/E1 lines. You choose the number of bundles and the number of T1/E1 lines in each bundle. This feature allows you to increase the bandwidth of your network links beyond that of a single T1/E1 line.

MLFR for T1/E1 Configuration Guidelines

MLFR will function in hardware if all of the following conditions are met:

- Only T1 or E1 member links.
- All links are on the same SPA.
- Maximum of 12 links in a bundle.

Create a Multilink Bundle

To create a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface mfr` *number*	Configures a multilink Frame Relay bundle interface. • *number*—The number for the Frame Relay bundle.
Step 3	`Router(config-if)# frame-relay multilink bid` *name*	(Optional) Assigns a bundle identification name to a multilink Frame Relay bundle. • *name*—The name for the Frame Relay bundle. **Note** The bundle identification (BID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shut** and **no shut** commands in interface configuration mode.

Assign an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface serial`** *`slot/subslot/port:channel-group`*	Selects the interface to assign. • *slot/subslot/port:channel-group*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 15-6.
Step 3	`Router(config-if)#` **`encapsulation frame-relay mfr`** *`number`* `[`*`name`*`]`	Creates a multilink Frame Relay bundle link and associates the link with a bundle. • *number*—The number for the Frame Relay bundle. • *name*—The name for the Frame Relay bundle.
Step 4	`Router(config-if)#` **`frame-relay multilink lid`** *`name`*	(Optional) Assigns a bundle link identification name with a multilink Frame Relay bundle link. • *name*—The name for the Frame Relay bundle. **Note** The bundle link identification (LID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shut** and **no shut** commands in interface configuration mode.
Step 5	`Router(config-if)#` **`frame-relay multilink hello`** *`seconds`*	(Optional) Configures the interval at which a bundle link will send out hello messages. The default value is 10 seconds. • *seconds*—Number of seconds between hello messages sent out over the multilink bundle.
Step 6	`Router(config-if)#` **`frame-relay multilink ack`** *`seconds`*	(Optional) Configures the number of seconds that a bundle link will wait for a hello message acknowledgment before resending the hello message. The default value is 4 seconds. • *seconds*—Number of seconds a bundle link will wait for a hello message acknowledgment before resending the hello message.
Step 7	`Router(config-if)#` **`frame-relay multilink retry`** *`number`*	(Optional) Configures the maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment. The default value is 2 tries. • *number*—Maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment.

Verifying Multilink Frame Relay

Use the **show frame-relay multilink detailed** command to verify the Frame Relay multilinks:

```
router# show frame-relay multilink detailed

Bundle: MFR49, State = down, class = A, fragmentation disabled
 BID = MFR49
 No. of bundle links = 1, Peer's bundle-id =
 Bundle links:

  Serial6/0/0:0, HW state = up, link state = Add_sent, LID = test
    Cause code = none, Ack timer = 4, Hello timer = 10,
    Max retry count = 2, Current count = 0,
    Peer LID = , RTT = 0 ms
    Statistics:
    Add_link sent = 21, Add_link rcv'd = 0,
    Add_link ack sent = 0, Add_link ack rcv'd = 0,
    Add_link rej sent = 0, Add_link rej rcv'd = 0,
    Remove_link sent = 0, Remove_link rcv'd = 0,
    Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,
    Hello sent = 0, Hello rcv'd = 0,
    Hello_ack sent = 0, Hello_ack rcv'd = 0,
    outgoing pak dropped = 0, incoming pak dropped = 0
```

Invert Data on the T1/E1 Interface

If the interface on the 8-Port Clear Channel T1/E1 SPA is used to drive a dedicated T1 line that does not have B8ZS encoding, you must invert the data stream on the connecting CSU/DSU or on the interface. Be careful not to invert data on both the CSU/DSU and the interface, as two data inversions will cancel each other out. To invert data on a T1/E1 interface, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface serial`** *`slot/subslot/port:channel-group`*	Selects the serial interface.
Step 3	`Router(config-if)#` **`invert data`**	Inverts the data stream.

Use the **show running configuration** command to verify that invert data has been set:

```
Router# show running configuration
.
.
.
interface Serial6/0/0:0
 no ip address
 encapsulation ppp
 logging event link-status
 load-interval 30
 invert data
 no cdp enable
 ppp chap hostname group1
 ppp multilink
 multilink-group 1
!
.
.
.
```

Changing a Channel Group Configuration

To alter the configuration of an existing channel group, the channel group needs to be removed first. To remove an existing channel group, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **controller** {**t1** | **e1**} *slot/subslot/port*	Select the controller to configure and enter controller configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 15-6.
Step 3	Router(config-controller)# **no channel-group** *t1 t1-number*	Select the channel group you want to remove. • *t1 t1-number*— Channel-group number.
	Follow the steps in the "Enabling the Interfaces on the Controller" section on page 15-3 to create a new channel group with the new configuration.	

Configuring Multipoint Bridging

Multipoint bridging (MPB) enables the connection of multiple ATM PVCs, Frame Relay PVCs, BCP ports, and WAN Gigabit Ethernet subinterfaces into a single broadcast domain (virtual LAN), together with the LAN ports on that VLAN. This enables service providers to add support for Ethernet-based Layer 2 services to the proven technology of their existing ATM and Frame Relay legacy networks. You can use your current VLAN-based networks over the ATM or Frame Relay cloud. This feature also allows service providers to gradually update their core networks to the latest Gigabit Ethernet optical technologies, while still supporting their existing customer base.

For MPB configuration guidelines and restrictions and feature compatibility tables, see the "Configuring Multipoint Bridging" section on page 4-17 of Chapter 4, "Configuring the SIPs and SSC."

Configuring Bridging Control Protocol Support

The Bridging Control Protocol (BCP) enables forwarding of Ethernet frames over SONET networks and provides a high-speed extension of enterprise LAN backbone traffic through a metropolitan area. The implementation of BCP on the SPAs includes support for IEEE 802.1D, IEEE 802.1Q Virtual LAN (VLAN), and high-speed switched LANs.

For BCP configuration guidelines and restrictions and feature compatibility tables, see the "Configuring PPP Bridging Control Protocol Support" section on page 4-18 of Chapter 4, "Configuring the SIPs and SSC."

Configuring BCP on MLPPP

Consider the following guidelines when configuring BCP on MLPPP:

- Only Distributed MLPPP is supported.
- Only channelized interfaces are allowed, and member links must be from the same controller card.
- Only trunk port BCP is supported on MLPPP.
- Bridging can be configured only on the bundle interface.

Note BCP on MLPPP operates only in trunk mode. For more inforation on trunk mode, see the "Configuring BCP in Trunk Mode" section on page 4-19 of Chapter 4, "Configuring the SIPs and SSC."

Configuring BCP on MLPPP Trunk Mode

To configure BCP on MLPPP trunk mode, perform this task:

	Command	Purpose
Step 1	Router(config)#**interface multilink**	Selects the multilink interface.
Step 2	Router(config-if)#**switchport**	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 3	Router(config-if)#**switchport trunk allowed vlan** *vlan-list*	By default, no VLANs are allowed. Use this command to explicitly allow VLANs; valid values for *vlan-list* are from 1 to 4094.
Step 4	Router(config-if)#**switchport mode trunk**	Configures the router port connected to the switch as a VLAN trunk port.
Step 5	Router(config-if)#**switchport nonegotiate**	Puts the LAN port into permanent trunking mode but prevents the port from generating DTP frames.
Step 6	Router(config-if)#**no ip address**	Unassigns the IP address.
Step 7	Router(config-if)#**ppp multilink**	Enables this interface to support MLP.
Step 8	Router(config-if)#**multilink-group 1**	Assigns this interface to the multilink group.
Step 9	Router(config-if)#**interface Serial1/0/0.1/1/1/1:0**	Designates a serial interface as a multilink bundle.
Step 10	Router(config-if)#**no ip address**	Unassigns the IP address.
Step 11	Router(config-if)#**encapsulation ppp**	Enables PPP encapsulation.
Step 12	Router(config-if)#**ppp multilink**	Enables this interface to support MLP.
Step 13	Router(config-if)#**multilink-group 1**	Assigns this interface to the multilink group 1.
Step 14	Router(config-if)#**interface Serial1/0/0.1/1/1/2:0**	Designates a serial interface as a multilink bundle.
Step 15	Router(config-if)#**no ip address**	Unassigns the IP address.
Step 16	Router(config-if)#**encapsulation ppp**	Enables PPP encapsulation.
Step 17	Router(config-if)#**ppp multilink**	Enables this interface to support MLP.
Step 18	Router(config-if)#**multilink-group 2**	Assigns this interface to the multilink group 2.
Step 19	Router(config-if)#**shutdown**	Shuts down an interface.
Step 20	Router(config-if)#**no shutdown**	Reopens an interface.
Step 21	Router(config-if)#**switchport trunk allowed vlan** *vlan-list*	By default, no VLANs are allowed. Use this command to explicitly allow VLANs; valid values for *vlan-list* are from 1 to 4094.

Verifying BCP on MLPPP Trunk Mode

To display information about Multilink PPP, perform this task in EXEC mode.

Command	Purpose
`Router(config-if)# show ppp multilink`	Displays information on a multilink group.

The following shows an example of **show ppp multilink**:

```
Router# show ppp multilink

Multilink1, bundle name is group 1
Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links: 4 active, 0 inactive (max no set, min not set)
Serial1/0/0/:1
Serial1/0/0/:2
Serial1/0/0/:3
Serial1/0/0/:4
```

FRF.12 Guidelines

FRF.12 functions in hardware. Note the following guidelines:

- The fragmentation is configured at the main interface.
- Only three fragmentation sizes are supported: 128, 256, and 512 bytes.

LFI Guidelines

LFI can function by using either FRF.12 or MLPPP. MLPPP LFI operates in both hardware and software while FRF.12 LFI operates only in hardware.

Hardware MLPPP LFI Guidelines

LFI using MLPPP will function only in hardware if there is just one member link in the MLPPP bundle. The link can be a fractional T1 or full T1. Note the following guidelines:

- The **ppp multilink interleave** command must be configured to enable interleaving.
- Only three fragmentation sizes are supported: 128 bytes, 256 bytes, and 512 bytes.
- Fragmentation is enabled by default, and the default size is 512 bytes.
- A policy map having a priority class must be applied to the main interface.
- When hardware-based LFI is enabled, fragmentation counters are not displayed.

FRF.12 LFI Guidelines

LFI using FRF.12 is always performed in hardware. Note the following guidelines:

- The fragmentation is configured at the main interface.
- Only three fragmentation sizes are supported: 128 bytes, 256 bytes, and 512 bytes.
- A policy map having a priority class must be applied to the main interface.

Configuring QoS Features on Serial SPAs

For information about the QoS features supported by the serial SPAs, see the "Configuring QoS Features on a SIP" section on page 4-33 of Chapter 4, "Configuring the SIPs and SSC."

Saving the Configuration

To save your running configuration to nonvolatile random-access memory (NVRAM), perform this task in privileged EXEC configuration mode:

Command	Purpose
`Router# copy running-config startup-config`	Writes the new configuration to NVRAM.

For more information about managing configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

Verifying the Interface Configuration

In addition to using the **show running-configuration** command to display your Catalyst 6500 Series switch configuration settings, you can use the **show interfaces serial** and the **show controllers serial** commands to get detailed information on a per-port basis for your 8-Port Clear Channel T1/E1 SPA.

Verifying Per-Port Interface Status

To find detailed interface information on a per-port basis for the 8-Port Clear Channel T1/E1 SPA, use the **show interfaces serial** command. For a description of the command output, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX.*

The following example provides sample output for interface port 0 on the SPA located in the first subslot of the SIP installed in slot 6 of a Catalyst 6509 switch:

```
Router# show interface serial 6/0/0:0
Serial6/0/0:0 is up, line protocol is up
  Hardware is SPA-T1E1
  MTU 1500 bytes, BW 1536 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, crc 32, loopback not set
  Keepalive set (10 sec)
  LCP Open, multilink Open
  Last input 00:00:38, output 00:00:00, output hang never
  Last clearing of "show interface" counters 01:46:16
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     1272 packets input, 20396 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     6 input errors, 3 CRC, 0 frame, 0 overrun, 0 ignored, 3 abort
     1276 packets output, 20460 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions no alarm present
  Timeslot(s) Used:1-24, subrate: 64Kb/s, transmit delay is 0 flags
```

Configuration Examples

This section includes the following configuration examples:

- Framing and Encapsulation Configuration Example, page 15-20
- CRC Configuration Example, page 15-21
- Facility Data Link Configuration Example, page 15-21
- MLPPP Configuration Example, page 15-21
- Invert Data on the T1/E1 Interface Example, page 15-23
- MFR Configuration Example, page 15-22

Framing and Encapsulation Configuration Example

The following example sets the framing and encapsulation for the controller and interface:

```
! Specify the controller and enter controller configuration mode
!
Router(config)# controller t1 6/0/0
!
! Specify the framing method
!
Router(config-controller)# framing esf
!
! Exit controller configuration mode and return to global configuration mode
!
Router(config-controller)# exit
!
! Specify the interface and enter interface configuration mode
!
Router(config)# interface serial 6/0/0:0
!
! Specify the encapsulation protocol
!
Router(config-if)# encapsulation ppp
!
! Exit interface configuratin mode
!
Router(config-if)# exit
!
! Exit global configuration mode
```

```
!
Router(config)# exit
```

CRC Configuration Example

The following example sets the CRC size for the interface:

```
! Specify the interface and enter interface configuration mode
!
Router(config)# interface serial 6/0/0:0
!
! Specify the CRC size
!
Router(config-if)# crc 32
!
! Exit interface configuration mode and return to global configuration mode
!
Router(config-if)# exit
!
! Exit global configuration mode
!
Router(config)# exit
```

Facility Data Link Configuration Example

The following example configures Facility Data Link:

```
! Specify the controller and enter controller configuration mode
!
Router(config)# controller t1 6/0/0
!
! Specify the FDL specification
!
Router(config-controller)# fdl ansi
!
! Exit controller configuration mode and return to global configuration mode
!
Router(config-controller)# exit
!
! Exit global configuration mode
!
Router(config)# exit
```

MLPPP Configuration Example

The following example creates a PPP Multilink bundle:

```
! Enter global configuration mode
!
Router# configure terminal
!
! Create a multilink bundle and assign a group number to the bundle
!
Router(config)# interface multilink 1
!
! Specify an IP address for the multilink group
!
Router(config-if)# ip addres 123.456.789.111 255.255.255.0
```

```
!
! Enable Multilink PPP
!
Router(config-if)# ppp multilink
!
! Leave interface multilink configuration mode
!
Router(config-if)# exit
!
! Specify the interface to assign to the multilink bundle
!
Router(config)# interface serial 3/1//0:1
!
! Enable PPP encapsulation on the interface
!
Router(config-if)# encapsulation PPP
!
! Assign the interface to a multilink bundle
!
Router(config-if)# multilink-group 1
!
! Enable Multilink PPP
!
Router(config-if)# ppp multilink
!
! Exit interface configuration mode
!
Router(config-if)# exit
!
! Exit global configuration mode
!
Router(config)# exit
```

MFR Configuration Example

The following example configures Multilink Frame Relay (MFR):

```
! Create a MFR interface and enter interface configuration mode
!
Router(config)# interface mfr 49
!
! Assign the bundle identification (BID) name 'test' to a multilink bundle.
!
Router(config-if)# frame-relay multilink bid test
!
! Exit interface configuration mode and return to global configuration mode
!
Router(config-if)# exit
!
! Specify the serial interface to assign to a multilink bundle
!
Router(config)# interface serial 5/1/3:0
!
! Creates a multilink Frame Relay bundle link and associates the link with a multilink
bundle
!
Router(config-if)# encapsulation frame-relay mfr 49
!
! Assigns a bundle link identification (LID) name with a multilink bundle link
!
Router(config-if)# frame-relay multilink lid test
!
```

```
! Configures the interval at which the interface will send out hello messages
!
Router(config-if)# frame-relay multilink hello 15
!
! Configures the number of seconds the interface will wait for a hello message
acknowledgement before resending the hello message
!
Router(config-if)# frame-relay multilink ack 6
!
! Configures the maximum number of times the interface will resend a hello message while
waiting for an acknowledgement
!
Router(config-if)# frame-relay multilink retry 5
!
! Exit interface configuration mode and return to global configuration mode
!
Router(config-if)# exit
!
! Exit global configuration mode
!
Router(config)# exit
```

Invert Data on the T1/E1 Interface Example

The following example inverts the data on the serial interface:

```
! Enter global configuration mode
!
Router# configure terminal
!
! Specify the serial interface and enter interface configuration mode
!
Router(config)# interface serial 5/1/3:0
!
! Configure invert data
!
Router(config-if)# invert data
!
! Exit interface configuration mode and return to global configuration mode
!
Router(config-if)# exit
!
! Exit global configuration mode
!
Router(config)# exit
```

CHAPTER 16

Configuring the 2-Port and 4-Port Clear Channel T3/E3 SPAs

This chapter provides information about configuring the 2-Port and 4-Port Clear Channel T3/E3 Shared Port Adapters (SPAs) on the Catalyst 6500 Series switch. It includes the following sections:

- Configuration Tasks, page 16-1
- Verifying the Interface Configuration, page 16-18
- Configuration Examples, page 16-20

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX.*. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Configuration Tasks

This section describes how to configure the 2-Port Clear Channel T3/E3 SPA for the Catalyst 6500 Series switch and includes information about verifying the configuration.

It includes the following topics:

- Required Configuration Tasks, page 16-2
- Specifying the Interface Address on a SPA, page 16-5
- Optional Configurations, page 16-5
- Saving the Configuration, page 16-18

Required Configuration Tasks

This section lists the required configuration steps to configure the 2-Port and 4-Port Clear Channel T3/E3 SPA. Some of the required configuration commands implement default values that might be appropriate for your network. If the default value is correct for your network, then you do not need to configure the command.

- Setting the Card Type, page 16-2
- Configure the Interface, page 16-3

Note To better understand the address format used to specify the physical location of the Spa Interface Processor (SIP), SPA, and interfaces, see the "Specifying the Interface Address on a SPA" section on page 16-5.

Setting the Card Type

The SPA is not functional until the card type is set. Information about the SPA is not indicated in the output of any show commands until the card type has been set. There is no default card type.

Note Mixing of interface types is not supported. All ports on a SPA will be the of the same type.

To set the card type for the 2-Port and 4-Port Clear Channel T3/E3 SPA, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# card type {t3 | e3} slot subslot`	Sets the serial mode for the SPA: • **t3**—Specifies T3 connectivity of 44210 kbps through the network, using B3ZS coding. • **e3**—Specifies a wide-area digital transmission scheme used predominantly in Europe that carries data at a rate of 34010 kbps. • *slot subslot*—Specifies the location of the SPA. See the "Specifying the Interface Address on a SPA" section on page 16-5.
Step 3	`Router(config)# exit`	Exits configuration mode and return to the EXEC command interpreter prompt.

Configure the Interface

To set the ip address for the 2-Port and 4-Port Clear Channel T3/E3 SPA, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface serial** *slot/subslot/port*	Selects the interface to configure and enters interface configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 16-5.
Step 2	Router(config-if)# **ip address** *address mask*	Sets the IP address and subnet mask. • *address*—IP address • *mask*—Subnet mask
Step 3	Router(config-if)# **clock source** {**internal** | **line**}	Sets the clock source to internal. • **internal**—Specifies that the internal clock source is used. • **line**—Specifies that the network clock source is used. This is the default.
Step 4	Router(config-if)# **no shutdown**	Enables the interface.
Step 5	Router(config)# **exit**	Exits configuration mode and returns to the EXEC command interpreter prompt.

Verifying Controller Configuration

Use the **show controllers** command to verify the controller configuration:

```
Router# show controllers serial 6/0/0
Serial6/0/0 -
   Framing is c-bit, Clock Source is Line
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 2, since reset 0
   Data in current interval (546 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
.
.
.
Data in Interval 44:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     560 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
   Total Data (last 44 15 minute intervals):
     0 Line Code Violations, 0 P-bit Coding Violation,
     0 C-bit Coding Violation,
     0 P-bit Err Secs, 0 P-bit Sev Err Secs,
     0 Sev Err Framing Secs, 0 Unavailable Secs,
     24750 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs

   Transmitter is sending AIS.

   Receiver has loss of signal.

    40434 Sev Err Line Secs, 0 Far-End Err Secs, 0 Far-End Sev Err Secs
    0 P-bit Unavailable Secs, 0 CP-bit Unavailable Secs
    0 CP-bit Far-end Unavailable Secs
    0 Near-end path failures, 0 Far-end path failures

   No FEAC code is being received
  MDL transmission is disabled
```

Use the **show controllers brief** command to view a subset of the **show controllers** output:

```
Router# show controllers serial 6/0/2 brief
Serial6/0/2 -
   Framing is c-bit, Clock Source is Internal
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 0, since reset 22

   No alarms detected.

   No FEAC code is being received
  MDL transmission is disabled
```

Verifying Interface Configuration

Use the **show interfaces** command to verify the interface configuration:

```
Router# show interfaces serial 6/0/0
Serial6/0/0 is up, line protocol is up
  Hardware is SPA-4T3E3
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 12/255, rxload 56/255
  Encapsulation FRAME-RELAY, crc 16, loopback not set
  Keepalive set (10 sec)
  LMI enq sent  13477, LMI stat recvd 13424, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 19, LMI stat sent  0, LMI upd sent  0
  LMI DLCI 1023  LMI type is CISCO  frame relay DTE
  FR SVC disabled, LAPF state down
  Broadcast queue 0/256, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:09, output 00:00:09, output hang never
  Last clearing of "show interface" counters 1d13h
  Input queue: 0/75/3/3891 (size/max/drops/flushes); Total output drops: 5140348
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 9716000 bits/sec, 28149 packets/sec
  5 minute output rate 2121000 bits/sec, 4466 packets/sec
     14675957334 packets input, 645694448563 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
              0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     14562482078 packets output, 640892196653 bytes, 0 underruns
     0 output errors, 0 applique, 4 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
   rxLOS inactive, rxLOF inactive, rxAIS inactive
   txAIS inactive, rxRAI inactive, txRAI inactive
```

```
Serial6/0/0.16 is up, line protocol is up
  Hardware is SPA-4T3E3
  Internet address is 110.1.1.2/24
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 11/255, rxload 53/255
  Encapsulation FRAME-RELAY
```

Specifying the Interface Address on a SPA

SPA interface ports begin numbering with “0” from left to right. Single-port SPAs use only the port number 0. To configure or monitor SPA interfaces, you need to specify the physical location of the SIP, SPA, and interface in the CLI. The interface address format is *slot*/*subslot*/*port*, where:

- *slot*—Specifies the chassis slot number in the Catalyst 6500 Series switch where the SIP is installed.
- *subslot*—Specifies the secondary slot of the SIP where the SPA is installed.
- *port*—Specifies the number of the individual interface port on a SPA.

The following example shows how to specify the first interface (0) on a SPA installed in the first subslot of a SIP (0) installed in chassis slot 3:

```
Router(config)# interface serial 3/0/0
```

This command shows a serial SPA as a representative example, however the same *slot*/*subslot*/*port* format is similarly used for other SPAs (such as ATM and POS) and other non-channelized SPAs.

For more information about identifying slots and subslots, see the “Identifying Slots and Subslots for SIPs, SSCs, and SPAs” section on page 4-2.

Optional Configurations

There are several standard, but optional configurations that might be necessary to complete the configuration of your serial SPA.

Note For additional command output details, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

- Configuring Data Service Unit Mode, page 16-6
- Configuring Maintenance Data Link, page 16-8
- Configuring Scramble, page 16-11
- Configuring Framing, page 16-13
- Configuring Encapsulation, page 16-14
- Configuring Cable Length, page 16-15
- Configuring Invert Data, page 16-16
- Configuring the Trace Trail Buffer, page 16-17
- Configuring Multipoint Bridging, page 16-18
- Configuring Bridging Control Protocol Support, page 16-18
- Saving the Configuration, page 16-18

Configuring Data Service Unit Mode

Configure the SPA to connect with customer premise Data Service Units (DSUs) by setting the DSU mode. Subrating a T3 or E3 interface reduces the peak access rate by limiting the data transfer rate. To configure the DSU mode and bandwidth, perform this task:

	Command	Purpose
Step 1	`Router# `**`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)# `**`interface serial`** *`slot/subslot/port`*	Selects the interface to configure and enters interface configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 16-5.

Step 3	`T3`
	`Router(config-if)# `**`dsu mode`**` {`**`0`**` | `**`1`**` | `**`2`**` | `**`3`**` | `**`4`**`}`
	`E3`
	`Router(config-if)# `**`dsu mode`**` {`**`0`**` | `**`1`**`}`
		Specifies the interoperability mode used by a T3 controller.
		• **0**—Connects a T3/E3 controller to another T3/E3 controller or to a Digital Link DSU (DL3100 in T3 mode and DL3100E in E3 mode). This is the default.
		• **1**—Connects a T3/E3 controller to a Kentrox DataSMART T3/E3 IDSU.
		• **2**—Connects a T3 controller to a Larscom Access-T45 DS3 DSU.
		• **3**—Connects a T3 controller to an Adtran T3SU 300.
		• **4**—Connects a T3 controller to a Verilink HDM 2182.

	Command	Purpose
Step 4	`Router(config-if)# dsu bandwidth kbps`	Specifies the allowable bandwidth. • *kbps*—The bandwidth range and increment values are based on the specific DSU. Default for T3 mode is 44010 kbps and 34010 kbps for E3 mode. • Digital Link DL3100 – range: 300 to 44210 kbps – increments: 300 kbps • Digital Link DL3100E – range: 358 to 34010 kbps – increments: 358 kbps • Kentrox DataSMART T3/E3 IDSU – range: 1000 to 34000 kbps (E3 mode) – range: 1500 to 44210 kbps (T3 mode) – increments: 500 kbps • Larscom Access-T45 DS3 – range: 3100 to 44210 kbps – increments: 3100 kbps • Adtran T3SU 300 – range: 80 to 44210 kbps – increments: 80 kbps • Verilink HDM 2182 – range: 1600 to 31600 kbps – increments: 1600 kbps
Step 5	`Router(config-if)# remote {accept | fullrate}`	Specifies where the DSU bandwidth is set. • **accept**—Accept incoming remote requests to reset the DSU bandwidth. • **fullrate**—Set far end DSU to its fullrate bandwidth.

Verifying DSU Mode

Use the **show controllers serial** command to display the DSU settings:

```
Router# show controllers serial 6/0/0
Serial6/0/0 -
   Framing is c-bit, Clock Source is Line
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 2, since reset 0
   Data in current interval (546 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
.
.
.
```

Configuring Maintenance Data Link

MDL messages are used to communicate identification information between local and remote ports. The type of information included in MDL messages includes the equipment identification code (EIC), location identification code (LIC), frame identification code (FIC), unit, Path Facility Identification (PFI), port number, and Generator Identification numbers.

Note C-bit framing has to be enabled in order to transport MDL messages between source and destination T3 ports.

To configure Maintenance Data Link (MDL), perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.

	Command	Purpose

Step 2	`Router(config)# `**`interface serial`**
	`slot/subslot/port`
		Selects the interface to configure.
		- *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 16-5.

	Command	Purpose
Step 3	`Router(config-if)# mdl [string {eic | fic | generator | lic | pfi | port | unit} string}] | [transmit {idle-signal | path | test-signal}]`	Configures the Maintenance Data Link (MDL) message. • **eic** *string*—Equipment identification code (up to 10 characters), which is a value used to describe a specific piece of equipment according to ANSI T1.107-1995. • **fic** *string*—Frame identification code (up to 10 characters), which is a value used to identify where the equipment is located within a building at a given location according to ANSI T1.107-1995. • **generator** *string*—Specifies the Generator number string sent in the MDL Test Signal message; can be up to 38 characters. • **lic** *string*—Location identification code (up to 11 characters), which is a value used to describe a specific location according to ANSI T1.107-1995. • **pfi** *string*—Specifies the Path Facility Identification Code sent in the MDL Path message; can be up to 38 characters. • **port** *string*—Specifies the Port number string sent in the MDL Idle Signal message; can be up to 38 characters. • **unit** *string*—Unit identification code (up to 6 characters), which is a value that identifies the equipment location within a subslot according to ANSI T1.107-1995. • **transmit idle-signal**—Enables transmission of the MDL idle signal message. An MDL idle signal message, as defined by ANSI T1.107, is distinguished from path and test signal messages in that it contains a port number as its final data element. • **transmit path**—Enables transmission of the MDL path message. An MDL path message, as defined by ANSI T1.107, is distinguished from idle and test signal messages in that it contains a facility identification code as its final data element. • **transmit test-signal**—Enables transmission of the MDL test signal message. An MDL test signal message, as defined by ANSI T1.107, is distinguished from path and idle signal messages in that it contains a generator number as its final data element.

Verifying MDL

Use the **show controllers serial** command to display the MDL settings:

```
Router# show controllers serial 6/0/0
Serial6/0/0 -
   Framing is c-bit, Clock Source is Line
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 2, since reset 0
   Data in current interval (546 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
.
.
.

  Data in Interval 96:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
   Total Data (last 24 hours)
     0 Line Code Violations, 0 P-bit Coding Violation,
     0 C-bit Coding Violation,
     0 P-bit Err Secs, 0 P-bit Sev Err Secs,
     0 Sev Err Framing Secs, 0 Unavailable Secs,
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs

   No alarms detected.

    0 Sev Err Line Secs, 1 Far-End Err Secs, 0 Far-End Sev Err Secs
    0 P-bit Unavailable Secs, 0 CP-bit Unavailable Secs
    0 CP-bit Far-end Unavailable Secs
    0 Near-end path failures, 0 Far-end path failures

No FEAC code is being received
  MDL transmission is enabled
     EIC: tst, LIC: 67,
     Test Signal GEN_NO: test
  Far-End MDL Information Received
     EIC: tst, LIC: 67,
     Test Signal GEN_NO: test
```

Configuring Scramble

T3/E3 scrambling is used to assist clock recovery on the receiving end. Scrambling is designed to randomize the pattern of 1s and 0s carried in the physical layer frame. Randomizing the digital bits can prevent continuous, nonvariable bit patterns—in other words, long strings of all 1s or all 0s. Several physical layer protocols rely on transitions between 1s and 0s to maintain clocking.

Scrambling can prevent some bit patterns from being mistakenly interpreted as alarms by switches placed between the Data Service Units (DSUs).

To configure scrambling, perform this task:

	Command	Purpose
Step 1	`Router# `**`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)# `**`interface serial`** *`slot/subslot/port`*	Selects the interface to configure. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 16-5.
Step 3	`Router(config-if)# [`**`no`**`] `**`scramble`**	Enables scrambling. Scrambling is disabled by default. • **scramble**—Enables scramble. • **no scramble**—Disables scramble. **Note** When using framing bypass, **no scrambling** must be configured.

Verifying Scramble Configuration

Use the **show controllers serial** command to display the scrambling setting:

```
Router# show controllers serial 6/0/0
Serial6/0/0 -
   Framing is c-bit, Clock Source is Line
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 2, since reset 0
   Scrambling is enabled
   Data in current interval (356 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
.
.
.
```

Configuring Framing

Framing is used to synchronize data transmission on the line. Framing allows the hardware to determine when each packet starts and ends. To configure framing, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial` *`slot/subslot/port`*	Selects the interface to configure. • *slot/subslot/port*—Specifies the location of the T3/E3 interface. See the "Specifying the Interface Address on a SPA" section on page 16-5.

Step 3	`T3`
	`Router(config-if)# framing {bypass | c-bit | m13}`
	`E3`
	`Router(config-if)# framing {bypass | g751 | g832}`
		Sets the framing on the interface.
		• **bypass**—Configure framing bypass to use the full T3 or E3 bandwidth.
		• **c-bit**—Specifies C-bit parity framing. This is the default for T3.
		• **m13**—Specifies M13 framing.
		• **g751**— Specifies g751 framing. This is the default for E3.
		• **g832**—Specifies g832 framing.

Verifying Framing Configuration

Use the **show controllers serial** command to display the framing method:

```
Router# show controllers serial 6/0/0
Serial6/0/0 -
   Framing is c-bit, Clock Source is Line
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 2, since reset 0
   Data in current interval (546 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
.
.
.
```

Configuring Encapsulation

When traffic crosses a WAN link, the connection needs a Layer 2 protocol to encapsulate traffic. To set the encapsulation method, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial slot/subslot/port`	Selects the interface to configure. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 16-5.
Step 3	`Router(config-if)# encapsulation {hdlc | ppp | frame-relay}`	Sets the encapsulation method on the interface. • **hdlc**—High-Level Data Link Control (HDLC) protocol for serial interface. This is the default. • **ppp**—PPP (for serial interface). • **frame-relay**—Frame Relay (for serial interface).

Verifying Encapsulation

Use the **show interfaces** command to display the encapsulation method:

```
Router# show interfaces serial 6/0/1
Serial6/0/1 is up, line protocol is up
  Hardware is SPA-4T3E3
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 223/255, rxload 222/255
  Encapsulation FRAME-RELAY, crc 16, loopback not set
  Keepalive set (10 sec)
  LMI enq sent  13076, LMI stat recvd 13076, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent  0, LMI upd sent  0
  LMI DLCI 0  LMI type is ANSI Annex D  frame relay DTE
  FR SVC disabled, LAPF state down
  Broadcast queue 0/256, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:04, output 00:00:04, output hang never
  Last clearing of "show interface" counters 1d12h
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 38579000 bits/sec, 109611 packets/sec
  5 minute output rate 38671000 bits/sec, 109852 packets/sec
     14374551065 packets input, 632486376132 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
              0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     14408526130 packets output, 633974757440 bytes, 0 underruns
     0 output errors, 0 applique, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
   rxLOS inactive, rxLOF inactive, rxAIS inactive
   txAIS inactive, rxRAI inactive, txRAI inactive
```

Configuring Cable Length

The **cablelength** command compensates for the loss in decibels based on the distance from the device to the first repeater in the circuit. A longer distance from the device to the repeater requires that the signal strength on the circuit be boosted to compensate for loss over that distance. To configure cable length, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial slot/subslot/port`	Selects the interface to configure and enters interface configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 16-5.
Step 3	`Router(config-if)# cablelength length`	Sets the cable length. • *length*—Range is 0-450 feet. The default is 10 feet.

Verify Cable Length Setting

Use the **show interfaces serial** command to verify the cable length setting:

```
Router# show interfaces serial 4/0/0
Serial4/0/0 -
   Framing is c-bit, Clock Source is Internal
   Bandwidth limit is 44210, DSU mode 0, Cable length is 200
   rx FEBE since last clear counter 0, since reset 22
   Data in current interval (446 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 2:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
.
.
.
```

Configuring Invert Data

Delays between the TE clock and data transmission indicate that the transmit clock signal might not be appropriate for the interface rate and length of cable being used. Different ends of the wire may have variances that differ slightly. Invert the clock signal to compensate for these factors. To configure invert data, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial slot/subslot/port`	Selects the interface to configure and enters interface configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 16-5.
Step 3	`Router(config-if)# invert {data}`	Inverts the data. • **data**—Invert the data stream.

Verify Invert Data Setting

Use the **show running configuration** command to verify that invert data was set on the interface:

```
Router# show running configuration
.
.
.
interface Serial6/0/0
 ip address 51.1.1.1 255.255.255.0
 logging event link-status
 dsu bandwidth 44210
 framing c-bit
 cablelength 10
 clock source internal
 invert data
 mdl string eic tst
 mdl string lic 67
 mdl string generator test
 mdl transmit path
 mdl transmit test-signal
 no cdp enable
!
.
.
.
```

Configuring the Trace Trail Buffer

Configure the Trace Trail Buffer (TTB) to send messages to the remote device. The TTB messages check for the continued presence of the transmitter. To configure TTB, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface serial`** *`slot/subslot/port`*	Selects the interface to configure and enters interface configuration mode. - *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 16-5.
Step 3	`Router(config-if)#` **`ttb`** `{`**`country`** `|` **`rnode`** `|` **`serial`** `|` **`snode`** `|` **`soperator`** `|` **`x`**`}` *`string`*	Sends a Trace Trail Buffer message in E3 g.832 framing mode. - **country**—Two character country code - **rnode**—Receive node code - **serial**—M.1400 serial - **snode**—Sending location/Node ID code - **soperator**—Sending operator code (Must be numeric.) - **x**—X0 - *string*—TTB message

Verify TTB Settings

Use the **show controllers serial** command to display the TTB settings for the interface:

```
Router# show controllers serial 6/0/0
Serial6/0/0 -
   Framing is c-bit, Clock Source is Line
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 2, since reset 0
   Data in current interval (546 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
.
.
.
No alarms detected.
TTB transmission is disabled
TTB Rx: country: us soperator: s snode: sn rnode: rn x: x serial: 1
```

Configuring Multipoint Bridging

Multipoint bridging (MPB) enables the connection of multiple ATM PVCs, Frame Relay PVCs, BCP ports, and WAN Gigabit Ethernet subinterfaces into a single broadcast domain (virtual LAN), together with the LAN ports on that VLAN. This enables service providers to add support for Ethernet-based Layer 2 services to the proven technology of their existing ATM and Frame Relay legacy networks. Customers can then use their current VLAN-based networks over the ATM or Frame Relay cloud. This also allows service providers to gradually update their core networks to the latest Gigabit Ethernet optical technologies, while still supporting their existing customer base.

For MPB configuration guidelines and restrictions and feature compatibility tables, see the "Configuring Multipoint Bridging" section on page 4-17 of Chapter 4, "Configuring the SIPs and SSC."

Configuring Bridging Control Protocol Support

The Bridging Control Protocol (BCP) enables forwarding of Ethernet frames over SONET networks and provides a high-speed extension of enterprise LAN backbone traffic through a metropolitan area. The implementation of BCP on the SPAs includes support for IEEE 802.1D, IEEE 802.1Q Virtual LAN (VLAN), and high-speed switched LANs.

For BCP configuration guidelines and restrictions and feature compatibility tables, see the "Configuring PPP Bridging Control Protocol Support" section on page 4-18 of Chapter 4, "Configuring the SIPs and SSC."

Configuring QoS Features on Serial SPAs

For information about the QoS features supported by the serial SPAs, see the "Configuring QoS Features on a SIP" section on page 4-33 of Chapter 4, "Configuring the SIPs and SSC."

Saving the Configuration

To save your running configuration to nonvolatile random-access memory (NVRAM), perform this task in privileged EXEC configuration mode:

Command	Purpose
`Router# copy running-config startup-config`	Writes the new configuration to NVRAM.

For more information about managing configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

Verifying the Interface Configuration

In addition to using the **show running-configuration** command to display your Catalyst 6500 Series switch configuration settings, you can use the **show interfaces serial** and the **show controllers serial** commands to get detailed information on a per-port basis for your 2-Port and 4-Port Clear Channel T3/E3 SPA.

Verifying Per-Port Interface Status

To find detailed interface information on a per-port basis for the 2-Port and 4-Port Clear Channel T3/E3 SPA, use the **show interfaces serial** command. For a description of the command output, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

The following example provides sample output for interface port 1 on the SPA located in the first subslot of the SIP installed in slot 5 of a Catalyst 6500 Series switch:

```
Router# show interface serial 5/0/1
Serial5/0/1 is up, line protocol is up
  Hardware is SPA-4T3E3
  Internet address is 120.1.1.1/24
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 234/255, rxload 234/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:00, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 40685000 bits/sec, 115627 packets/sec
  5 minute output rate 40685000 bits/sec, 115624 packets/sec
     4652915554 packets input, 204728203496 bytes, 0 no buffer
     Received 4044 broadcasts (0 IP multicast)
     130 runts, 0 giants, 0 throttles
              0 parity
     1595 input errors, 543 CRC, 0 frame, 0 overrun, 0 ignored, 922 abort
     4653081242 packets output, 204735493748 bytes, 0 underruns
     0 output errors, 0 applique, 4 interface resets
     0 output buffer failures, 0 output buffers swapped out
     2 carrier transitions
```

Monitoring Per-Port Interface Statistics

To find detailed status and statistical information on a per-port basis for the 2-Port and 4-Port Clear Channel T3/E3 SPA, use the **show controllers serial** command. For a description of the command output, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

The following example provides sample output for interface port 1 on the SPA located in the first subslot of the SIP that is installed in slot 5 of the Catalyst 6500 Series switch:

```
Router# show controller serial 5/0/2
Serial5/0/2 -
   Framing is c-bit, Clock Source is Line
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 0, since reset 0
   Data in current interval (807 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 306 Unavailable Secs
     500 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     564 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 2:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     564 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 3:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     562 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 4:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     560 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
    .
    .
    .
    Total Data (last 44 15 minute intervals):
         0 Line Code Violations, 0 P-bit Coding Violation,
         0 C-bit Coding Violation,
         0 P-bit Err Secs, 0 P-bit Sev Err Secs,
         0 Sev Err Framing Secs, 0 Unavailable Secs,
         24750 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs

       Transmitter is sending AIS.

       Receiver has loss of signal.

        40434 Sev Err Line Secs, 0 Far-End Err Secs, 0 Far-End Sev Err Secs
        0 P-bit Unavailable Secs, 0 CP-bit Unavailable Secs
        0 CP-bit Far-end Unavailable Secs
        0 Near-end path failures, 0 Far-end path failures

       No FEAC code is being received
      MDL transmission is disabled
```

Configuration Examples

This section includes the following configuration examples:

- DSU Configuration Example, page 16-21
- MDL Configuration Example, page 16-21
- Scrambling Configuration Example, page 16-21
- Framing Configuration Example, page 16-22
- Encapsulation Configuration Example, page 16-22
- Cable Length Configuration Example, page 16-22
- Invert Data Configuration Example, page 16-22
- Trace Trail Buffer Configuration Example, page 16-23

DSU Configuration Example

The following example confgiures DSU on interface port 0 on slot 4, subslot 1.

```
! Specify the serial interface and enter interface configuration mode
!
Router(config)# interface serial 4/1/0
!
! Specify the DSU mode
!
Router(config-if)# dsu mode 0
!
! Specify the DSU bandwidth
!
Router(config-if)# dsu bandwidth 10000
!
! Set the DSU bandwidth to accept or reject the incoming remote requests
!
Router(config-if)# dsu remote accept
```

MDL Configuration Example

The following example configures the MDL strings on interface port 0 on slot 4, subslot 1.

```
! Specify the serial interface and enter interface configuration mode
!
Router(config)# interface serial 4/1/0
!
! Specify the MDL strings
!
Router(config-if)# mdl string eic beic
Router(config-if)# mdl string lic beic
Router(config-if)# mdl string fic bfix
Router(config-if)# mdl string unit bunit
Router(config-if)# mdl string pfi bpfi
Router(config-if)# mdl string port bport
Router(config-if)# mdl string generator bgen
Router(config-if)# mdl transmit path
Router(config-if)# mdl transmit idle-signal
Router(config-if)# mdl transmit test-signal
```

Scrambling Configuration Example

The following example configures scrambling on the T3/E3 interface:

```
! Enter global configuration mode
!
Router# configure terminal
!
! Specify the serial interface and enter interface configuration mode
!
Router(config)# interface serial 4/1/3
!
! Enable scrambling
!
Router(config-if)# scrambling
```

Framing Configuration Example

The following example configures framing on interface port 1 on slot 4, subslot 1.

```
! Specify the serial interface and enter interface configuration mode
!
Router(config)# interface serial 4/1/1
!
! Specify the framing method
!
Router(config-if)# framing m13
```

Encapsulation Configuration Example

The following example configures encapsulation on interface port 1 on slot 4, subslot 1.

```
! Specify the serial interface and enter interface configuration mode
!
Router(config)# interface serial 4/1/1
!
! Specify the encapsulation method
!
Router(config-if)# encapsulation PPP
```

Cable Length Configuration Example

The following example configures sets the cable length to 200 feet:

```
! Enter global configuration mode
!
Router# configure terminal
!
! Specify the serial interface and enter interface configuration mode
!
Router(config)# interface serial 4/1/3
!
! Specify the cable length
!
Router(config-if)# cablelength 200
```

Invert Data Configuration Example

The following example enables invert data:

```
! Enter global configuration mode
!
Router# configure terminal
!
! Specify the serial interface and enter interface configuration mode
!
Router(config)# interface serial 4/1/3
!
! Enable invert data
!
Router(config-if)# invert data
```

Trace Trail Buffer Configuration Example

The following example configures the TTB attributes:

```
! Enter global configuration mode
!
Router# configure terminal
!
! Specify the serial interface and enter interface configuration mode
!
Router(config)# interface serial 4/1/3
!
! Specify the TTB attributes
!
Router(config-if)# ttb country ab
Router(config-if)# ttb soperator 56
Router(config-if)# ttb snode 34
Router(config-if)# ttb rnode cd
Router(config-if)# ttb x 7
Router(config-if)# ttb serial 12
```

CHAPTER 17

Configuring the 2-Port and 4-Port Channelized T3 SPAs

This chapter provides information about configuring the 2-Port and 4-Port Channelized T3 Shared Port Adapters (SPAs) on the Catalyst 6500 Series switch. It includes the following sections:

- Configuration Tasks, page 17-1
- Verifying the Interface Configuration, page 17-24
- Configuration Examples, page 17-26

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Configuration Tasks

This section describes how to configure the serial SPAs for the Catalyst 6500 Series switch and includes information about verifying the configuration.

It includes the following topics:

- Required Configuration Tasks, page 17-2
- Specifying the Interface Address on a SPA, page 17-7
- Optional Configurations, page 17-8
- Configuring QoS Features on Serial SPAs, page 17-23

Required Configuration Tasks

This section lists the required configuration steps to configure the 2-Port and 4-Port Channelized T3 SPA. Some of the required configuration commands implement default values that might be appropriate for your network.

- Configuring the T3 Controller, page 17-2
- Configuring the Logical T1 Interfaces, page 17-3
- Verifying T3 Controller Configuration, page 17-5
- Verifying Interface Configuration, page 17-6

Note To better understand the address format used to specify the physical location of the SPA Interface Processor (SIP), SPA, and interfaces, see the "Specifying the Interface Address on a SPA" section on page 17-7.

Configuring the T3 Controller

To configure the T3 controller for the 2-Port and 4-Port Channelized T3 SPA, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# controller t3 slot/subslot/port`	Selects the controller to configure and enters controller configuration mode. • *slot/subslot/port*—Specifies the location of the CT3 SPA port. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-controller)# [no] channelized`	(Optional) Specifies the channelization mode. • **channelized**—In channelized mode, the T3 link can be channelized into 28 T1s, and each T1 can be further channelized into 24 DS0s. This is the default. • **no channelized**—In the unchannelized mode, the T3 link provides a single high-speed data channel of 44210 kbps.

	Command	Purpose
Step 4	`Router(config-controller)# framing {auto-detect | c-bit | m23}`	(Optional) Specifies the framing type in channelized mode. • **auto-detect**—Detects the framing type at the device at the end of the line and switches to that framing type. If both devices are set to auto-detect, c-bit framing is used. • **c-bit**—Specifies C-bit parity framing. This is the default. • **m23**—Specifies M23 framing. **Note** To set the framing type for an unchannelized T3, see the "Configuring T3 Framing" section on page 17-13.
Step 5	`Router(config-controller)# clock source {internal | line}`	(Optional) Specifies the clock source. • **internal**—Specifies that the internal clock source is used. Default for channelized mode. • **line**—Specifies that the network clock source is used. Default for unchannelized mode.
Step 6	`Router(config-controller)# cablelength length`	(Optional) Specifies the cable length. • *length*—Range is 0-450 feet. The default is 224 feet.

Configuring the Logical T1 Interfaces

If channelized mode is configured for the T3 controller, perform this task to configure the logical T1 interfaces:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# controller t3 slot/subslot/port`	Selects the controller to configure and enters controller configuration mode. • *slot/subslot/port*—Specifies the location of the CT3 SPA port. See the "Specifying the Interface Address on a SPA" section on page 17-7.

	Command	Purpose
Step 3	Router(config-controller)# **t1** *t1-number* **channel-group** *channel-number* **timeslots** *range* [**speed** {**56** | **64**}]	Specifies the T1 channel and timeslots to be mapped to each channel. • *t1-number*—T1 number from 1–28. • *channel-number*—Specifies a channel-group mapping(0–23) under the designated T1. • *range*—List of timeslots under the channel-group. Timeslots assigned to this T1 can be 1–24 or a combination of subranges within 1– 24. You can indicate a range using a hyphen, commas, or a combination of both. One timeslot equals one DS0. • **speed 56** or **64**— Specifies the speed of a timeslot as either 56 or 64 kbps. The default speed of 64 kbps is not mentioned in the configuration.
Step 4	Router(config-controller)# **t1** *t1-number* **framing** {**esf** | **sf** [**hdlc-idle** {**0x7e** | **0xff**}] [**mode** {**j1**}]}	(Optional) Specifies the T1 framing type using the **framing** command. • **sf**—Specifies Super Frame as the T1 frame type. **Note** If you select sf framing, you should consider disabling yellow alarm detection because the yellow alarm can be incorrectly detected with sf framing. • **esf**—Specifies Extended Super Frame as the T1 frame type. This is the default. • **hdlc-idle**— The hdlc-idle option allows you to set the idle pattern for the T1 interface to either **0x7e** (the default) or **0xff**.
Step 5	Router(config-controller)# **t1** *channel-number* **clock source** {**internal** | **line**}	(Optional) Specifies the T1 clock source. • **internal**—Specifies that the internal clock source is used. This is the default. • **line**—Specifies that the network clock source is used.

After configuring a logical T1 interface, configure the serial interfaces. For detailed interface configuration information, see the *Cisco IOS Interface Configuration Guide, Release 12.2*.

Note After a T1 channel is configured, it appears to the Cisco IOS software as a serial interface; therefore, all the configuration commands for a serial interface are available. However, not all commands are applicable to the T1 interface. All the encapsulation formats, such as PPP, HDLC, and Frame Relay are applicable to the configured T1. Encapsulation can be set via the serial interface configuration commands.

Verifying T3 Controller Configuration

Use the **show controllers** command to verify the controller configuration:

```
Router# show controllers t3
T3 3/1/0 is administratively down.
T3 3/1/1 is administratively down.
T3 3/1/2 is up.  Hardware is 4 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Channelized T3
  No alarms detected.
  Framing is M23, Line Code is B3ZS, Clock Source is Internal
  Equipment customer loopback
  Data in current interval (746 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     0 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 0 LOS Defect Secs

  T1 1 is up
  timeslots: 1-24
  FDL per AT&T 54016 spec.
  No alarms detected.
  Framing is ESF, Clock Source is Internal
  Data in current interval (177 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs
     0 Unavail Secs, 0 Stuffed Secs
     0 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs
  Total Data (last 2 15 minute intervals):
     0 Line Code Violations,0 Path Code Violations,
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs
     0 Unavail Secs, 0 Stuffed Secs
     0 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs

  T1 2
    Not configured.

  T1 3
    Not configured.
.
.
```

```
.
T3 3/1/3 is up.  Hardware is 4 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Subrate T3
  No alarms detected.
  MDL transmission is disabled

  FEAC code received: No code is being received
  Framing is C-BIT Parity, Line Code is B3ZS, Clock Source is Line
  Equipment customer loopback
  Data in current interval (657 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     0 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 0 LOS Defect Secs
```

Verifying Interface Configuration

Use the **show interface serial** command to verify the interface configuration. The following example shows the ouput for the serial interface for an unchannelized T3:

```
Router# show interface serial3/0/0
Serial3/0/0 is down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
              0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 applique, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions alarm present
  DSU mode 0, bandwidth 44210 Kbit, scramble 0, VC 0
```

The following example shows the output for a serial interface for the first T1 on a channelized T3:

```
Router# show interface serial3/0/1/1:0
Serial3/0/1/1:0 is administratively down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 1500 bytes, BW 832 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
```

```
Keepalive set (10 sec)
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 no buffer
   Received 0 broadcasts (0 IP multicast)
   0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   0 packets output, 0 bytes, 0 underruns
   0 output errors, 0 collisions, 1 interface resets
   0 output buffer failures, 0 output buffers swapped out
   0 carrier transitions alarm present
VC 1: timeslot(s): 2-14, Transmitter delay 0, non-inverted data
```

Specifying the Interface Address on a SPA

SPA interface ports begin numbering with “0” from left to right. Single-port SPAs use only the port number 0. To configure or monitor SPA interfaces, you need to specify the physical location of the SIP, SPA, and interface in the CLI. The interface address format is *slot***/***subslot***/***port*, where:

- *slot*—Specifies the chassis slot number in the Catalyst 6500 Series switch where the SIP is installed.
- *subslot*—Specifies the secondary slot of the SIP where the SPA is installed.
- *port*—Specifies the number of the individual interface port on a SPA.

The following example shows how to specify the first interface (0) on a SPA installed in the first subslot of a SIP (0) installed in chassis slot 3:

```
Router(config)# interface serial 3/0/0
```

This command shows a serial SPA as a representative example, however the same *slot*/*subslot*/*port* format is similarly used for other SPAs (such as ATM and POS) and other non-channelized SPAs.

For the 4-Port Channelized T3 SPA, the interface address format is *slot/subslot/port/t1-number***:***channel-group*, where:

- *t1-number*—Specifies the logical T1 number in channelized mode.
- *channel-group*—Specifies the logical channel group assigned to the timeslots within the T1 link.

For more information about identifying slots and subslots, see the “Identifying Slots and Subslots for SIPs, SSCs, and SPAs” section on page 4-2.

Optional Configurations

There are several standard, but optional, configurations that might be necessary to complete the configuration of your serial SPA.

Note For additional command output details, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication.

- Configuring the Data Service Unit Mode, page 17-9
- Configuring Maintenance Data Link, page 17-10
- Configuring Encapsulation, page 17-12
- Configuring T3 Framing, page 17-13
- Configuring FDL, page 17-14
- Configuring Scramble, page 17-15
- Configuring Multilink Point-to-Point Protocol (Hardware-Based), page 17-16
- Configuring MLFR for T1/E1, page 17-18
- Configuring Multipoint Bridging, page 17-21
- Configuring Bridging Control Protocol Support, page 17-21
- Configuring BCP on MLPPP, page 17-21
- Configuring Multipoint Bridging, page 17-21
- Link Fragmentation and Interleaving (LFI) Guidelines, page 17-23
- Hardware MLPPP LFI Guidelines, page 17-23
- FRF.12 LFI Guidelines, page 17-23
- Configuring QoS Features on Serial SPAs, page 17-23

Configuring the Data Service Unit Mode

Configure the SPA to connect with customer premise Data Service Units (DSUs) by setting the DSU mode. Subrating a T3 or E3 interface reduces the peak access rate by limiting the data transfer rate. To configure the Data Service Unit (DSU) mode, perform this task:

z

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface serial`** *`slot/subslot/port`*	Selects the controller to configure and enters controller configuration mode. • *slot/subslot/port*—Specifies the location of the controller. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-if)#` **`dsu mode`** `{`**`0`** `|` **`1`** `|` **`2`** `|` **`3`** `|` **`4`**`}`	Specifies the interoperability mode used by the T3 controller. • **0**—Connects a T3 controller to another T3 controller or to a Digital Link DSU. Bandwidth range is from 300 to 44210 kbps. This is the default. • **1**—Connects a T3 controller to a Kentrox DSU. Bandwidth range is from 1500 to 35000, or 44210 kbps. **Note** If the bandwidth is set between 35000–44210 kbps, an error message is displayed. • **2**—Connects a T3 controller to a Larscom DSU. Bandwidth range is from 3100 to 44210 kbps. • **3**—Connects a T3 controller to an Adtran T3SU 300. Bandwidth range is from 75 to 44210 kbps. • **4**—Connects a T3 controller to a Verilink HDM 2182. Bandwidth range is from 1500 to 44210 kbps.
Step 4	`Router(config-if)#` **`dsu bandwidth`** *`kbps`*	Specifies the maximum allowable bandwidth. • *kbps*—Bandwidth range is from 1 to 44210 kbps.

Verifying DSU Mode

To display the DSU mode of the controller, enter the **show controllers serial** command:

```
Router# show controllers serial
Serial3/1/0 -
   Framing is c-bit, Clock Source is Internal
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 0, since reset 0
   Data in current interval (0 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
```

```
        0 AIS Defect Secs, 0 LOS Defect Secs

      Transmitter is sending AIS.
.
.
```

Configuring Maintenance Data Link

MDL messages are used to communicate identification information between local and remote ports. The type of information included in MDL messages includes the equipment identification code (EIC), location identification code (LIC), frame identification code (FIC), unit, Path Facility Identification (PFI), port number, and Generator Identification numbers. To configure Maintenance Data Link (MDL), perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`controller t3`** *`slot/subslot/port`*	Selects the controller to configure and enters controller configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-controller)#` **`mdl`** `[`**`string`** `{`**`eic`** `|` **`fic`** `|` **`generator`** `|` **`lic`** `|` **`pfi`** `|` **`port`** `|` **`unit`**`}` *`string`*`}] |` **`[transmit {idle-signal`** `|` **`path`** `|` **`test-signal`**`}]`	Configures the MDL message. • **string eic**—Specifies the Equipment Identification Code; can be up to 10 characters. • **string fic**—Specifies the Frame Identification Code; can be up to 10 characters. • **string generator**—Specifies the Generator number string sent in the MDL Test Signal message; can be up to 38 characters. • **string lic**— Specifies the Location Identification Code; can be up to 11 characters. • **string pfi**—Specifies the Path Facility Identification Code sent in the MDL Path message; can be up to 38 characters. • **string port**—Specifies the Port number string sent in the MDL Idle Signal message; can be up to 38 characters. • **string unit**—Specifies the Unit Identification Code; can be up to 6 characters. • **transmit idle-signal**—Enable MDL Idle-Signal message transmission • **transmit path**—Enable MDL Path message transmission. • **transmit test-signal**—Enable MDL Test-Signal message transmission.

Verifying MDL

To display the MDL settings, enter the **show controller** command:

```
Router# show controller t3 3/0/0
T3 3/0/0 is down.  Hardware is 2 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Subrate T3
  Receiver has loss of signal.
  MDL transmission is enabled
     EIC: new, LIC: US, FIC: 23, UNIT: myunit
     Path FI: test pfi
     Idle Signal PORT_NO: New-port
     Test Signal GEN_NO: test-message
  FEAC code received: No code is being received
  Framing is C-BIT Parity, Line Code is B3ZS, Clock Source is Line
  Equipment customer loopback
  Data in current interval (869 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     869 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     869 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 870 LOS Defect Secs
```

Configuring Encapsulation

When traffic crosses a WAN link, the connection needs a Layer 2 protocol to encapsulate traffic. To set the encapsulation method, perform this task:

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

Step 2	For channelized:
	`Router(config)# interface serial slot/subslot/port/t1-number:channel-group`
	For unchannelized:
	`Router(config)# interface serial slot/subslot/port`
		Selects the interface to configure and enters interface configuration mode.
		• Channelized:
		slot/subslot/port/t1-number:channel-group—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
		• Unchannelized:
		slot/subslot/port—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.

Step 3	`Router(config-if)# encapsulation {hdlc | ppp | frame-relay}`	Set the encapsulation method on the interface. • **hdlc**—High-Level Data Link Control (HDLC) protocol for serial interface. This is the default. • **ppp**—Point-to-Point Protocol (PPP) (for serial interface). • **frame-relay**—Frame Relay (for serial interface).

Verifying Encapsulation

To display the encapsulation method, enter the **show interface serial** command:

```
Router# show interface serial3/0/0
Serial3/0/0 is down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
              0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 applique, 2 interface resets
```

```
      0 output buffer failures, 0 output buffers swapped out
      1 carrier transitions alarm present
   DSU mode 0, bandwidth 44210 Kbit, scramble 0, VC 0
```

Configuring T3 Framing

To set the T3 framing type, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial` *`slot/subslot/port`*	Selects the interface to configure and enters interface configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-if)# framing {c-bit | m13}`	Specifies the framing type in unchannelized mode. • **c-bit**—Specifies C-bit parity framing. This is the default. • **m13**—Specifies DS3 Framing M13 (same as M23).

Verifying Framing

To display the framing type, enter the **show controller** command:

```
Router# show controller t3 3/0/0
T3 3/0/0 is down.  Hardware is 2 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Subrate T3
  Receiver has loss of signal.
  Framing is M13, Line Code is B3ZS, Clock Source is Line
  Equipment customer loopback
  Data in current interval (656 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     666 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 666 LOS Defect Secs
```

Configuring FDL

Facility Data Link (FDL) is a far-end performance reporting tool. In ANSI mode, you can enable 1-second transmissions of performance reports on both ends of the T1 connection. To configure FDL, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial slot/subslot/port`	Selects the interface to configure and enters interface configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-controller)# t1 number fdl {ansi}`	(Optional) Enables FDL. • *number*—Specifies the T1 channel number. • **ansi**—Specifies the FDL bit per the ANSI T1.403 specification.

Verifying FDL

To display the FDL setting, enter the **show controller** command:

```
Router# show controller t3 3/0/1/1
T3 3/0/1 is down.  Hardware is 2 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Channelized T3
  Receiver has loss of signal.
  Framing is M23, Line Code is B3ZS, Clock Source is Internal
  Equipment customer loopback
  Data in current interval (456 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     456 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 456 LOS Defect Secs

  T1 1 is down
  timeslots: 2-14
  FDL per ANSI T1.403 and AT&T 54016 spec.
  Configured for FDL remotely line looped (bell)
  Transmitter is sending LOF Indication.
  Receiver is getting AIS.
  Framing is ESF, Clock Source is Line
  BERT running on timeslots 2,3,4,5,6,7,8,9,10,11,12,13,14,
  BERT test result (running)
     Test Pattern : All 1's, Status : Not Sync, Sync Detected : 0
     Interval : 2 minute(s), Time Remain : 2 minute(s)
     Bit Errors (since BERT started): 0 bits,
        Bits Received (since BERT started): 0 Kbits
        Bit Errors (since last sync): 0 bits
        Bits Received (since last sync): 0 Kbits
     Data in current interval (703 seconds elapsed):
        0 Line Code Violations, 0 Path Code Violations
        0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
        0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs
        713 Unavail Secs, 0 Stuffed Secs
        357 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs
```

Configuring Scramble

T3 scrambling is used to assist clock recovery on the receiving end. Scrambling is designed to randomize the pattern of 1s and 0s carried in the physical layer frame. Randomizing the digital bits can prevent continuous, nonvariable bit patterns (long strings of all 1s or all 0s). Several physical layer protocols rely on transitions between 1s and 0s to maintain clocking.

Scrambling can prevent some bit patterns from being mistakenly interpreted as alarms by switches placed between the Data Service Units (DSUs).

To configure scrambling, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface serial`** *`slot/subslot/port`*	Selects the interface to configure and enters interface configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the “Specifying the Interface Address on a SPA” section on page 17-7.
Step 3	`Router(config-if)#` **`scramble`** `[`**`0`** `|` **`1`**`]`	Enables scrambling. Scrambling is disabled by default. • Scramble settings: **1**—enabled **0**—disabled

Verifying Scrambling

To display the scramble setting, enter the **show interface serial** command:

```
Router# show interface serial3/0/0
Serial3/0/0 is down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
```

```
      0 runts, 0 giants, 0 throttles
               0 parity
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
      0 packets output, 0 bytes, 0 underruns
      0 output errors, 0 applique, 4 interface resets
      0 output buffer failures, 0 output buffers swapped out
      1 carrier transitions alarm present
   DSU mode 0, bandwidth 44210 Kbit, scramble 1, VC 0
```

Configuring Multilink Point-to-Point Protocol (Hardware-Based)

Multilink Point-to-Point Protocol (MLPPP) allows you to combine T1 or E1 lines into a bundle that has the combined bandwidth of multiple T1/E1 lines. You choose the number of bundles and the number of T1 or E1 lines in each bundle.

MLPPP for T1/E1 Configuration Guidelines

The required conditions are as follows:

- Only T1 or E1 links in a bundle.
- All links on the same SPA.
- Maximum of 12 links in a bundle.

Consider these guidelines about hardware-based MLPPP:

- Only three fragmentation sizes are supported: 128, 256, and 512 bytes.
- Fragmentation is enabled by default, with a default size of 512 bytes.
- Fragmentation size is configured using the **ppp multilink fragment-delay** command after using the **interface multilink** command. Among the three possible fragmentationsizes, the least size satisfying the delay criteria is configured. For example, a 192 byte packet causes a delay of 1 millisecond on a T1 link, so the nearest fragmentation size is 128 bytes.

  Use the **show ppp multilink** command to indicate the MLPPP type and the fragmentation size:

  ```
  Router# show ppp multilink
  Multilink1, bundle name is Patriot2
  Bundle up for 00:00:13
  Bundle is Distributed
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received, 206/255 load
  0x0 received sequence, 0x0 sent sequence
  Member links: 2 active, 0 inactive (max not set, min not set)
  Se4/2/0/1:0, since 00:00:13, no frags rcvd
  Se4/2/0/2:0, since 00:00:10, no frags rcvd
  Distributed fragmentation on. Fragment size 512.  Multilink in Hardware.
  ```

Fragmentation is disabled explicitly by using the **no ppp multilink fragmentation** command after using the **interface multilink** command.

Creating a Multilink Bundle

To create a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface multilink group-number`	Creates a multilink interface and enter multilink interface mode. • *group-number*—The group number for the multilink bundle.
Step 3	`Router(config-if)# ip address address mask`	Sets the IP address for the multilink group. • *address*—The IP address. • *mask*—The IP netmask.

Assigning an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial slot/subslot/port/t1-number:channel-group`	Selects the interface to configure and enters interface configuration mode. See the "Specifying the Interface Address on a SPA" section on page 17-7. • *slot/subslot/port/t1-number:channel-group*—Select the interface to configure.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# multilink-group group-number`	Assigns the interface to a multilink bundle. • *group-number*—The multilink group number for the T1 or E1 bundle.
Step 5	`Router(config-if)# ppp multilink`	Enables multilink PPP on the interface.
	Repeat these commands for each interface you want to assign to the multilink bundle.	

Configuring Fragmentation Size on an MLPPP Bundle (optional)

To configure the fragmentation size on a multilink ppp bundle, perform this task:

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

	Command	Purpose
Step 2	Router(config)# **interface multilink** *slot/subslot/port/t1-number:channel-group*	Creates a multilink interface and enters multilink interface mode. - *group-number*—The group number for the multilink bundle. The range is 1 to 2147483647.
Step 3	Router(config-if)# **ppp multilink fragment-delay** *delay*	Sets the fragmentation size satisfying the configured delay on the multilink bundle. - *delay*—delay in milliseconds

Disabling the Fragmentation on an MLPPP Bundle (optional)

To assign an interface to a multilink bundle, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface multilink** *group-number*	Creates a multilink interface and enters multilink interface mode. - *group-number*—The group number for the multilink bundle. The range is 1 to 2147483647.
Step 3	Router(config-if)# **no ppp multilink fragmentation**	Disables the fragmentation on the multilink bundle.

Verifying Multilink PPP

To verify the PPP multilinks, enter the **show ppp multilink** command:

```
Router# show ppp multilink
Multilink1, bundle name is mybundle
    Bundle up for 01:40:50
    Bundle is Distributed
    0 lost fragments, 0 reordered, 0 unassigned
    0 discarded, 0 lost received, 1/255 load
    0x0 received sequence, 0x0 sent sequence
Member links: 5 active, 0 inactive (max not set, min not set)
    Se6/0/0/1:0, since 01:40:50, no frags rcvd
    Se6/0/1/1:0, since 01:40:09, no frags rcvd
    Se6/0/3/1:0, since 01:15:44, no frags rcvd
    Se6/0/4/1:0, since 01:03:17, no frags rcvd
    Se6/0/6/1:0, since 01:01:06, no frags rcvd
    Se6/0/6:0, since 01:01:06, no frags rcvd
```

Configuring MLFR for T1/E1

Multilink Frame Relay (MLFR) allows you to combine T1/E1 lines into a bundle that has the combined bandwidth of multiple T1/E1 lines. You choose the number of bundles and the number of T1/E1 lines in each bundle. This allows you to increase the bandwidth of your network links beyond that of a single T1/E1 line.

MLFR for T1/E1 Configuration Guidelines

MLFR will function in hardware if all of the following conditions are met:

- Only T1 or E1 member links.
- All links are on the same SPA.
- Maximum of 12 links in a bundle.

Create a Multilink Bundle

To create a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface mfr number`	Configures a multilink Frame Relay bundle interface. • *number*—The number for the Frame Relay bundle.
Step 3	`Router(config-if)# frame-relay multilink bid name`	(Optional) Assigns a bundle identification name to a multilink Frame Relay bundle. • *name*—The name for the Frame Relay bundle. **Note** The bundle identification (BID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shutdown** and **no shutdown** commands in interface configuration mode.

Assign an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial slot/subslot/port:channel-group`	Selects the interface to assign. • *slot/subslot/port:channel-group*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-if)# encapsulation frame-relay mfr number [name]`	Creates a multilink Frame Relay bundle link and associates the link with a bundle. • *number*—The number for the Frame Relay bundle. • *name*—The name for the Frame Relay bundle.

	Command	Purpose
Step 4	`Router(config-if)# `**`frame-relay multilink lid`** *`name`*	(Optional) Assigns a bundle link identification name with a multilink Frame Relay bundle link. • *name*—The name for the Frame Relay bundle. **Note** The bundle link identification (LID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shut** and **no shut** commands in interface configuration mode.
Step 5	`Router(config-if)# `**`frame-relay multilink hello`** *`seconds`*	(Optional) Configures the interval at which a bundle link will send out hello messages. The default value is 10 seconds. • *seconds*—Number of seconds between hello messages sent out over the multilink bundle.
Step 6	`Router(config-if)# `**`frame-relay multilink ack`** *`seconds`*	(Optional) Configures the number of seconds that a bundle link will wait for a hello message acknowledgment before resending the hello message. The default value is 4 seconds. • *seconds*—Number of seconds a bundle link will wait for a hello message acknowledgment before resending the hello message.
Step 7	`Router(config-if)# `**`frame-relay multilink retry`** *`number`*	(Optional) Configures the maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment. The default value is 2 tries. • *number*—Maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment.

Verifying Multilink Frame Relay

To verify the Frame Relay multilinks, enter the **show frame-relay multilink detailed** command:

```
Router# show frame-relay multilink detailed
Bundle: MFR49, State = down, class = A, fragmentation disabled
 BID = MFR49
 No. of bundle links = 1, Peer's bundle-id =
 Bundle links:

  Serial6/0/0:0, HW state = up, link state = Add_sent, LID = test
    Cause code = none, Ack timer = 4, Hello timer = 10,
    Max retry count = 2, Current count = 0,
    Peer LID = , RTT = 0 ms
    Statistics:
    Add_link sent = 21, Add_link rcv'd = 0,
    Add_link ack sent = 0, Add_link ack rcv'd = 0,
    Add_link rej sent = 0, Add_link rej rcv'd = 0,
    Remove_link sent = 0, Remove_link rcv'd = 0,
    Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,
    Hello sent = 0, Hello rcv'd = 0,
    Hello_ack sent = 0, Hello_ack rcv'd = 0,
    outgoing pak dropped = 0, incoming pak dropped = 0
```

Configuring Multipoint Bridging

Multipoint bridging (MPB) enables the connection of multiple ATM PVCs, Frame Relay PVCs, BCP ports, and WAN Gigabit Ethernet subinterfaces into a single broadcast domain (virtual LAN), together with the LAN ports on that VLAN. This feature enables service providers to add support for Ethernet-based Layer 2 services to the proven technology of their existing ATM and Frame Relay legacy networks. Customers can then use their current VLAN-based networks over the ATM or Frame Relay cloud. This feature also allows service providers to gradually update their core networks to the latest Gigabit Ethernet optical technologies, while still supporting their existing customer base.

For MPB configuration guidelines and restrictions and feature compatibility tables, see the "Configuring Multipoint Bridging" section on page 4-17 of Chapter 4, "Configuring the SIPs and SSC."

Configuring Bridging Control Protocol Support

The Bridging Control Protocol (BCP) enables forwarding of Ethernet frames over SONET networks and provides a high-speed extension of enterprise LAN backbone traffic through a metropolitan area. The implementation of BCP on the SPAs includes support for IEEE 802.1D, IEEE 802.1Q Virtual LAN (VLAN), and high-speed switched LANs.

For BCP configuration guidelines and restrictions and feature compatibility tables, see the "Configuring PPP Bridging Control Protocol Support" section on page 4-18 of Chapter 4, "Configuring the SIPs and SSC."

Configuring BCP on MLPPP

Consider the following guidelines when configuring BCP on MLPPP:

- Only Distributed MLPPP is supported.
- Only channelized interfaces are allowed, and member links must be from the same controller card.
- Only trunk port BCP is supported on MLPPP.
- Bridging can be configured only on the bundle interface.

Note BCP on MLPPP operates only in trunk mode.

Configuring BCP on MLPPP Trunk Mode

To configure BCP on MLPPP trunk mode, perform this task:

	Command	Purpose
Step 1	`Router(config)#`**`interface multilink`**	Selects the multilink interface.
Step 2	`Router(config-if)#`**`switchport`**	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 3	`Router(config-if)#`**`switchport trunk allowed vlan`** *`vlan-list`*	By default, no VLANs are allowed. Use this command to explicitly allow VLANs; valid values for *vlan-list* are from 1 to 4094.

	Command	Purpose
Step 4	`Router(config-if)#`**`switchport mode trunk`**	Configures the router port connected to the switch as a VLAN trunk port.
Step 5	`Router(config-if)#`**`switchport nonegotiate`**	Puts the LAN port into permanent trunking mode but prevents the port from generating DTP frames
Step 6	`Router(config-if)#`**`no ip address`**	Unassigns the IP address.
Step 7	`Router(config-if)#`**`ppp multilink`**	Enables this interface to support MLP.
Step 8	`Router(config-if)#`**`multilink-group 1`**	Assigns this interface to the multilink group.
Step 9	`Router(config-if)#`**`interface Serial1/0/0.1/1/1/1:0`**	Designates a serial interface as a multilink bundle.
Step 10	`Router(config-if)#`**`no ip address`**	Unassigns the IP address.
Step 11	`Router(config-if)#`**`encapsulation ppp`**	Enables PPP encapsulation.
Step 12	`Router(config-if)#`**`ppp multilink`**	Enables this interface to support MLP.
Step 13	`Router(config-if)#`**`multilink-group 1`**	Assigns this interface to the multilink group 1.
Step 14	`Router(config-if)#`**`interface Serial1/0/0.1/1/1/2:0`**	Designates a serial interface as a multilink bundle.
Step 15	`Router(config-if)#`**`no ip address`**	Unassigns the IP address.
Step 16	`Router(config-if)#`**`encapsulation ppp`**	Enables PPP encapsulation.
Step 17	`Router(config-if)#`**`ppp multilink`**	Enables this interface to support MLP.
Step 18	`Router(config-if)#`**`multilink-group 2`**	Assigns this interface to the multilink group 2.
Step 19	`Router(config-if)#`**`shutdown`**	Shuts down an interface.
Step 20	`Router(config-if)#`**`no shutdown`**	Reopens an interface.
Step 21	`Router(config-if)#`**`switchport trunk allowed vlan`** *`vlan-list`*	By default, no VLANs are allowed. Use this command to explicitly allow VLANs; valid values for *vlan-list* are from 1 to 4094.

Verifying BCP on MLPPP Trunk Mode

To display information about Multilink PPP, perform this task in EXEC mode.

Command	Purpose
`Router(config-if)#` **`show ppp multilink`**	Displays information on a multilink group.

The following example provides sample output of the **show ppp multilink** command:

```
Router# show ppp multilink

Multilink1, bundle name is group 1
Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links: 4 active, 0 inactive (max no set, min not set)
Serial1/0/0/:1
Serial1/0/0/:2
Serial1/0/0/:3
Serial1/0/0/:4
```

Link Fragmentation and Interleaving (LFI) Guidelines

LFI can function by using either FRF.12 or MLPPP. MLPPP LFI operates in both hardware and software while FRF.12 LFI operates only in hardware.

Hardware MLPPP LFI Guidelines

LFI using MLPPP will function only in hardware if there is just one member link in the MLPPP bundle. The link can be a fractional T1 or full T1. Note the following guidelines:

- The **ppp multilink interleave** command must be configured to enable interleaving.
- Only three fragmentation sizes are supported: 128 bytes, 256 bytes, and 512 bytes.
- Fragmentation is enabled by default, and the default size is 512 bytes.
- A policy map having a priority class must be applied to the main interface.
- When hardware-based LFI is enabled, fragmentation counters are not displayed.

FRF.12 LFI Guidelines

LFI using FRF.12 is always performed in hardware. Note the following guidelines:

- The fragmentation is configured at the main interface.
- Only three fragmentation sizes are supported: 128 bytes, 256 bytes, and 512 bytes.
- A policy map having a priority class must be applied to the main interface.

Configuring QoS Features on Serial SPAs

For information about the QoS features supported by the serial SPAs, see the "Configuring QoS Features on a SIP" section on page 4-33 of Chapter 4, "Configuring the SIPs and SSC."

Saving the Configuration

To save your running configuration to nonvolatile random-access memory (NVRAM), perform this task in privileged EXEC configuration mode:

Command	Purpose
`Router# copy running-config startup-config`	Writes the new configuration to NVRAM.

For more information about managing configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

Verifying the Interface Configuration

Besides using the **show running-configuration** command to display your Catalyst 6500 Series switch configuration settings, you can use the **show interfaces serial** and the **show controllers serial** commands to get detailed information on a per-port basis for your 2-Port and 4-Port Clear Channel T3/E3 SPA.

Verifying Per-Port Interface Status

To find detailed interface information on a per-port basis for the 2-Port and 4-Port Channelized T3 SPA, use the **show interfaces serial** command. For a description of the command output, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication.

The following example provides sample output for the serial interface on an unchannelized T3:

```
Router# show interface serial3/0/0
Serial3/0/0 is down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
              0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 applique, 4 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions alarm present
  DSU mode 0, bandwidth 44210 Kbit, scramble 1, VC 0
```

The following example provides sample output for the serial interface on a channelized T3:

```
Router# show interface serial3/0/1/1:0
Serial3/0/1/1:0 is down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 1500 bytes, BW 832 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions alarm present
  VC 1: timeslot(s): 2-14, Transmitter delay 0, non-inverted data
```

To find detailed status and statistical information on a per-port basis for the 2-Port and 4-Port Clear Channel T3/E3 SPA, use the **show controllers serial** command. For a description of the command output, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication.

The following example provides sample controller statistics for the third port on the SPA located in the first subslot of the SIP-200 that is installed in slot 5 of a Catalyst 6509 switch:

```
Router# show controller serial 5/0/2
Serial5/0/2 -
   Framing is c-bit, Clock Source is Line
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 0, since reset 0
   Data in current interval (807 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 306 Unavailable Secs
     500 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     564 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 2:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     564 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 3:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     562 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 4:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     560 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
    .
    .
    .
    Total Data (last 44 15 minute intervals):
         0 Line Code Violations, 0 P-bit Coding Violation,
         0 C-bit Coding Violation,
         0 P-bit Err Secs, 0 P-bit Sev Err Secs,
         0 Sev Err Framing Secs, 0 Unavailable Secs,
         24750 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs

      Transmitter is sending AIS.

      Receiver has loss of signal.

       40434 Sev Err Line Secs, 0 Far-End Err Secs, 0 Far-End Sev Err Secs
       0 P-bit Unavailable Secs, 0 CP-bit Unavailable Secs
```

```
        0 CP-bit Far-end Unavailable Secs
        0 Near-end path failures, 0 Far-end path failures

       No FEAC code is being received
      MDL transmission is disabled
```

Configuration Examples

This section includes the following configuration examples:

- DSU Configuration Example, page 17-26
- MDL Configuration Example, page 17-26
- Encapsulation Configuration Example, page 17-27
- Framing—Unchannelized Mode Configuration Example, page 17-27
- Facility Data Link Configuration Example, page 17-27
- Scrambling Configuration Example, page 17-27
- Creating a Multilink Bundle Configuration Example, page 17-28
- Assigning a T1 Interface to a Multilink Bundle Configuration Example, page 17-28

DSU Configuration Example

The following example sets the DSU mode on interface port 0 on slot 4, subslot 1:

```
! Specify the interface and enter interface configuration mode.
!
Router(config-int)# interface t3 4/1/0
!
!Specifies the interoperability mode used by the T3 interface.
!
Router(config-int)# dsu mode 2
!
!Specifies the maximum allowable bandwidth.

Router(config-int)# dsu bandwidth 23000
```

MDL Configuration Example

The following example configures the MDL strings on controller port 0 on slot 4, subslot 1:

```
! Enter controller configuration mode.
!
Router(config)# controller t3 4/1/0
!
! Specify the mdl strings.
!
Router(config-controller)# mdl string eic beic
Router(config-controller)# mdl string lic beic
Router(config-controller)# mdl string fic bfix
Router(config-controller)# mdl string unit bunit
Router(config-controller)# mdl string pfi bpfi
Router(config-controller)# mdl string port bport
Router(config-controller)# mdl string generator bgen
Router(config-controller)# mdl transmit path
Router(config-controller)# mdl transmit idle-signal
Router(config-controller)# mdl transmit test-signal
```

Encapsulation Configuration Example

The following example configures encapsulation on a channelized T1 interface:

```
! Specify the interface to configure and enter interface configuration mode.
!
Router(config)# interface serial 4/1/1/1:0
!
! Specify the encapsulation method.
!
Router(config-if)# encapsulation ppp
```

The following example configures encapsulation and framing on a unchannelized T3 interface:

```
! Specify the interface to configure and enter interface configuration mode.
!
Router(config)# interface serial 4/1/1
!
! Specify the encapsulation method.
!
Router(config-if)# encapsulation ppp
```

Framing—Unchannelized Mode Configuration Example

The following example configures framing on an unchannelized T3 interface:

```
! Specify the interface to configure and enter interface configuration mode.
!
Router(config)# interface serial 4/1/1
!
! Specify the framing type.
!
Router(config-if)# framing m13
```

Facility Data Link Configuration Example

The following example configures FDL on a channelized T1 interface:

```
! Specify the controller to configure and enter controller configuration mode.
!
Router(config)# controller t3 3/1/0
!
! Specify the T1 controller and set the FDL bit.
!
Router(config-controller)# t1 1 fdl ansi
```

Scrambling Configuration Example

The following example configures scrambling on the T3 interface:

```
! Enter global configuration mode.
!
Router# configure terminal
!
```

```
! Specify the interface to configure and enter interface configuration mode.
!
Router(config)# interface serial 4/1/3
!
! Enable scrambling.
!
Router(config-if)# scrambling
```

Creating a Multilink Bundle Configuration Example

The following example creates a multilink bundle and assigns an IP address:

```
! ! Enter global configuration mode.
!
Router# configure terminal
!
! Create a multilink interface and enter interface configuration mode.
!
Router(config)# interface multilink 1
!
! Specify the IP address for the interface.
!
Router(config-if)# ip address 123.345.678.21 255.255.255.0
!
```

Assigning a T1 Interface to a Multilink Bundle Configuration Example

The following example assigns a T1 interface to a multilink bundle:

```
! ! Enter global configuration mode.
!
Router# configure terminal
!
! Specify the T1 interface and enter interface configuration mode.
!
Router(config)# interface serial 1/0/1/1:0
!
! Specify PPP encapsulation.
!
Router(config-if)# encapsulation ppp
!
! Specify the multilink bundle the T1 will belong to.
!
Router(config-if)# multilink-group 1
!
```

CHAPTER 18

Configuring the 1-Port Channelized OC-3/STM-1 SPA

This chapter provides information about configuring the 1-Port Channelized OC-3/STM-1 SPA on the Catalyst 6500 Series switch. It includes the following sections:

- Configuration Tasks, page 18-1
- Verifying the Interface Configuration, page 18-34

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX.* Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Configuration Tasks

This section describes how to configure the 1-Port Channelized OC-3/STM-1 SPA for the Catalyst 6500 Series switch and includes information about verifying the configuration. This document shows how to configure the 1-Port Channelized OC-3/STM-1 SPA in either Synchronous Optical Networking (SONET) or Synchronous Digital Hierarchy (SDH) framing modes.

It includes the following topics:

- Required Configuration Tasks, page 18-1
- Optional Configurations, page 18-18
- Saving the Configuration, page 18-33

Required Configuration Tasks

This section lists the required configuration steps to configure the 1-Port Channelized OC-3/STM-1 SPA. Some of the required configuration commands implement default values that might be appropriate for your network. If the default value is correct for your network, then you do not need to configure the command.

- Selecting the Physical Port and Controller, page 18-2

- Configuring for SONET Framing, page 18-3
- Configuring for SDH Framing, page 18-5
- Configuring Channels, page 18-6
- Serial Interface Naming, page 18-18
- Optional Configurations, page 18-18

Selecting the Physical Port and Controller

To configure or monitor the 1-Port Channelized OC-3/STM-1 SPA, you must specify the physical location of the SIP, SPA, and interface in the configuration commands. To select the physical port and controller, use the following command in configuration mode:

```
Router(config)# controller sonet slot/subslot/port
```

where:

- *slot*—Specifies the chassis slot number in the Catalyst 6500 Series switch where the SIP is installed.
- *subslot*—Specifies the secondary slot of the SIP where the SPA is installed.
- *port*—Specifies the number of the individual interface port on a SPA. Since there is only 1 port on the 1-Port Channelized OC-3/STM-1 SPA, the port number is always 0.

The following example shows how to specify the port of a 1-Port Channelized OC-3/STM-1 SPA installed in subslot 1 of a Cisco 7600 SIP-200 in slot 3:

```
Router(config)# controller sonet 3/1/0
```

For more information about identifying slots and subslots, see the "Identifying Slots and Subslots for SIPs, SSCs, and SPAs" section on page 4-2.

To configure the interface for the 1-Port Channelized OC-3/STM-1 SPA, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# controller sonet slot/subslot/port`	Selects the controller to configure and enters controller configuration mode. • *slot/subslot/port*—Specifies the location of the interface. **Note** On the 1-Port Channelized OC-3/STM-1 SPA, the port number is always 0.
	To continue the configuration using SONET framing, perform the configuration described in the "Configuring for SONET Framing" section on page 18-3. To continue the configuration using SDH framing, perform the configuration described in the "Configuring for SDH Framing" section on page 18-5.	

Configuring for SONET Framing

To configure the 1-Port Channelized OC-3/STM-1 SPA for SONET framing, perform this task:

	Command	Purpose
Step 1	`Router(config-controller)# framing sonet`	Specifies SONET as the frame type. This is the default.
Step 2	`Router(config-controller)# clock source {internal | line}`	(Optional) Sets the clock source. • **internal**—Specifies that the internal clock source is used. • **line**—Specifies that the network clock source is used. This is the default. **Note** The clock source must be set to internal if the opposite end of the connection is set to line and the clock source must be set to line if the opposite end of the connection is set to internal.
Step 3	`Router(config-controller)# [no] loopback {local | network}`	(Optional) Enables or disables loopback mode on a SONET controller. • **local**—Loops data from the transmit path to the receive path. • **network**—Loops data received on the external port to the transmit path and back out the external port. By default, loopback is disabled.
Step 4	`Router(config-controller)# [no] ais-shut`	(Optional) By default, a Line Alarm Indication Signal (LAIS) is sent to the far end when a port is shut down. The **no** option disables sending LAIS when the port is administratively shut down.
Step 5	`Router(config-controller)# [no] idle pattern 0-255`	(Optional) Sets the data to be written to the idle (disabled, or unprovisioned) time-slots of a channelized path. The default idle pattern is 127 (hex 7F).
Step 6	`Router(config-controller)# [no] ber-threshold {b1-tca | b2-tca | b3-tca | sd-ber | sf-ber} exponent`	(Optional) Sets bit error rate (BER) thresholds. The BER is 10 to the negative *exponent*. These are the thresholds: • **b1-tca**—B1 BER threshold crossing alarm. • **b2-tca**—B2 BER threshold crossing alarm. • **b3-tca**—B3 BER threshold crossing alarm, applied to all channels. • **sd-ber**—Sets Signal Degrade BER threshold. • **sf-ber**—Sets Signal Fail BER threshold. The *exponent* range is 3 to 9 for all except Signal Degrade BER, which has a range of 5 to 9.
Step 7	`Router(config-controller)# sts-1 sts1#`	Selects the SONET STS-1 level to configure. For the 1-Port Channelized OC-3/STM-1 SPA, the range of *sts1#* is 1 to 3.

	Command	Purpose
Step 8	`Router(config-ctrlr-sts1)# [no] mode {vt-15 | ct3 | ct3-e1 | t3}`	Specifies the mode of operation of a STS-1 path: • **vt-15**—The STS-1 is divided into seven Virtual Tributary Groups (VTG). Each VTG is then divided into four VT1.5 channels, each carrying a DS1. • **ct3**—The STS-1 carries a DS3 signal divided into 28 DS1 channels (PDH). • **ct3-e1**—The STS-1 carries a DS3 signal divided into 21 E1 channels (PDH). • **t3**—The STS-1 carries an unchannelized (clear channel) DS3. SONET framing does not support E3 modes.
		If you select **mode vt-15**, perform the configuration described in the "Configuring DS1 (Channelized T3 mode)" section on page 18-8**.** If you select **mode ct3**, perform the configuration described in the "Configuring Channelized DS3" section on page 18-6**.** If you select **mode ct3-e1**, perform the configuration described in the "Configuring E1 (SONET Channelized T3-E1 mode)" section on page 18-11**.** If you select **mode t3**, perform the configuration described in the "Configuring an Unchannelized DS3 Serial Interface" section on page 18-13**.**
	Repeat from Step 7 for each SONET STS-1 level.	

Configuring for SDH Framing

To configure the 1-Port Channelized OC-3/STM-1 SPA for SDH framing, perform this task:

	Command	Purpose
Step 1	Router(config-controller)# **framing sdh**	Specifies SDH as the frame type.
Step 2	Router(config-controller)# **aug mapping** {**au-3** | **au-4**}	Configures administration units group (AUG) mapping. This command is available only when SDH framing is configured. • **au-3**—The following muxing/alignment/mapping will be used: VC-3 <--> AU-3 <--> AUG • **au-4**—The following muxing/alignment/mapping will be used: TUG-3 <--> VC-4 <--> AU-4 <--> AUG The default is **au-4.**
Step 3	Router(config-controller)# **au-3** *au-3#*	If you selected AUG mapping as **au-3**, you can further specify AU-3 muxing. The CLI command parser will enter into config-ctrlr-au3 parser mode, which will make only relevant commands visible. The *au-3#* range is from 1 to 3.
	Router(config-controller)# **au-4 1 tug-3** *tug-3#*	If you selected AUG mapping as **au-4**, you can further specify TUG-3 muxing. The CLI command parser will enter into config-ctrlr-tug3 parser mode, which will make only relevant commands visible. The *tug-3#* range is from 1 to 3.
Step 4	Router(config-ctrlr-au3)# [**no**] **mode c-11**	If you selected AUG mapping as **au-3**, you can specify c-11 mode, a container level-n channelized DS3, subdivided into 28 DS1 channels. In this mode, the AU-3 is divided into seven TUG-2 channels. Each TUG-2 is then divided into four TU-11 channels, each carrying a DS1 channel.
	Router(config-ctrlr-tug3)# [**no**] **mode** {**c-12** | **t3** | **e3**}	If you selected AUG mapping as **au-4**, you can specify the following modes: • **c-12**—A container level-n channelized DS3, subdivided into 21 E1 channels. The AU-4/TUG-3 is divided into seven TUG-2 channels. Each TUG-2 is then divided into three TU-12 channels, each carrying an E1 channel. • **t3**—The AU-4/TUG-3 carries an unchannelized (clear channel) DS3 channel. • **e3**—The AU-4/TUG-3 carries a unchannelized (clear channel) E3 channel.

Command	Purpose
	If you select **mode c-11**, perform the configuration described in the "Configuring DS1 (Channelized T3 mode)" section on page 18-8. If you select **mode c-12**, perform the configuration described in the "Configuring E1 (SDH Channelized T3/E3 mode)" section on page 18-12. If you select **mode t3**, perform the configuration described in the "Configuring an Unchannelized DS3 Serial Interface" section on page 18-13. If you select **mode e3**, perform the configuration described in the "Configuring an Unchannelized E3 Serial Interface" section on page 18-16.

Repeat from Step 3 for each AU-3 or TUG-3 group.

Note If you configure an AU-3 or TUG-3 group as mode c-11 or t3, you cannot use modes c-12 or e3 for the other AU-3 or TUG-3 groups.

Configuring Channels

Depending on the framing and channel modes you have selected, use the appropriate section for the configuration of the channels.

- Configuring Channelized DS3, page 18-6
- Configuring DS1 (Channelized T3 mode), page 18-8
- Configuring E1 (SONET Channelized T3-E1 mode), page 18-11
- Configuring E1 (SDH Channelized T3/E3 mode), page 18-12
- Configuring an Unchannelized DS3 Serial Interface, page 18-13
- Configuring an Unchannelized E3 Serial Interface, page 18-16

Configuring Channelized DS3

To configure channelized DS3, you must have selected SONET framing mode ct3.

Channelized DS3 can be configured in either config-ctrlr-sts1 or config-ctrlr-au3 parser mode. To configure channelized DS3 mode, perform this task:

	Command	Purpose
Step 1	Router(config-ctrlr-*xxx*[1])# [**no**] **t3 framing** {**c-bit** | **m23** | **auto-detect**}	(Optional) Specifies the T3 framing mode. • **c-bit**—Use C-bit parity framing. • **m23**—Use M23 framing. • **auto-detect—**Detects the framing type at the device at the end of the line and switches to that framing type. If both devices are set to auto-detect, c-bit framing is used.
Step 2	Router(config-ctrlr-*xxx*)# [**no**] **t3 clock source** {**internal** | **line**}	(Optional) Sets the clock source. • **internal**—Specifies that the internal clock source is used. • **line**—Specifies that the network clock source is used. This is the default for T1 and E1. **Note** The clock source must be set to internal if the opposite end of the connection is set to line and the clock source must be set to line if the opposite end of the connection is set to internal.
Step 3	Router(config-ctrlr-*xxx*)# [**no**] **t3 loopback** {**local** | **network** [**line** | **payload**] | **remote** [**line** | **payload**]}	(Optional) Enables or disables loopback mode on a SONET controller. These are the supported loopback modes: • **local**—Loops data from the transmit path to the receive path. • **network**—Loops all data or only payload data received on the external port to the transmit path and back out the external port. • **remote**—Sends Far End Alarm and Control (FEAC) to set remote system in either **line** or **payload** loopback. Applicable only to C-bit framing. The default is no loopback.

	Command	Purpose
Step 4	Router(config-ctrlr-*xxx*)# [**no**] **t3 mdl string** {**eic** | **fic** | **generator** | **lic** | **pfi** | **port** | **unit**} *string*	(Optional) Maintenance Data Link (MDL) messages are used to communicate identification information between local and remote ports. Configures the contents of the MDL message. • **eic**—Specifies the Equipment Identification Code; can be up to 10 characters. • **fic**—Specifies the Frame Identification Code; can be up to 10 characters. • **generator**—Specifies the Generator number string sent in the MDL Test Signal message; can be up to 38 characters. • **lic**— Specifies the Location Identification Code; can be up to 11 characters. • **pfi**—Specifies the Path Facility Identification Code sent in the MDL Path message; can be up to 38 characters. • **port**—Specifies the port number string sent in the MDL Idle Signal message; can be up to 38 characters. • **unit**—Specifies the Unit Identification Code; can be up to 6 characters. The default is no MDL string.
Step 5	Router(config-ctrlr-*xxx*)# [**no**] **t3 mdl transmit** {**path** | **idle-signal** | **test-signal**}	(Optional) Configures the transmission of the MDL message. • **path**—Enables MDL Path message transmission. • **idle-signal**—Enables MDL Idle-Signal message transmission • **test-signal**—Enables MDL Test-Signal message transmission. The default is no MDL transmit.
Step 6	Router(config-ctrlr-*xxx*)# [**no**] **t3 equipment** {**customer** | **network**} **loopback**	(Optional) Determines response to remote loopback request. • **customer**—Enables the port to honor remote loopback requests. • **network**—Disables remote loopback requests. **Note** Remote loopbacks are only available in c-bit framing mode.

1. The actual command prompt is `Router(config-ctrlr-sts1)#` for SONET and `Router(config-ctrlr-au3)#` for SDH.

Configuring DS1 (Channelized T3 mode)

Two modes of operation support the DS1 channel configuration shown in this section.

With SONET framing mode VT-15, the STS-1 channel is divided into seven Virtual Tributary Groups (VTG). Each VTG contains four VT1.5 channels, each of which carries a DS1 channel. When configuring the DS1 channels, the following substitutions should be made in the commands shown:

- The command prompt is `Router(config-ctrlr-sts1)#`

- The command prefix is **vtg** *vtg-number*

  The *vtg-number* selects which VTG is being configured. The range is 1 to 7.

With SDH framing mode AU-3/C-11, the AU-3 channel is divided into seven TUG-2 channels. Each TUG-2 channel is then divided into four TU-11 channels, each carrying a DS1 channel. When configuring the DS1 channels, the following substitutions should be made in the commands shown:

- The command prompt is `Router(config-ctrlr-au3)`
- The command prefix is **tug-2** *tug-2-number*

  The *tug-2-number* selects which TUG-2 channel is being configured. The range is 1 to 7.

In configuring each DS1 channel, specify the DS1 channel number *t1#* (range 1 to 4). To configure a DS1 channel, perform this task, substituting the appropriate command prefix for *prefix* in the commands:

	Command	Purpose

Step 1	`Router(config-ctrlr-`*xxx*[1]`)#`
	[**no**] *prefix*[2] **t1** *t1#* **clock source** {**internal** | **line**}
		(Optional) Sets the clock source.
		• **internal**—Specifies that the internal clock source is used.
		• **line**—Specifies that the network clock source is used. This is the default.
		Note The clock source must be set to internal if the opposite end of the connection is set to line and the clock source must be set to line if the opposite end of the connection is set to internal.

Step 2	`Router(config-ctrlr-`*xxx*`)#`
	[**no**] *prefix* **t1** *t1#* **framing** {**sf** | **esf**}
		(Optional) Specifies the T1 framing type using the **framing** command.
		• **sf**—Specifies Super Frame as the T1 frame type.
		• **esf**—Specifies Extended Super Frame as the T1 frame type. This is the default.
		Note If you select sf framing, consider disabling yellow alarm detection because the yellow alarm can be incorrectly detected with sf framing.

Step 3	`Router(config-ctrlr-`*xxx*`)#`
	[**no**] *prefix* **t1** *t1#* **yellow** {**detection** | **generation**}
		(Optional) Enables detection or generation of DS1 yellow alarms,

	Command	Purpose
Step 4	Router(config-ctrlr-*xxx*)# *prefix* **t1** *t1#* **channel-group** *channel-number#* **timeslots** *range* [**speed** {**56** | **64**}]	Specifies the DS1 channel and timeslots to be mapped to each channel. • *channel-number*—Specifies a channel-group mapping (0–23) under the designated T1. • *range*—List of timeslots under the channel-group. Timeslots assigned to this T1 can be 1–24 or a combination of subranges within 1– 24. You can indicate a range using a hyphen, commas, or a combination of both. One timeslot equals one DS0. • **speed 56** or **64**— Specifies the speed of a timeslot as either 56 or 64 kbps. The default speed is 64 kbps.
	Router(config-ctrlr-*xxx*)# **no** *prefix* **t1** *t1#* **channel-group** *channel-number#*	To alter the configuration of an existing channel group, the channel group must be removed first using the **no** form of the **channel-group** command.
Step 5	Router(config-ctrlr-*xxx*)# [**no**] *prefix* **t1** *t1#* **fdl ansi**	(Optional) Enables the one-second transmission of remote performance reports via the Facility Data Link (FDL) per ANSI T1.403. This function requires that the T1 framing type is Extended Super Frame (ESF).
Step 6	Router(config-ctrlr-*xxx*)# [**no**] *prefix* **t1** *t1#* **loopback** {**local** | **network line** | **remote** {**line fdl** {**ansi** | **bellcore**} | **payload fdl ansi**}}	(Optional) Specifies the loopback mode for testing. • **local**—Loops data from the transmit path to the receive path. • **network line**—Loops data received on the external port to the transmit path and back out the external port before going through the T1 framer. • **remote line fdl**—Sends a repeating 16-bit ESF data link code word to the remote end requesting that it enter into a network line loopback. – For ansi, the code word is (00001110 11111111). – For bellcore, the code word is (00010010 11111111). • **remote payload fdl ansi**—Sends a repeating 16-bit ESF data link code word (00010100 11111111) to the remote end requesting that it enter into a network payload loopback. **Note** Local network payload loopback is not supported due to TEMUX-84/TEMUX-84E limitations.
Step 7	Router(config-ctrlr-*xxx*)# [**no**] *prefix* **t1** *t1#* **shutdown**	The **no** *args* **shutdown** command enables the interface. The *args* **shutdown** command disables the interface.
Step 8	Configure the serial interfaces. **Note** After a T1 channel is configured, it appears to the Cisco IOS software as a serial interface; therefore, all the configuration commands for a serial interface are available. However, not all commands are applicable to the T1 interface. For serial interface addressing information, refer to the "Serial Interface Naming" section on page 18-18.	

1. The actual command prompt is Router(config-ctrlr-sts1)# for SONET and Router(config-ctrlr-au3)# for SDH.
2. The actual command prefix is **vtg** *vtg-number* for SONET and **tug-2** *tug-2-number* for SDH.

Configuring E1 (SONET Channelized T3-E1 mode)

For E1 channel configuration in SONET channelized DS3 mode, you must have previously selected SONET framing mode ct3-e1. In this mode, the STS-1 is divided into 21 E1 channels.

In configuring each E1 channel, specif the E1 channel number *e1#* (range 1-21). To configure each E1, perform this task:

	Command	Purpose

Step 1	`Router(config-ctrlr-sts1)#`
	`e1` *`e1#`* **`channel-group`** *`channel-group#`* **`timeslots`** *`list-of-timeslots`* `[`**`speed`** `{`**`56`** `|` **`64`**`}]`
		Specifies the E1 channel and timeslots to be mapped to each channel.
		• *channel-number*—Specifies a channel-group mapping (0–30) under the designated E1.
		• *range*—List of timeslots under the channel-group. Timeslots assigned to this E1 can be 1–31 or a combination of subranges within 1–31. You can indicate a range using a hyphen, commas, or a combination of both. One timeslot equals one DS0.
		• **speed 56** or **64**— Specifies the speed of a timeslot as either 56 or 64 kbps. The default speed is 64 kbps.

	`Router(config-ctrlr-sts1)#`
	`no e1` *`e1#`* **`channel-group`** *`channel-group#`*
		To alter the configuration of an existing channel group, the channel group must be removed first using the **no** form of the **channel-group** command.

Step 2	`Router(config-ctrlr-sts1)#`
	`[`**`no`**`]` **`e1`** *`e1#`* **`unframed`** `|` **`framing`** `{`**`crc4`** `|` **`no-crc4`**`}`
		(Optional) Sets the framing on the interface.
		• **unframed**—Unframed mode (G.703) uses all 32 time slots for data, none for framing signals.
		• **framing**—Uses a time slot for framing signals.
		– **crc4**—Specifies CRC4 as the E1 frame type.
		– **no-crc4**—Specifies no CRC4 as the E1 frame type.
		The default is framing with crc4.

Step 3	`Router(config-ctrlr-sts1)#`
	`[`**`no`**`]` **`e1`** *`e1#`* **`clock source`** `{`**`internal`** `|` **`line`**`}`
		(Optional) Sets the clock source.
		• **internal**—Specifies that the internal clock source is used.
		• **line**—Specifies that the network clock source is used. This is the default.
		Note The clock source must be set to internal if the opposite end of the connection is set to line and the clock source must be set to line if the opposite end of the connection is set to internal.

Step 4	`Router(config-ctrlr-sts1)#`
	`[`**`no`**`]` **`e1`** *`e1#`* **`national bits`** *`pattern`*
		(Optional) The national bit is reserved for national use and is set to 0 by default. Change this bit only when required for interoperability with your telephone company.

	Command	Purpose
Step 5	Router(config-ctrlr-sts1)# [**no**] **e1** *e1#* **loopback** [**local** | **network**]	(Optional) Enables or disables loopback mode on a serial port. These are the supported loopback modes: • **local**—Loops data from the transmit path to the receive path. • **network**—Loops data received on the external port to the transmit path and back out the external port. The default is no loopback.
Step 6	Router(config-ctrlr-sts1)# [**no**] **e1** *e1#* **shutdown**	The **no** *args* **shutdown** command enables the interface. The *args* **shutdown** command disables the interface.

Step 7 Configure the serial interfaces.

Note After an E1 channel is configured, it appears to the Cisco IOS software as a serial interface; therefore, all the configuration commands for a serial interface are available. However, not all commands are applicable to the E1 interface. For serial interface addressing information, refer to the "Serial Interface Naming" section on page 18-18.

Configuring E1 (SDH Channelized T3/E3 mode)

For E1 channel configuration in SDH channelized DS3 mode, you must have previously selected SDH framing mode with AU-4 mode C-12. In this mode, the AU-4/TUG-3 is divided into seven TUG-2 channels. Each TUG-2 channel is then divided into three TU-12 channels, each carrying an E1 channel.

In configuring each E1 channel, specify the TUG-2 channel number *tug-2#* (range 1–7) and the E1 channel number *e1#* (range 1–3). To configure each E1, perform this task:

	Command	Purpose
Step 1	Router(config-ctrlr-tug3)# **tug-2** *tug-2#* **e1** *e1#* **channel-group** *channel-group#* **timeslots** *list-of-timeslots* [**speed** {**56** | **64**}]	Specifies the E1 channel and timeslots to be mapped to each channel. • *channel-number*—Specifies a channel-group mapping (0–30) under the designated E1. • *range*—List of timeslots under the channel group. Timeslots assigned to this E1 can be 1–31 or a combination of subranges within 1–31. You can indicate a range using a hyphen, commas, or a combination of both. One timeslot equals one DS0. • **speed 56** or **64**— Specifies the speed of a timeslot as either 56 or 64 kbps. The default speed is 64 kbps.
	Router(config-ctrlr-tug3)# **no tug-2** *tug-2#* **e1** *e1#* **channel-group** *channel-group#*	To alter the configuration of an existing channel group, the channel group must be removed first using the **no** form of the **channel-group** command.

	Command	Purpose
Step 2	Router(config-ctrlr-tug3)# [**no**] **tug-2** *tug-2#* **e1** *e1#* **unframed** | **framing** {**crc4** | **no-crc4**}	(Optional) Sets the framing on the interface. • **unframed**—Unframed mode (G.703) uses all 32 time slots for data, none for framing signals. • **framing**—Uses a time slot for framing signals. – **crc4**—Specifies CRC4 as the E1 frame type. – **no-crc4**—Specifies no CRC4 as the E1 frame type. The default is framing with crc4.
Step 3	Router(config-ctrlr-tug3)# [**no**] **tug-2** *tug-2#* **e1** *e1#* **clock source** {**internal** | **line**}	(Optional) Sets the clock source. • **internal**—Specifies that the internal clock source is used. • **line**—Specifies that the network clock source is used. This is the default. **Note** The clock source must be set to internal if the opposite end of the connection is set to line and the clock source must be set to line if the opposite end of the connection is set to internal.
Step 4	Router(config-ctrlr-tug3)# [**no**] **tug-2** *tug-2#* **e1** *e1#* **national bits** *pattern*	(Optional) The national bit is reserved for national use and is set to 0 by default. Change this bit only when required for interoperability with your telephone company.
Step 5	Router(config-ctrlr-tug3)# [**no**] **tug-2** *tug-2#* **e1** *e1#* **loopback** [**local** | **network**]	(Optional) Enables or disables loopback mode on a serial port. These are the supported loopback modes: • **local**—Loops data from the transmit path to the receive path. • **network**—Loops data received on the external port to the transmit path and back out the external port. The default is no loopback.
Step 6	Router(config-ctrlr-tug3)# [**no**] **tug-2** *tug-2#* **e1** *e1#* **shutdown**	The **no** *args* **shutdown** command enables the interface. The *args* **shutdown** command disables the interface.
Step 7	Configure the serial interfaces. **Note** After an E1 channel is configured, it appears to the Cisco IOS software as a serial interface; therefore, all the configuration commands for a serial interface are available. However, not all commands are applicable to the E1 interface. For serial interface addressing information, refer to the "Serial Interface Naming" section on page 18-18.	

Configuring an Unchannelized DS3 Serial Interface

Two modes of operation support the unchannelized DS3 configuration shown in this section:

- With SONET framing mode t3, an STS-1 channel carries a DS3 channel.
- With SDH framing mode AU-4 mode t3, an AU-4/TUG3 channel carries a DS3 channel.

To configure a DS3 serial interface, perform this task:

	Command	Purpose
Step 1	Router(config-ctrlr-*xxx*[1])# **end**	Exits SONET or SDH framing mode configuration.
Step 2	Router(config)# **interface serial** *slot/subslot/port.au-4/tug-3* or **interface serial** *slot/subslot/port.sts1*	Selects the T3 serial interface to configure. For serial interface addressing information, refer to the "Serial Interface Naming" section on page 18-18.
Step 3	Router(config-if)# [**no**] **dsu mode** {**0** | **1** | **2** | **3** | **4**}	Specifies the interoperability mode used by a DS3 controller. • **0**—Connects to another DS3 controller or to a Digital Link DSU (DL3100 in T3 mode). This is the default. • **1**—Connects to a Kentrox DataSMART DS3 IDSU. • **2**—Connects to a Larscom Access-T45 DS3 DSU. • **3**—Connects to an Adtran T3SU 300. • **4**—Connects to a Verilink HDM 2182. The default is **0** (Digital Link).

	Command	Purpose
Step 4	Router(config-if)# [**no**] **dsu bandwidth** *kbps*	(Optional) Specifies the allowable bandwidth in *kbps*. The default for DS3 mode is 44010 kbps. The bandwidth range and increment values are based on the specific DSU, as follows: • Digital Link DL3100 – range: 300 to 44210 kbps – increments: 300 kbps • Digital Link DL3100E – range: 358 to 34010 kbps – increments: 358 kbps • Kentrox DataSMART T3/E3 IDSU – range: 1000 to 34000 kbps (E3 mode) – range: 1500 to 44210 kbps (T3 mode) – increments: 500 kbps • Larscom Access-T45 DS3 – range: 3100 to 44210 kbps – increments: 3100 kbps • Adtran T3SU 300 – range: 80 to 44210 kbps – increments: 80 kbps • Verilink HDM 2182 – range: 1600 to 31600 kbps – increments: 1600 kbps
Step 5	Router(config-if)# [**no**] **scramble**	(Optional) Scrambling randomizes the pattern of ones and zeros carried in the physical layer frame in order to assist clock recovery on the receiving end. The default is no scramble.
Step 6	Router(config-if)# [**no**] **framing** {**c-bit** | **m13**}	(Optional) Specifies framing mode. • **c-bit**—Specifies C-bit parity framing. • **m13**—Specifies M13 framing. Unframed DS3 is not supported. Default is C-bit parity framing.
Step 7	Router(config-if)# [**no**] **dsu remote** {**fullrate** | **accept**}	(Optional) Specifies where the DSU bandwidth is set. • **fullrate**—Sets far end DSU to its full rate bandwidth. • **accept**—Accepts incoming remote requests to reset DSU bandwidth.
Step 8	Router(config-if)# [**no**] **crc** {**16** | **32**}	(Optional) Specifies CRC word size. Default is 16 bits (CRC-CITT).

	Command	Purpose
Step 9	`Router(config-if)# [no] loopback {local	network	dte	remote [line	payload]}`	(Optional) Specifies loopback. • **local**—Loops data from the transmit path to the receive path. • **network**—Loops data received on the external port to the transmit path and back out the external port. • **dte**—Loops back after the line interface unit (LIU) towards the terminal. • **remote**—Sends Far End Alarm and Control (FEAC) to set remote system in either **line** or **payload** loopback.
Step 10	`Router(config-if)# no shutdown`	The **no shutdown** command enables the interface. The **shutdown** command disables the interface.

1. The actual command prompt is `Router(config-ctrlr-sts1)#` for SONET and `Router(config-ctrlr-tug3)#` for SDH.

This example shows how to verify the controller configuration:

```
Router(config)# show controllers t1
T1 6/0/1 is up.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  No alarms detected.
blarm-trigger is not set
  Framing is ESF, Line Code is B8ZS, Clock Source is Line.
  Data in current interval (395 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Total Data (last 24 hours)
     0 Line Code Violations, 0 Path Code Violations,
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Configuring an Unchannelized E3 Serial Interface

For unchannelized E3 operation, you must have selected SDH framing mode with AU-4 mode e3, which specifies an E3 channel carried over a T3 channel. The configuration must be done in serial interface configuration mode.

To configure an unchannelized E3 serial interface, perform this task:

	Command	Purpose
Step 1	`Router(config-ctrlr-tug3)# end`	Exits SDH framing mode configuration.
Step 2	`Router(config)# interface serial` *slot/subslot/port.au-4/tug-3*	Selects the E3 serial interface to configure. For serial interface addressing information, refer to the "Serial Interface Naming" section on page 18-18.

	Command	Purpose
Step 3	`Router(config-if)# dsu mode {cisco	digital-link	kentrox}`	Specifies the interoperability mode. • **cisco**—Specifies Cisco as the dsu mode. • **digital-link**—Specifies Digital Link as the dsu mode. • **kentrox**—Specifies Kentrox as the dsu mode. Default is **cisco**.
Step 4	`Router(config-if)# [no] dsu bandwidth kbps`	(Optional) Specifies the maximum allowed bandwidth in *kbps*. The available range for each DSU type is: • Cisco—Range is 300-34010 kbps. • Digital Link—Range is 300-34010 kbps. • Kentrox—Range is 1000-24500, 34010 kbps.
Step 5	`Router(config-if)# [no] scramble`	(Optional) Scrambling randomizes the pattern of ones and zeros carried in the physical layer frame in order to assist clock recovery on the receiving end. The default is no scramble.
Step 6	`Router(config-if)# [no] national bit {0	1}`	(Optional) The national bit is reserved for national use and is set to 0 by default. Change this bit only when required for interoperability with your telephone company.
Step 7	`Router(config-if)# [no] framing {g751	g832}`	(Optional) Sets the framing on the interface. • **g751**—Specifies g751 framing. This is the default for E3. • **g832**—Specifies g832 framing.
Step 8	`Router(config-if)# [no] crc {16	32}`	(Optional) Specifies CRC word size. Default is 16 bits (CRC-CITT).
Step 9	`Router(config-if)# [no] loopback {network	local	remote}`	(Optional) Specifies loopback. • **local**—Loops data from the transmit path to the receive path. • **network**—Loops data received on the external port to the transmit path and back out the external port. • **remote**—Sends Far End Alarm and Control (FEAC) to set remote system in loopback.
Step 10	`Router(config-if)# no shutdown`	The **no shutdown** command enables the interface. The **shutdown** command disables the interface.

This example shows how to verify the controller configuration:

```
Router(config)# show controllers t1
T1 6/0/1 is up.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  No alarms detected.
blarm-trigger is not set
  Framing is ESF, Line Code is B8ZS, Clock Source is Line.
  Data in current interval (395 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Total Data (last 24 hours)
     0 Line Code Violations, 0 Path Code Violations,
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
          0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Serial Interface Naming

After you have configured the framing type and mode, serial interface names are automatically generated, and you can configure the serial interfaces using the **interface serial** command. The interface naming format will be dependent on the framing type and mode. The name formats of the serial interfaces created are listed below.

SONET framing

- If the mode is vt-15 (VTG groups channelized to DS1):

  interface serial [*slot/subslot/port*].[*sts1/vtg/t1*]:[*channel-group*]

- If the mode is ct3 (DS3 channelized to DS1):

  interface serial [*slot/subslot/port*].[*sts1/ds1*]:[*channel-group*]

  If the mode is ct3-e1 (DS3 channelized to E1):

  interface serial [*slot/subslot/port*].[*sts1/e1*]:[*channel-group*]

  If the mode is t3 (unchannelized DS3):

  interface serial [*slot/subslot/port*]:[*sts1*]

SDH framing

If the administration units group (AUG) mapping is au-4, the au-4 value is always 1; if the AUG mapping is au-3, then the only supported mode is c-11 (carrying a T1).

- If the mode is t3 or e3 (unchannelized DS3 or E3):

  interface serial [*slot/subslot/port*].[*au-4/tug-3*]

- If the mode is ct-12 mode (DS3 container level-n channelized into E1):

  interface serial [*slot/subslot/port*].[*tug-3/tug-2/e1*]:[*channel-group*]

- If the mode is c-11 mode (DS3 container level-n channelized into DS1):

  interface serial [*slot/subslot/port*].[*au-3/tug-2/t1*]:[*channel-group*]

Optional Configurations

There are several standard, but optional, configurations that might be necessary to complete the configuration of your serial SPA.

- Configuring Encapsulation, page 18-19
- Configuring the CRC Size, page 18-19
- Configuring FDL, page 18-20
- Configuring Distributed Multilink Point-to-Point Protocol (Hardware-Based), page 18-21
- Configuring MLFR, page 18-23
- Invert Data on the T1/E1 Interface, page 18-26
- Configuring Multipoint Bridging, page 18-27
- Configuring Bridging Control Protocol Support, page 18-27
- Link Fragmentation and Interleaving (LFI) Guidelines, page 18-27

- Hardware MLPPP LFI Guidelines, page 18-27
- FRF.12 LFI Guidelines, page 18-27
- Configuring QoS Features on Serial SPAs, page 18-28
- Configuring CRTP, page 18-28
- Configuring SONET and SDH Overhead Bytes, page 18-32
- Configuring a Bit Error Rate Test (BERT), page 18-33

Configuring Encapsulation

When traffic crosses a WAN link, the connection needs a Layer 2 protocol to encapsulate traffic. To set the encapsulation method, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial interface-name`	Selects the serial interface to configure. For addressing information, refer to the "Serial Interface Naming" section on page 18-18.
Step 3	`Router(config-if)# encapsulation {hdlc | ppp | frame-relay}`	Sets the encapsulation method on the interface. • **hdlc**—High-level Data Link Control (HDLC) protocol for serial interface. This encapsulation method provides the synchronous framing and error detection functions of HDLC without windowing or retransmission. This is the default for synchronous serial interfaces. • **ppp**—PPP (for serial interface). • **frame-relay**—Frame Relay (for serial interface).

Configuring the CRC Size

The 1-Port Channelized OC-3/STM-1 SPA interface uses a 16-bit cyclic redundancy check (CRC) by default, but also supports a 32-bit CRC. CRC is an error-checking technique that uses a calculated numeric value to detect errors in transmitted data. The designators 16 and 32 indicate the length (in bits) of the frame check sequence (FCS). A CRC of 32 bits provides more powerful error detection, but adds overhead. Both the sender and receiver must use the same setting.

CRC-16, the most widely used CRC throughout the United States and Europe, is used extensively with WANs. CRC-32 is specified by IEEE 802 and as an option by some point-to-point transmission standards. It is often used on Switched Multimegabit Data Service (SMDS) networks and LANs.

To set the length of the cyclic redundancy check (CRC) on an interface, use these commands:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface serial** *interface-name*	Selects the serial interface to configure. For addressing information, refer to the "Serial Interface Naming" section on page 18-18.
Step 3	Router(config-if)# **crc** {**16** | **32**}	Selects the CRC size in bits. • **16**—16-bit CRC. This is the default • **32**—32-bit CRC.

Configuring FDL

Facility Data Link (FDL) is a 4-kbps channel provided by the Extended Super Frame (ESF) T1 framing format. The FDL performs outside the payload capacity and allows you to check error statistics on terminating equipment without intrusion. To configure FDL, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface serial** *interface-name*	Selects the serial interface to configure. For addressing information, refer to the "Serial Interface Naming" section on page 18-18.
Step 3	Router(config-if)# [**no**] **t1** *t1#* **fdl** {**ansi** | **att**}	Enables the transmission of remote performance reports via the Facility Data Link (FDL). This function requires that the T1 framing type is Extended Super Frame (ESF). • **ansi**—Reports conform to the ANSI T1.403 protocol, and are sent at one-second intervals. • **att**—Reports conform to the AT&T TR54016 protocol (a subset of ANSI T1.403), and are sent only when a request has been received.

Verifying FDL

This example shows how to verify the **fdl** setting:

```
router# show controllers t1
```

```
T1 6/0/1 is up.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  No alarms detected.
  alarm-trigger is not set
  Framing is ESF, FDL is ansi, Line Code is B8ZS, Clock Source is Line.
  Data in current interval (742 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Total Data (last 73 15 minute intervals):
     1278491 Line Code Violations, 3 Path Code Violations,
     0 Slip Secs, 1 Fr Loss Secs, 177 Line Err Secs, 0 Degraded Mins,
     3 Errored Secs, 0 Bursty Err Secs, 1 Severely Err Secs, 227 Unavail Secs
.
.
.
```

Configuring Distributed Multilink Point-to-Point Protocol (Hardware-Based)

Distributed Multilink Point-to-Point Protocol (dMLPPP) allows you to combine interfaces which correspond to an entire T1 or E1 multilink bundle. You choose the number of bundles and the number of T1 or E1 lines in each bundle.

MLPPP Configuration Restrictions and Guidelines

The following restrictions and guidelines apply to hardware-based MLPPP:

- The MLPPP command is only available on serial interfaces.
- You must enable PPP encapsulation before configuring the MLPPP commands.
- The following are link restrictions for using MLPPP:
  - Only T1 or E1 links may be in a bundle.
  - All links must be on the same SPA.
  - A maximum of 12 links may be in a bundle.
- Only three fragmentation sizes are supported: 128, 256, and 512 bytes.
- Fragmentation is enabled by default, with a default size of 512 bytes.
- Fragmentation size is configured using the **ppp multilink fragment-delay** command after using the **interface multilink** command. Among the three possible fragmentationsizes, the least size satisfying the delay criteria is configured. For example, a 192 byte packet causes a delay of 1 millisecond on a T1 link, so the nearest fragmentation size is 128 bytes.

  Use the **show ppp multilink** command to indicate the MLPPP type and the fragmentation size:

  ```
  Router# show ppp multilink
  Multilink1, bundle name is Patriot2
  Bundle up for 00:00:13
  Bundle is Distributed
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received, 206/255 load
  0x0 received sequence, 0x0 sent sequence
  Member links: 2 active, 0 inactive (max not set, min not set)
  Se4/2/0/1:0, since 00:00:13, no frags rcvd
  Se4/2/0/2:0, since 00:00:10, no frags rcvd
  Distributed fragmentation on. Fragment size 512.  Multilink in Hardware.
  ```

- Fragmentation is disabled explicitly by using the **no ppp multilink fragmentation** command after using the **interface multilink** command.

Create a Multilink Bundle

To create a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# [no] interface multilink group-number`	Creates or configures a multilink interface and enters multilink interface mode. • *group-number*—The group number for the multilink bundle.
Step 3	`Router(config-if)# [no] multilink fragment [fragment-size size]`	(Optional) Enables fragmentation and sets the fragmentation size. • *size*—Allowed sizes are 128, 256 or 512 bytes. Fragmentation is disabled by default. When fragmentation is enabled, the default size is 128 bytes.
Step 4	`Router(config-if)# [no] multilink bundle-name {authenticated | endpoint | both}`	(Optional) Specifies the criteria for naming the multilink bundle, as defined in RFC 1990. • **authenticated**—Use the peer authenticated name as the bundle name. This is the default. • **endpoint**—Use the peer endpoint discriminator as the bundle name. • **both**—Use the peer authenticated name and endpoint discriminator as the bundle name.
Step 5	`Router(config-if)# [no] multilink min-links number [mandatory]`	(Optional) Specifies the preferred minimum number of links in a multilink bundle. • *number*—Sets the minimum number of links, from 0 to 255. • **mandatory**—If the number of links in the bundle falls below the number specified, the bundle is disabled. By default, the bundle goes down only when there are no links in the bundle.
	Configure each serial interface you want to assign to the multilink bundle.	

Assign an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters configuration mode.
Step 2	Router(config)# **interface serial** *interface-name*	Selects the interface to configure and enters interface configuration mode. For addressing information, refer to the "Serial Interface Naming" section on page 18-18.
Step 3	Router(config-if)# [**no**] **encapsulation ppp**	Enables PPP encapsulation.
Step 4	Router(config-if)# [**no**] **multilink-group** *group-number*	Assigns the interface to a multilink bundle. • *group-number*—The multilink group number for the T1 or E1 bundle.
Step 5	Router(config-if)# **ppp chap hostname** *name*	Specifies a pool of dialup routers that all appear to be the same host when authenticating with CHAP. **Note** This command is mandatory when there is more than one bundle across two routers; otherwise, it is optional.
Step 6	Router(config-if)# [**no**] **ppp multilink**	Enables the negotiation of multilink on an interface. This command is automatically implied for interfaces that are configured as part of a multilink bundle.
	Repeat these commands for each interface you want to assign to the multilink bundle.	

Verifying MLPPP

Use the **show ppp multilink** command to display a list of active and inactive bundles, the configured member links, and packet statistics information.

Use the **show interface multilink** *group-number* command to display multilink interface information and the LCP and multilink status.

Configuring MLFR

Multilink Frame Relay (MLFR) allows you to combine T1/E1 lines into a bundle that has the combined bandwidth of multiple T1/E1 lines. You choose the number of bundles and the number of T1/E1 lines in each bundle. This allows you to increase the bandwidth of your network links beyond that of a single T1/E1 line.

MLFR Configuration Guidelines

MLFR will function in hardware if all of the following conditions are met:

- Only T1 or E1 links may be in a bundle.
- All links must be on the same SPA.
- A maximum of 12 links may be in a bundle.

Create a Multilink Frame Relay Bundle

To create a multilink Frame Relay bundle, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters configuration mode.
Step 2	Router(config)# [**no**] **interface mfr** *number*	Creates or configures a multilink Frame Relay bundle interface. • *number*—The number for the Frame Relay bundle. The range is 0 to 2147483647.
Step 3	Router(config-if)# [**no**] **frame-relay multilink bid** *name*	(Optional) Assigns a bundle identification name to a multilink Frame Relay bundle. • *name*—The name for the Frame Relay bundle. **Note** The bundle identification (BID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shutdown** and **no shutdown** commands in interface configuration mode.
Step 4	Router(config-if)# [**no**] **frame-relay multilink bandwidth-class** *class* [*threshold*]	(Optional) Specifies FRF.16 *class* A, B, or C, configuring the trigger point for activating or deactivating a bundle. • **a**—Class A will bring up the bundle if at least one bundle link is active. • **b**—Class B will bring up the bundle only if all links are active, and will bring down the bundle if any link becomes inactive. • **c**—Class C wil bring the bundle up or down depending on the *threshold* number of links being active.

Assign an Interface to a Multilink Frame Relay Bundle

To assign an interface to a multilink Frame Relay bundle, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters configuration mode.
Step 2	Router(config)# **interface serial** *interface-name*	Selects the interface to configure and enters interface configuration mode. For addressing information, refer to the "Serial Interface Naming" section on page 18-18.

	Command	Purpose
Step 3	`Router(config-if)# [no] encapsulation frame-relay mfr number [name]`	Creates a multilink Frame Relay bundle link and associates the link with a bundle. • *number*—The number for the Frame Relay bundle. • *name*—The name for the Frame Relay bundle.
Step 4	`Router(config-if)# [no] frame-relay multilink lid name`	(Optional) Assigns a bundle link identification name with a multilink Frame Relay bundle link. • *name*—The name for the Frame Relay bundle. **Note** The bundle link identification (LID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shutdown** and **no shutdown** commands in interface configuration mode.
Step 5	`Router(config-if)# [no] frame-relay fragment size end-to-end`	(Optional) Enables and configures Frame Relay end-to-end fragmentation (FRF.12) on the serial interface. The range of fragment *size* is 16 to 1600.
Step 6	`Router(config-if)# frame-relay multilink hello seconds`	(Optional) Configures the interval at which a bundle link will send out hello messages. The default value is 10 seconds. • *seconds*—Number of seconds between hello messages sent out over the multilink bundle.
Step 7	`Router(config-if)# frame-relay multilink ack seconds`	(Optional) Configures the number of seconds that a bundle link will wait for a hello message acknowledgment before resending the hello message. The default value is 4 seconds. • *seconds*—Number of seconds a bundle link will wait for a hello message acknowledgment before resending the hello message.
Step 8	`Router(config-if)# frame-relay multilink retry number`	(Optional) Configures the maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment. The default value is 2 tries. • *number*—Maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment.

Verifying Multilink Frame Relay

To verify the Frame Relay multilinks, use the **show frame-relay multilink detailed** command:

```
router# show frame-relay multilink detailed

Bundle: MFR49, State = down, class = A, fragmentation disabled
```

```
 BID = MFR49
 No. of bundle links = 1, Peer's bundle-id =
 Bundle links:

  Serial6/0/0:0, HW state = up, link state = Add_sent, LID = test
    Cause code = none, Ack timer = 4, Hello timer = 10,
    Max retry count = 2, Current count = 0,
    Peer LID = , RTT = 0 ms
    Statistics:
    Add_link sent = 21, Add_link rcv'd = 0,
    Add_link ack sent = 0, Add_link ack rcv'd = 0,
    Add_link rej sent = 0, Add_link rej rcv'd = 0,
    Remove_link sent = 0, Remove_link rcv'd = 0,
    Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,
    Hello sent = 0, Hello rcv'd = 0,
    Hello_ack sent = 0, Hello_ack rcv'd = 0,
    outgoing pak dropped = 0, incoming pak dropped = 0
```

Invert Data on the T1/E1 Interface

If the interface on the 1-Port Channelized OC-3/STM-1 SPA is used to drive a dedicated T1 line that does not have B8ZS encoding, you must invert the data stream on the connecting CSU/DSU or on the interface. Be careful not to invert data on both the CSU/DSU and the interface, as two data inversions will cancel each other out. To invert data on a T1/E1 interface, perform this task:

	Command	Purpose
Step 1	`Router# **configure terminal**`	Enters configuration mode.
Step 2	`Router(config)# **interface serial** *interface-name*`	Selects the serial interface. For addressing information, refer to the "Serial Interface Naming" section on page 18-18.
Step 3	`Router(config-if)# **invert data**`	Inverts the data stream.

This example shows how to verify that invert data has been set:

```
router# show running configuration
.
.
.
interface Serial6/0/0:0
 no ip address
 encapsulation ppp
 logging event link-status
 load-interval 30
 invert data
 no cdp enable
 ppp chap hostname group1
 ppp multilink
 multilink-group 1
!
.
.
.
```

Configuring Multipoint Bridging

Multipoint bridging (MPB) enables the connection of multiple ATM PVCs, Frame Relay PVCs, BCP ports, and WAN Gigabit Ethernet subinterfaces into a single broadcast domain (virtual LAN), together with the LAN ports on that VLAN. This enables service providers to add support for Ethernet-based Layer 2 services to the proven technology of their existing ATM and Frame Relay legacy networks. Customers can then use their current VLAN-based networks over the ATM or Frame Relay cloud. This also allows service providers to gradually update their core networks to the latest Gigabit Ethernet optical technologies, while still supporting their existing customer base.

For MPB configuration guidelines and restrictions and feature compatibility tables, see the "Configuring Multipoint Bridging" section on page 4-17 of Chapter 4, "Configuring the SIPs and SSC."

Configuring Bridging Control Protocol Support

The Bridging Control Protocol (BCP) enables forwarding of Ethernet frames over SONET networks and provides a high-speed extension of enterprise LAN backbone traffic through a metropolitan area. The implementation of BCP on the SPAs includes support for IEEE 802.1D, IEEE 802.1Q Virtual LAN (VLAN), and high-speed switched LANs.

For BCP configuration guidelines and restrictions and feature compatibility tables, see the "Configuring PPP Bridging Control Protocol Support" section on page 4-18 of Chapter 4, "Configuring the SIPs and SSC."

Link Fragmentation and Interleaving (LFI) Guidelines

LFI can function by using either FRF.12 or MLPPP. MLPPP LFI operates in both hardware and software while FRF.12 LFI operates only in hardware.

Hardware MLPPP LFI Guidelines

LFI using MLPPP will function only in hardware if there is just one member link in the MLPPP bundle. The link can be a fractional T1 or full T1. Note the following guidelines:

- The **ppp multilink interleave** command must be configured to enable interleaving.
- Only three fragmentation sizes are supported: 128 bytes, 256 bytes, and 512 bytes.
- Fragmentation is enabled by default, and the default size is 512 bytes.
- A policy map having a priority class must be applied to the main interface.
- When hardware-based LFI is enabled, fragmentation counters are not displayed.

FRF.12 LFI Guidelines

LFI using FRF.12 is always performed in hardware. Note the following guidelines:

- The fragmentation is configured at the main interface.
- Only three fragmentation sizes are supported: 128 bytes, 256 bytes, and 512 bytes.
- A policy map having a priority class must be applied to the main interface.
- FRF.12 can be enabled only on plain FR links.

Configuring QoS Features on Serial SPAs

For information about the QoS features supported by the serial SPAs, see the "Configuring QoS Features on a SIP" section on page 4-33 of Chapter 4, "Configuring the SIPs and SSC."

Configuring CRTP

For information about configuring cRTP, see the "Configuring Compressed Real-Time Protocol" section on page 4-4 or *Configuring Distributed Compressed Real-Time Protocol* at the following URL:

http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfdcrtp.html

Configuring Automatic Protection Switching (APS)

The automatic protection switching (APS) feature allows switchover of interfaces in the event of an interface failure. The protection mechanism has a 1+1 architecture in which a Protect interface is paired with each Working interface. The Working and Protect circuits are synchronized over an independent out of band (OoB) communication channel.

For detailed information about APS, see the "Configuring Automatic Protection Switching" section on page 7-50 of Chapter 7, "Configuring the ATM SPAs." For complete information on APS, including information on additional APS features, refer to the *Cisco IOS Interface Configuration Guide, Release 12.2*.

Automatic Protection Switching Configuration Guidelines

When configuring APS, consider the following guidelines:

- The Working and Protect interfaces must be compatible (for example, both OC-3c interfaces). The interfaces can be on the same SPA, different SPAs in the same switch, or different SPAs in a different switch.
- If using interfaces on different switches, the two switches must have a network connection other than the SONET connection (such as through an Ethernet LAN). Because the APS Protect Group Protocol (PGP) is UDP traffic, this network connection should be reliable with a minimum number of hops.
- The IP addresses on the Working and Protect interfaces should be in the same subnet.
- APS is not supported on SVCs.

Tip Always configure the Working interface before the Protect interface. This will prevent the Protect interface from becoming active and disabling the circuits on the Working interface.

Automatic Protection Switching Configuration Task

To configure the Working and Protect interfaces on the ATM SPAs for basic APS operation, perform the following procedure beginning in global configuration mode.

To configure an interface for APS, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# interface serial interface-name`	Selects the Working interface to configure. For serial interface addressing information, refer to the "Serial Interface Naming" section on page 18-18.
Step 3	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies the IP address and subnet mask for the Working interface. Repeat this command with the **secondary** keyword to specify additional IP addresses to be used for the interface.
Step 4	`Router(config-if)# aps group group-number`	Enables the use of the APS Protect Group Protocol for this Working interface. • *group-number*—Unique number identifying this pair of Working and Protect interfaces. **Note** The **aps group** command is optional if this is the only pair of Working and Protect interfaces on the switch, but is required when you configure more than one pair of Working and Protect interfaces on the same switch.
Step 5	`Router(config-if)# aps working circuit-number`	Identifies the interface as the Working interface. • *circuit-number*—Identification number for this particular channel in the APS pair. Because only 1+1 redundancy is supported, the only valid values are 0 or 1, and the Working interface defaults to 1.
Step 6	`Router(config-if)# aps authentication security-string`	(Optional) Specifies a security string that must be included in every OOB message sent between the Working and Protect interfaces. • *security-string*—Arbitrary string to be used as a password between the Working and Protect interfaces. This string must match the one configured on the Protect interface.
Step 7	`Router(config)# interface serial interface-name`	Selects the Protect interface to configure. For serial interface addressing information, refer to the "Serial Interface Naming" section on page 18-18.

	Command	Purpose
Step 8	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies the IP address and subnet mask for the Protect interface. **Note** This should be the same address that was configured on the Working interface in Step 3. Repeat this command with the **secondary** keyword to specify additional IP addresses to be used for the interface. These should match the secondary IP addresses that are configured on the Working interface.
Step 9	`Router(config-if)# aps group group-number`	Enables the use of the APS Protect Group Protocol for this Protect interface. • *group-number*—Unique number identifying this pair of Working and Protect interfaces. **Note** The **aps group** command is optional if this is the only pair of Working and Protect interfaces on the switch, but is required when you configure more than one pair of Working and Protect interfaces on the same switch.
Step 10	`Router(config-if)# aps protect circuit-number ip-address`	Identifies this interface as the Protect interface: • *circuit-number*—Identification number for this particular channel in the APS pair. Because only 1+1 redundancy is supported, the only valid values are 0 or 1, and the Protect interface defaults to 0. • *ip-address*—The Protect interface uses this IP address to communicate with the Working interface. **Note** This IP address should be the address of the Working interface if the Protect and Working interfaces are on the same switch. If the Working and Protect interfaces are on different switches, this should be the IP address of the Ethernet interface that provides interconnectivity between the two switches.
Step 11	`Router(config-if)# aps authentication security-string`	(Optional) Specifies a security string that must be included in every OOB message sent between the Working and Protect interfaces. • *security-string*—Arbitrary string to be used as a password between the Working and Protect interfaces. This string must match the one configured on the Working interface.

	Command	Purpose
Step 12	`Router(config-if)# **aps revert** *minutes*`	(Optional) Enables the Protect interface to automatically switch back to the Working interface after the Working interface has been up for a specified number of minutes. • *minutes*—Number of minutes until the interface is switched back to the Working interface after the Working interface comes back up. **Note** If this command is not used, you must manually switch back to the Working interface using either the **aps force** *circuit-number* or the **aps manual** *circuit-number* command.
Step 13	`Router(config-if)# [**no**] **aps unidirectional**`	(Optional) Configures the Protect interface for unidirectional mode. The **no** option configures for bidirectional, the default.
Step 14	`Router(config-if)# [**no**] **aps signalling** {**sonet** | **sdh**}`	(Optional) Configures the signaling method. The default is SONET.
Step 15	`Router(config-if)# [**no**] **aps timers** *hello-time hold-time*`	(Optional) Sets the time in seconds between hello packets (*hello-time*) and the waiting time (*hold-time*) before the Protect interface process declares a working interface to be down. The default hello-time is 1 second. The default hold-time is 3 seconds.

Operational Commands for Automatic Protection Switching

To force or prevent APS during operation, perform this task:

	Command	Purpose
Step 1	`Router# **configure terminal**`	Enters configuration mode.
Step 2	`Router(config)# **interface serial** *interface-name*`	Selects the Working interface to configure. For serial interface addressing information, refer to the "Serial Interface Naming" section on page 18-18.
Step 3	`Router(config-if)# [**no**] **aps force** *circuit-number*`	(Optional) Manually switches the specified circuit to a Protect interface, unless a request of equal or higher priority is in effect. • *circuit-number*—Identification number for this particular channel in the APS pair. Because only 1+1 redundancy is supported, the only valid values are 0 or 1, and the Working interface defaults to 1.

	Command	Purpose
Step 4	Router(config-if)# [**no**] **aps manual** *circuit-number*	(Optional) Manually switches the circuit to a Protect interface, unless a request of equal or higher priority is in effect.
Step 5	Router(config-if)# [**no**] **aps lockout** *circuit-number*	(Optional) Prevents a working interface from switching to a protect interface.

Configuring SONET and SDH Overhead Bytes

To modify overhead bytes in SONET or SDH, perform one of these task:

Command	Purpose
Router(config-*xxx*[1])# [**no**] **overhead c2** *0-255*	Changes the C2 byte in SONET or SDH/AU-3 modes. For modes vt-15, c-11, and c-12, the default is 2. For mode ct3, the default is 4.
Router(config-controller)# [**no**] **overhead s1s0** *0-3*	Configures automatically the S1S0 bits of H1 according to the framing unless specified here.
Router(config-controller)# [**no**] **overhead s1s0 ignore**	Causes the s1s0 overhead bits to be ignored. This may be necessary for Australian conformance, due to the limitation of Spectra-4x155 and possibly Spectra-622.
Router(config-*xxx*)# [**no**] **overhead j0** *0-255*	Specifies the J0 trace byte to be transmitted in the regenerator section overhead.
Router(config-*xxx*)# [**no**] **overhead j0 expect** *0-255*	Specifies the expected value of the received J0 trace byte.
Router(config-*xxx*)# [**no**] **overhead rs-tim ignore**	Supresses the J0 trace byte mismatch alarm.
Router(config-*xxx*)# [**no**] **overhead j1 length** [**16** | **64**]	Sets the J1 trace bytes in the high order (STS) path overhead to either 16 or 64 bytes.
Router(config-*xxx*)# [**no**] **overhead j1 message** *ascii_line*	Specifies the message to transmit in the the J1 trace bytes.
Router(config-*xxx*)# [**no**] **overhead j1 expect message** *ascii_line*	Specifies the expected value of the received message in the J1 trace bytes.
Router(config-*xxx*)# [**no**] **overhead hp-tim ignore**	Specifies to ignore the J1 trace identifier mismatch alarm.
Router(config-*xxx*)# [**no**] **overhead j2 message** *ascii_line*	Specifies the message to transmit in the the J2 trace byte byte in the low order (VT) path overhead.
Router(config-*xxx*)# [**no**] **overhead j2 expect message** *ascii_line*	Specifies the expected value of the received message in the J2 trace byte.

Command	Purpose
`Router(config-xxx)# [no] overhead lp-tim ignore`	Specifies to ignore the J2 trace identifier mismatch alarm.
`Router(config-controller)# [no] overhead s1byte ignore`	Specifies to ignore the received synchronization byte S1 value of 0xF (do not switch to internal clock).

[1] The actual command prompt is `Router(config-ctrlr-sts1)#` for SONET, `Router(config-ctrlr-au3)#` for SDH/AU-3, or `Router(config-controller)# for SDH/AU-4`.

Configuring a Bit Error Rate Test (BERT)

To start a BERT pattern on a port, perform this task in interface configuration mode:

Command	Purpose
Router(config-*xxx*[1])# [**no**] [*prefix*[2]] [**t1** [*t1#*] | [**e1** [*e1#*] | **t3**] **bert pattern** {**0s** | **1s** | **2^15** | **2^20** | **2^23** | **alt-0-1** | **qrss**} **interval** *minutes*	Starts or stops a specific bit pattern on a DS1/E1/T3 line. For DS1 and E1, you can optionally specify a specific channel number as the **t1** or **e1** number. The interval specifies the length of the test in minutes. The range is 1 to 14400. The **no** option stops the test. **Note** Only six E1 BERTs can be performed concurrently due to TEMUX-84/TEMUX-84E limitations.

[1] The actual command prompt depends on the interface selected.

[2] The actual command prefix is **vtg** *vtg-number* for SONET and **tug-2** *tug-2-number* for SDH.

Following are the available BERT pattern options:

0s	Repeating pattern of zeros (...000...).
1s	Repeating pattern of ones (...111...).
2^15	Pseudorandom 0.151 test pattern that is 32,768 bits in length.
2^20	Pseudo-andom 0.153 test pattern that is 1,048,575 bits in length.
2^23	Pseudorandom 0.151 test pattern that is 8,388,607 bits in length.
alt-0-1	Repeating pattern of alternating zeros and ones (...01010...).
qrss	Pseudorandom quasi-random signal sequence (QRSS) 0.151 test pattern that is 1,048,575 bits in length.

Saving the Configuration

To save your running configuration to nonvolatile random-access memory (NVRAM), perform this task in privileged EXEC configuration mode:

Command	Purpose
`Router# copy running-config startup-config`	Writes the new configuration to NVRAM.

For more information about managing configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

Verifying the Interface Configuration

Besides using the **show running-configuration** command to display your Catalyst 6500 Series switch configuration settings, you can use the **show interface serial** and the **show controllers serial** commands to get detailed information on a per-port basis for your 1-Port Channelized OC-3/STM-1 SPA.

To view the overall or per-port information, perform this task:

Command	Purpose
Router(config)# **show interface serial** [*interface-name*]	Displays the configuration settings of all serial interfaces or of a specific serial interface. For serial interface addressing information, refer to the "Serial Interface Naming" section on page 18-18.
Router(config)# **show controllers sonet** [*interface-name*][**brief** | **tabular** | **remote performance** [**brief** | **tabular**]]	Displays the configuration settings of all DS1 or E1 interfaces or of a specific interface. The *interface-name* is the DS1 or E1 serial interface address as shown in the "Serial Interface Naming" section on page 18-18, with the channel-group omitted. The remote performance option is only available for DS1 interfaces.
Router(config)# **show controllers t3** *interface-name* [**brief** | **tabular** | **remote performance** [**brief** | **tabular**]]	Displays the configuration settings of all DS1 interfaces on a DS3 or of a specific DS1 interface. The *interface-name* can be *slot/subslot/port or slot/subslot/port/ds1*.

Verifying Interface Configuration and Status

To find detailed interface information for the 1-Port Channelized OC-3/STM-1 SPA, use the **show interface serial** command.

The following example provides sample output for a SPA located in the first subslot of the Cisco 7600 SIP-200 installed in slot 2 of a Catalyst 6500 Series switch:

```
Router(config)# show interface serial
Serial2/0/0.1/2 unassigned YES TFTP administratively down down
Serial2/1/0.1/1/1:0 unassigned YES unset down down
Serial2/1/0.1/2/4:0 unassigned YES unset down down
Serial2/1/0.1/2/4:1 unassigned YES unset down down
Serial2/1/0.2/1:0 unassigned YES unset down down
Serial2/1/0.2/2:0 unassigned YES unset down down
Serial2/1/0.2/3:0 unassigned YES unset down down
Serial2/1/0.3 unassigned YES unset down down
UUT#sh int Serial2/1/0.1/1/1:0
```

```
Serial2/1/0.1/1/1:0 is down, line protocol is down
Hardware is Channelized-T3
MTU 1500 bytes, BW 192 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, crc 16, loopback not set
Keepalive set (10 sec)
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
Available Bandwidth 192 kilobits/sec
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions alarm present
VC 2: timeslot(s): 1-3, Transmitter delay 0, non-inverted data
UUT#sh run | beg 2/1/0
controller SONET 2/1/0
ais-shut
framing sonet
clock source line
overhead j0 1
!
sts-1 1
mode vt-15
vtg 1 t1 1 channel-group 0 timeslots 1-3
vtg 2 t1 4 channel-group 0 timeslots 1-2,5-6
vtg 2 t1 4 channel-group 1 timeslots 3,7,9
!
sts-1 2
mode ct3
t1 1 channel-group 0 timeslots 1-24
t1 2 channel-group 0 timeslots 1-12
t1 3 channel-group 0 timeslots 1
!
sts-1 3
mode t3
!
controller T3 3/1/0
shutdown
cablelength 224
!
controller T3 3/1/1
shutdown
cablelength 224
!
!
interface Loopback0
ip address 172.10.11.1 255.255.255.255
.
.
```

Verifying Per-Port Interface Configuration and Status

To find detailed interface information on a per-port basis, use the **show interface serial** command and specify the port as described in the "Serial Interface Naming" section on page 18-18.

The following example provides sample output for interface port 0 on the SPA located in the first subslot of the Cisco 7600 SIP-200 installed in slot 2 of a Catalyst 6500 Series switch:

```
Router# show interface serial 2/1/0.2/1:0
Serial2/1/0.2/1:0 is down, line protocol is down
Hardware is Channelized-T3
MTU 1500 bytes, BW 1536 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, crc 16, loopback not set
Keepalive set (10 sec)
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
Available Bandwidth 1536 kilobits/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions alarm present
VC 5: timeslot(s): 1-24, Transmitter delay 0, non-inverted data
UUT#sh int Serial2/1/0.3
Serial2/1/0.3 is down, line protocol is down
Hardware is CHOCx SPA
MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec, rely 255/255, load 1/255
Encapsulation HDLC, crc 16, loopback not set
Keepalive set (10 sec)
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
Available Bandwidth 44210 kilobits/sec
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 parity
```

CHAPTER 19

Troubleshooting the Serial SPAs

This chapter describes techniques that you can use to troubleshoot the operation of your serial SPAs.

It includes the following sections:

The first section provides information about basic interface troubleshooting. If you are having a problem with your SPA, use the steps in the "General Troubleshooting Information" section on page 19-1 to begin your investigation of a possible interface configuration problem.

To perform more advanced troubleshooting, see the other sections in this chapter.

For more information about troubleshooting serial lines, see the *Internetwork Troubleshooting Handbook* at: http://www.cisco.com/univercd/cc/td/doc/cisintwk/itg_v1/index.htm.

General Troubleshooting Information

This section describes general information for troubleshooting SIPs and SPAs. It includes the following sections:

Interpreting Console Error Messages

To view the explanations and recommended actions for Catalyst 6500 Series switch error messages, including messages related to Catalyst 6500 Series switch SIPs and SPAs, see the *Catalyst 6500 Series Cisco IOS System Message Guide, 12.2SX*.

System error messages are organized in the documentation according to the particular system facility that produces the messages. The SIP and SPA error messages use the following facility names:

- Cisco 7600 SIP-200—C7600_SIP200
- 2-Port and 4-Port Channelized T3 SPA—SPA_CHOC_DSX

Using debug Commands

Along with the other **debug** commands supported on the Catalyst 6500 Series switch, you can obtain specific debug information for SPAs on the Catalyst 6500 Series switch using the **debug hw-module subslot** privileged exec command.

The **debug hw-module subslot** command is intended for use by Cisco Systems technical support personnel. For more information about the **debug hw-module subslot** command and other **debug** commands, see the *Cisco IOS Debug Command Reference, Release 12.2*.

Caution Because debugging output is assigned high priority in the CPU process, it can cause the system to be unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. You should use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

For information about other **debug** commands supported on the Catalyst 6500 Series switch, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* and any related feature documents for Cisco IOS Release 12.2 SX.

Using show Commands

There are several **show** commands that you can use to monitor and troubleshoot the SIPs and SPAs on the Catalyst 6500 Series switch. This chapter describes using the **show interfaces** and **show controllers** commands to perform troubleshooting of your SPA.

For more information about **show** commands to verify and monitor SIPs and SPAs, see the following chapters of this guide:

- Chapter 16, “Configuring the 2-Port and 4-Port Clear Channel T3/E3 SPAs”
- Chapter 15, “Configuring the 8-Port Channelized T1/E1 SPA”
- Chapter 17, “Configuring the 2-Port and 4-Port Channelized T3 SPAs”

Performing Basic Interface Troubleshooting

You can perform most of the basic interface troubleshooting using the **show interfaces serial** command and examining several areas of the output to determine how the interface is operating.

The output of the **show interfaces serial** exec command displays information specific to serial interfaces.

Note The output of the **show interfaces serial** command will vary depending on the type of serial SPA. Other fields that may be shown in the display are described in detail in the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX*.

This section describes how to use the **show interfaces serial** command to diagnose serial line connectivity problems in a wide-area network (WAN) environment. The following sections describe some of the important fields of the command output:

- Serial Lines: show interfaces serial Status Line Conditions, page 19-3
- Serial Lines: Increasing Output Drops on Serial Link, page 19-7
- Serial Lines: Increasing Input Drops on Serial Link, page 19-8
- Serial Lines: Increasing Input Errors in Excess of 1 Percent of Total Interface Traffic, page 19-9
- Serial Lines: Troubleshooting Serial Line Input Errors, page 19-10
- Serial Lines: Increasing Interface Resets on Serial Link, page 19-13
- Serial Lines: Increasing Carrier Transitions Count on Serial Link, page 19-14

Serial Lines: show interfaces serial Status Line Conditions

You can identify five possible problem states in the interface status line of the **show interfaces serial** display:

- Serial *x* is down, line protocol is down
- Serial *x* is up, line protocol is down
- Serial *x* is up, line protocol is up (looped)
- Serial *x* is up, line protocol is down (disabled)
- Serial *x* is administratively down, line protocol is down

The following example shows the interface statistics on the first port of a T3/E3 SPA installed in subslot 0 of the SIP located in chassis slot 5.

```
Router# show interfaces serial

Serial5/0/0 is up, line protocol is up
  Hardware is SPA-4T3E3
  Internet address is 110.1.1.2/24
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 234/255, rxload 234/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:05, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 40685000 bits/sec, 115624 packets/sec
  5 minute output rate 40685000 bits/sec, 115627 packets/sec
     4653081241 packets input, 204735493724 bytes, 0 no buffer
     Received 4044 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
```

```
         0 parity
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
4652915555 packets output, 204728203520 bytes, 0 underruns
0 output errors, 0 applique, 4 interface resets
0 output buffer failures, 0 output buffers swapped out
2 carrier transitions
```

Table 19-1 shows the interface status conditions, possible problems associated with the conditions, and solutions to those problems

Table 19-1 Serial Lines: show interfaces serial Status Line Conditions

Status Line Condition	Possible Problem	Solution
Serial *x* is up, line protocol is up	—	This is the proper status line condition. No action is required.
Serial *x* is down, line protocol is down	The switch is not sensing a carrier detect (CD) signal (that is, the CD is not active). The line is down or is not connected on the far end. Cabling is faulty or incorrect. Hardware failure has occurred (CSU/DSU).	**1.** Check the CD LEDs to see whether the CD is active, or insert a breakout box on the line to check for the CD signal. **2.** Verify that you are using the proper cable (see your hardware installation documentation). **3.** Insert a breakout box and check all control leads. **4.** Contact your leased-line or other carrier service to see whether there is a problem. **5.** Swap faulty parts. **6.** If you suspect faulty switch hardware, change the serial line to another port. If the connection comes up, the previously connected interface has a problem.

Table 19-1 Serial Lines: show interfaces serial Status Line Conditions (continued)

Status Line Condition	Possible Problem	Solution
Serial *x* is up, line protocol is down	A local or remote switch is misconfigured. Keepalives are not being sent by the remote router. A leased-line or other carrier service problem has occurred (noisy line or misconfigured or failed switch). A timing problem has occurred on the cable. A local or remote CSU/DSU has failed. Router hardware (local or remote) has failed.	1. Put the line in local loopback mode and use the **show interfaces serial** command to determine whether the line protocol comes up. **Note** If the line protocol comes up, a failed remote device is the likely problem. This solution will only work with HDLC encapsulation. For FR and PPP encapsulation a looped interface will always have the line protocol down. In addition, you may need to change the encapsulation to HDLC to debug this issues. 2. If the problem appears to be on the remote end, repeat Step 1 on the remote interface. 3. Verify all cabling. Make certain that the cable is attached to the correct interface, the correct CSU/DSU, and the correct remote termination point. 4. Enable the **debug serial interface** exec command. **Note** First enable per- interface debugging using the **debug interface serial** *x* command. Depending on the encapsulation, also enable the corresponding debug. HDLC: **debug serial interface** PPP: **debug ppp negotiation** FR: **debug frame-relay lmi** ⚠ **Caution** Because debugging output is assigned high priority in the CPU process, it can cause the system to become unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. You should use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

Table 19-1 Serial Lines: show interfaces serial Status Line Conditions (continued)

Status Line Condition	Possible Problem	Solution
		5. If the line protocol does not come up in local loopback mode, and if the output of the **debug serial interface** exec command shows that the keepalive counter is not incrementing, a switch hardware problem is likely. Swap switch interface hardware. 6. If the line protocol comes up and the keepalive counter increments, the problem is *not* in the local switch. Troubleshoot the serial line, as described in the sections "Troubleshooting Clocking Problems" and "CSU and DSU Loopback Tests," later in this chapter. 7. If you suspect faulty switch hardware, change the serial line to an unused port. If the connection comes up, the previously connected interface has a problem.
Serial *x* is up, line protocol is up (looped)	A loop exists in the circuit. The sequence number in the keepalive packet changes to a random number when a loop is initially detected. If the same random number is returned over the link, a loop exists.	1. Use the **show running-config** privileged exec command to look for any **loopback** interface configuration command entries. 2. If you find a **loopback** interface configuration command entry, use the **no loopback** interface configuration command to remove the loop. 3. If you do not find the **loopback** interface configuration command, examine the CSU/DSU to determine whether they are configured in manual loopback mode. If they are, disable manual loopback. 4. Reset the CSU or DSU, and inspect the line status. If the line protocol comes up, no other action is needed. 5. If the CSU or DSU is not configured in manual loopback mode, contact the leased-line or other carrier service for line troubleshooting assistance.

Table 19-1 Serial Lines: show interfaces serial Status Line Conditions (continued)

Status Line Condition	Possible Problem	Solution
Serial *x* is up, line protocol is down (disabled)	A high error rate has occurred due to a remote device problem. A CSU or DSU hardware problem has occurred. Switch hardware (interface) is bad.	1. Troubleshoot the line with a serial analyzer and breakout box. Examine the output of **show controller T1** or **show controller T3** or **show controller serial x** depending on whether the SPA is a T1/E1 or CT3 or T3/E3. 2. Loop CSU/DSU (DTE loop). If the problem continues, it is likely that there is a hardware problem. If the problem does not continue, it is likely that there is a telephone company problem. 3. Swap out bad hardware, as required (CSU, DSU, switch, local or remote switch).
Serial *x* is administratively down, line protocol is down	The switch configuration includes the **shutdown** interface configuration command. A duplicate IP address exists.	1. Check the switch configuration for the **shutdown** command. 2. Use the **no shutdown** interface configuration command to remove the **shutdown** command. 3. Verify that there are no identical IP addresses using the **show running-config** privileged exec command or the **show interfaces** exec command. 4. If there are duplicate addresses, resolve the conflict by changing one of the IP addresses.

Serial Lines: Increasing Output Drops on Serial Link

Output drops appear in the output of the **show interfaces serial** command when the system is attempting to hand off a packet to a transmit buffer but no buffers are available.

Symptom: Increasing output drops on serial link.

Table 19-2 shows the possible problem that might cause this symptom and describes solutions to that problem.

Table 19-2 Serial Lines: Increasing Output Drops on Serial Link

Possible Problem	Solution
Input rate to serial interface exceeds bandwidth available on serial link	1. Minimize periodic broadcast traffic, such as routing and SAP[1] updates, by using access lists or by other means. For example, to increase the delay between SAP updates, use the **ipx sap-interval** interface configuration command. 2. Increase the output hold queue size in small increments (for instance, 25 percent), using the **hold-queue out** interface configuration command. 3. Implement priority queuing on slower serial links by configuring priority lists. For information on configuring priority lists, see the Cisco IOS configuration guides and command references. **Note:** Output drops are acceptable under certain conditions. For instance, if a link is known to be overused (with no way to remedy the situation), it is often considered preferable to drop packets than to hold them. This is true for protocols that support flow control and can retransmit data (such as TCP/IP and Novell IPX[2]). However, some protocols, such as DECnet and local-area transport, are sensitive to dropped packets and accommodate retransmission poorly, if at all.

[1] SAP = Service Advertising Protocol

[2] IPX = Internetwork Packet Exchange

Serial Lines: Increasing Input Drops on Serial Link

Input drops appear in the output of the **show interfaces serial** exec command when too many packets from that interface are still being processed in the system.

Symptom: Increasing number of input drops on serial link.

Table 19-3 shows the possible problems that might cause this symptom and describes solutions to that problem.

Table 19-3 Serial Lines: Increasing Input Drops on Serial Link

Possible Problem	Solution
Input rate exceeds the capacity of the switch, or input queues exceed the size of output queues	**Note**: Input drop problems are typically seen when traffic is being routed between faster interfaces (such as Ethernet, Token Ring, and FDDI[1]) and serial interfaces. When traffic is light, there is no problem. As traffic rates increase, backups start occurring. Switches drop packets during these congested periods. 1. Increase the output queue size on common destination interfaces for the interface that is dropping packets. Use the **hold-queue** *number* **out** interface configuration command. Increase these queues by small increments (for instance, 25 percent) until you no longer see drops in the **show interfaces** output. The default output hold queue limit is 40 packets. 2. Reduce the input queue size, using the **hold-queue** *number* **in** interface configuration command, to force input drops to become output drops. Output drops have less impact on the performance of the switch than do input drops. The default input hold queue is 75 packets.

[1] FDDI = Fiber Distributed Data Interface

Serial Lines: Increasing Input Errors in Excess of 1 Percent of Total Interface Traffic

If input errors appear in the **show interfaces serial** output, there are several possible sources of those errors. The most likely sources are summarized in Table 19-4.

Note Any input error value for cyclic redundancy check (CRC) errors, framing errors, or aborts above 1 percent of the total interface traffic suggests some kind of link problem that should be isolated and repaired.

Symptom: Increasing number of input errors in excess of 1 percent of total interface traffic.

Table 19-4 Serial Lines: Increasing Input Errors in Excess of 1 Percent of Total Interface Traffic

Possible Problem	Solution
The following problems can result in this symptom: • Faulty telephone company equipment • Noisy serial line • Incorrect clocking configuration • Incorrect cable or cable that is too long • Bad cable or connection • Bad CSU or DSU • Bad switch hardware • Data converter or other device being used between switch and DSU	**Note:** Cisco strongly recommends against the use of data converters when you are connecting a switch to a WAN or a serial network. 1. Use a serial analyzer to isolate the source of the input errors. If you detect errors, there likely is a hardware problem or a clock mismatch in a device that is external to the switch. 2. Use the **loopback** and **ping** tests to isolate the specific problem source. For more information, see the sections "Using Extended ping Tests" and "CSU and DSU Loopback Tests," later in this chapter. 3. Look for patterns. For example, if errors occur at a consistent interval, they could be related to a periodic function, such as the sending of routing updates.

Serial Lines: Troubleshooting Serial Line Input Errors

Table 19-5 describes the various types of input errors displayed by the **show interfaces serial** command, possible problems that might be causing the errors, and solutions to those problems.

Table 19-5 Serial Lines: Troubleshooting Serial Line Input Errors

Input Error Type (Field Name)	Possible Problem	Solution
CRC errors (CRC)	CRC errors occur when the CRC calculation does not pass (indicating that data is corrupted) for one of the following reasons: • The serial line is noisy. • The serial cable is too long, or the cable from the CSU/DSU to the switch is not shielded. • SCTE mode is not enabled on DSU. • The CSU line clock is incorrectly configured. • A ones density problem has occurred on the T1 link (incorrect framing or coding specification).	1. Ensure that the line is clean enough for transmission requirements. Shield the cable, if necessary. 2. Make sure that the cable is within the recommended length (no more than 50 feet [15.24 meters], or 25 feet [7.62 meters] for a T1 link). 3. Ensure that all devices are properly configured for a common line clock. Set SCTE on the local and remote DSU. If your CSU/DSU does not support SCTE, see the section "Inverting the Transmit Clock," later in this chapter. 4. Make certain that the local and remote CSU/DSU are configured for the same framing and coding scheme as that used by the leased-line or other carrier service (for example, ESF/B8ZS). 5. Contact your leased-line or other carrier service, and have it perform integrity tests on the line.

Table 19-5 Serial Lines: Troubleshooting Serial Line Input Errors (continued)

Input Error Type (Field Name)	Possible Problem	Solution
Framing errors (frame)	A framing error occurs when a packet does not end on an 8-bit byte boundary for one of the following reasons: • The serial line is noisy. • The cable is improperly designed; the serial cable is too long; the cable from the CSU or DSU to the switch is not shielded. • SCTE mode is not enabled on the DSU; the CSU line clock is incorrectly configured; one of the clocks is configured for local clocking. • A ones density problem has occurred on the T1 link (incorrect framing or coding specification).	1. Ensure that the line is clean enough for transmission requirements. Shield the cable, if necessary. Make certain that you are using the correct cable. 2. Make sure that the cable is within the recommended length (no more than 50 feet [15.24 meters], or 25 feet [7.62 meters] for a T1 link). 3. Ensure that all devices are properly configured to use a common line clock. Set SCTE on the local and remote DSU. If your CSU/DSU does not support SCTE, see the section "Inverting the Transmit Clock," later in this chapter. 4. Make certain that the local and remote CSU/DSU is configured for the same framing and coding scheme as that used by the leased-line or other carrier service (for example, ESF[1]/B8ZS[2]). 5. Contact your leased-line or other carrier service, and have it perform integrity tests on the line.

Table 19-5 Serial Lines: Troubleshooting Serial Line Input Errors (continued)

Input Error Type (Field Name)	Possible Problem	Solution
Aborted transmission (abort)	Aborts indicate an illegal sequence of 1 bit (more than seven in a row) The following are possible reasons for this to occur: • SCTE mode is not enabled on DSU. • The CSU line clock is incorrectly configured. • The serial cable is too long, or the cable from the CSU or DSU to the switch is not shielded. • A ones density problem has occurred on the T1 link (incorrect framing or coding specification). • A packet terminated in middle of transmission (typical cause is an interface reset or a framing error or a buffer overrun). • A hardware problem has occurred (bad circuit, bad CSU/DSU, or bad sending interface on remote switch).	1. Ensure that all devices are properly configured to use a common line clock. Set SCTE on the local and remote DSU. If your CSU/DSU does not support SCTE, see the section “Inverting the Transmit Clock,” later in this chapter. 2. Shield the cable, if necessary. Make certain that the cable is within the recommended length (no more than 50 feet [15.24 meters], or 25 feet [7.62 meters] for a T1 link). Ensure that all connections are good. 3. Check the hardware at both ends of the link. Swap faulty equipment, as necessary. 4. Lower data rates and determine whether aborts decrease. 5. Use local and remote loopback tests to determine where aborts are occurring (see the section “Special Serial Line Tests,” later in this chapter). 6. Contact your leased-line or other carrier service, and have it perform integrity tests on the line.

[1] ESF = Extended Superframe Format

[2] B8ZS = binary eight-zero substitution

Serial Lines: Increasing Interface Resets on Serial Link

Interface resets that appear in the output of the **show interfaces serial** exec command are the result of missed keepalive packets.

Symptom: Increasing interface resets on serial link.

Table 19-6 shows the possible problems that might cause this symptom and describes solutions to those problems.

Table 19-6 Serial Lines: Increasing Interface Resets on Serial Link

Possible Problem	Solution
The following problems can result in this symptom: • Congestion on link (typically associated with output drops) • Bad line causing CD transitions • Possible hardware problem at the CSU, DSU, or switch	When interface resets are occurring, examine other fields of the **show interfaces serial** command output to determine the source of the problem. Assuming that an increase in interface resets is being recorded, examine the following fields: 1. If there is a high number of output drops in the **show interfaces serial** output, see the section "Serial Lines: Increasing Output Drops on Serial Link," earlier in this chapter. 2. Check the Carrier Transitions field in the **show interfaces serial** display. If carrier transitions are high while interface resets are being registered, the problem is likely to be a bad link or a bad CSU or DSU. Contact your leased-line or carrier service, and swap faulty equipment, as necessary. 3. Examine the Input Errors field in the **show interfaces serial** display. If input errors are high while interface resets are increasing, the problem is probably a bad link or a bad CSU/DSU. Contact your leased-line or other carrier service, and swap faulty equipment, as necessary.

Serial Lines: Increasing Carrier Transitions Count on Serial Link

Carrier transitions appear in the output of the **show interfaces serial** exec command whenever there is an interruption in the carrier signal (such as an interface reset at the remote end of a link).

Symptom: Increasing carrier transitions count on serial link.

Table 19-7 shows the possible problems that might cause this symptom and describes solutions to those problems.

Table 19-7 Serial Lines: Increasing Carrier Transitions Count on Serial Link

Possible Problem	Solution
The following problems can result in this symptom: • Line interruptions due to an external source (such as physical separation of cabling, red or yellow T1 alarms, or lightning striking somewhere along the network) • Faulty switch, DSU, or switch hardware	1. Check hardware at both ends of the link (attach a breakout box or a serial analyzer, and test to determine the source of problems). 2. If an analyzer or breakout box is incapable of identifying any external problems, check the switch hardware. 3. Swap faulty equipment, as necessary.

Using Bit Error Rate Tests

Bit Error Rate (BER) test circuitry is built into the 4-Port Clear Channel T3/E3 SPA. With BER tests, you can test cables and signal problems in the field. You can configure individual T1 lines to run BER tests, but only one BER test circuit exists for all 28 T1 lines. Only one BER test can be run on a single T3 port at any given time.

There are two categories of test patterns that can be generated by the onboard BER test circuitry: pseudorandom and repetitive. Pseudorandom test patterns are exponential numbers and conform to the CCITT/ITU O.151 and O.153 specifications; repetitive test patterns are all zeros, all ones, or alternating zeros and ones.

A description of each type of test pattern follows:

- Pseudorandom test patterns:
  - 2^15 (per CCITT/ITU O.151)
  - 2^20 (per CCITT/ITU O.153)
  - 2^23 (per CCITT/ITU O.151)
- Repetitive test patterns:
  - All zeros (0s)
  - All ones (1s)
  - Alternating zeros (0s) and ones (1s)

Both the total number of error bits received and the total number of bits received are available for analysis. You can set the testing period from 1 minute to 14,400 minutes (240 hours), and you can also retrieve the error statistics anytime during the BER test.

When running a BER test, your system expects to receive the same pattern that it is transmitting. To help ensure this:

- Use a loopback at a location of your choice in the link or network. To see how to configure a loopback, go to the "Using loopback Commands" section on page 19-17.
- Configure remote testing equipment to transmit the same BER test pattern at the same time.

Configuring a BER Test

To send a BER test pattern on an interface, use the **bert pattern** command as described in the *Cisco IOS Interface and Hardware Component Command Reference* at this URL:

http://www.cisco.com/en/US/docs/ios/interface/command/reference/ir_a1.html#wp1013940

Viewing a BER Test

You can view the results of a BER test with the **show controllers** command.

You can view the results of a BER test at the following times:

- After you terminate the test using the no t1 bert command.
- After the test runs completely.
- Anytime during the test (in real time).

```
Router# show controllers serial T3 1/0/0
T3 1/0/0 is up.
C2T3 H/W Version : 3, C2T3 ROM Version : 0.79, C2T3 F/W Version : 0.29.0
T3 1/0/0 T1 1
No alarms detected.
Clock Source is internal.
BERT test result (running)
   Test Pattern : 2^15, Status : Sync, Sync Detected : 1
   Interval : 5 minute(s), Time Remain : 5 minute(s)
   Bit Errors(Since BERT Started): 6 bits,
   Bits Received(Since BERT start): 8113 Kbits
   Bit Errors(Since last sync): 6 bits
   Bits Received(Since last sync): 8113 Kbits
```

Interpreting BER Test Results

Table 19-8 explains the output of the preceding command.

Table 19-8 Interpreting BER Test Results

Field	Description
BERT test result (running)	Indicates the current state of the test. In this case, “running” indicates that the BER test is still in progress. After a test is completed, “done” is displayed.
Test Pattern : 2^15, Status : Sync, Sync Detected : 1	Indicates the test pattern you selected for the test (2^15), the current synchronization state (sync), and the number of times synchronization has been detected during this test (1).
Interval : 5 minute(s), Time Remain : 5 minute(s)	Indicates the time the test takes to run and the time remaining for the test to run. If you terminate a BER test, you receive a message similar to the following: `Interval : 5 minute(s), Time Remain : 2 minute(s) (unable to complete)` "Interval: 5 minutes" indicates the configured run time for the test. "Time Remain : 2 minutes" indicates the time remaining in the test prior to termination. "(Unable to complete)" signifies that you interrupted the test.
Bit Errors(Since BERT Started): 6 bits, Bits Received(Since BERT start): 8113 Kbits Bit Errors(Since last sync): 6 bits Bits Received(Since last sync): 8113 Kbits	These four lines show the bit errors that have been detected versus the total number of test bits that have been received since the test started and since the last synchronization was detected.

Using loopback Commands

Loopback support is useful for testing the interface without connectivity to the network, or for diagnosing equipment malfunctions between the interface and a device. The 2-Port and 4-Port Clear Channel T3/E3 SPA supports both an internal and an external loopback mode. The external loopback mode requires the use of a loopback cable and implements a loopback through the transceiver on the SPA.

You can also configure an internal loopback without the use of a loopback cable that implements a loopback at the PHY device internally. By default, loopback is disabled.

To configure local loopback, perform this task:

Command	Purpose
`Router#` **`configure terminal`**	Enters global configuration mode.

T3/E3
`Router(config)#` **`interface serial`** *`slot/subslot/port`*
T1/E1
`Router(config)#` **`controller`** *`slot/subslot/port`*
	Selects the interface to configure.
	- *slot/subslot/port*—Specifies the location of the interface.
	- *slot/subslot/port*—Specifies the location of the T1/E1 controller.

T3/E3
`Router(config-if)#` **`loopback`** `{`**`local`** `|` **`dte`** `|` **`network`** `{`**`line`** `|` **`payload`**`}` `|` **`remote`**`}`
T1/E1
`Router(config-controller)#` **`loopback`** `{`**`local`** `[`**`line`** `|` **`payload`**`]` `|` **`diag`**`}`
	Specifies the loopback mode.
	- **local**—Loopback after going through the framer toward the terminal.
	- **dte**—Loopback after the LIU towards the terminal.
	- **network**—Loopback towards the network.
	- **remote**—Send FEAC to set remote in loopback.
	- **line**—Loopback toward network before going through framer.
	- **payload**—Loopback toward network after going through framer.
	- **diag**—Loopback after going through the HDLC controller towards the terminal.

Verifying Loopback Mode

This example shows how to verify loopback mode:

```
Router# show interfaces serial 6/0/0
Serial6/0/0 is up, line protocol is up (looped)
  Hardware is SPA-4T3E3
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 225/255, rxload 221/255
  Encapsulation FRAME-RELAY, crc 16, loopback set (local)
  Keepalive set (10 sec)
  LMI enq sent  13281, LMI stat recvd 13280, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 1, LMI stat sent  0, LMI upd sent  0
  LMI DLCI 1023  LMI type is CISCO  frame relay DTE
  FR SVC disabled, LAPF state down
```

```
Broadcast queue 0/256, broadcasts sent/dropped 0/0, interface broadcasts 0
Last input 00:00:07, output 00:00:00, output hang never
Last clearing of "show interface" counters 1d12h
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 612756
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 38446000 bits/sec, 109217 packets/sec
5 minute output rate 39023000 bits/sec, 110854 packets/sec
   14601577951 packets input, 642478074437 bytes, 0 no buffer
   Received 0 broadcasts (0 IP multicast)
   0 runts, 0 giants, 0 throttles
            0 parity
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   14545044296 packets output, 639982568049 bytes, 0 underruns
   0 output errors, 0 applique, 1 interface resets
   0 output buffer failures, 0 output buffers swapped out
   0 carrier transitions
 rxLOS inactive, rxLOF inactive, rxAIS inactive
 txAIS inactive, rxRAI inactive, txRAI inactive
```

Using the Cisco IOS Event Tracer to Troubleshoot Problems

Note This feature is intended for use as a software diagnostic tool and should be configured only under the direction of a Cisco Technical Assistance Center (TAC) representative.

The Event Tracer feature provides a binary trace facility for troubleshooting Cisco IOS software. This feature gives Cisco service representatives additional insight into the operation of the Cisco IOS software and can be useful in helping to diagnose problems in the unlikely event of an operating system malfunction or, in the case of redundant systems, Route Processor switchover.

Event tracing works by reading informational messages from specific Cisco IOS software subsystem components that have been preprogrammed to work with event tracing, and by logging messages from those components into system memory. Trace messages stored in memory can be displayed on the screen or saved to a file for later analysis.

The SPAs currently support the “spa” component to trace SPA OIR-related events.

For more information about using the Event Tracer feature, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/evnttrcr.html

Preparing for Online Insertion and Removal of a SPA

The Catalyst 6500 Series switch supports online insertion and removal (OIR) of the SIP, in addition to each of the SPAs. Therefore, you can remove a SIP with its SPAs still intact, or you can remove a SPA independently from the SIP, leaving the SIP installed in the switch.

This means that a SIP can remain installed in the switch with one SPA remaining active, while you remove another SPA from one of the SIP subslots. If you are not planning to immediately replace a SPA into the SIP, then be sure to install a blank filler plate in the subslot. The SIP should always be fully installed with either functional SPAs or blank filler plates.

For more information about activating and deactivating SPAs in preparation for OIR, see the “Preparing for Online Insertion and Removal of SIPs and SPAs” topic in the “Troubleshooting a SIP” chapter in this guide.

PART 7

IPsec VPN Shared Port Adapter

CHAPTER 20

Overview of the IPsec VPN SPA

This chapter provides an overview of the release history, and feature and Management Information Base (MIB) support for the IPsec VPN SPA.

This chapter includes the following sections:

Release History

Release	Modification
Cisco IOS Release 12.2(33)SXI	The following modifications were made: • Support was introduced for the following features: – Platform QoS on tunnel interface – Platform ACLs on tunnel interface – Multicast over VTI • Fragmentation behavior was changed. • The **ip tcp adjust-mss** command is supported in crypto-connect and VRF modes on GRE, GRE/TP, and sVTI tunnels.

Cisco IOS Release 12.2(33)SXH	The following modifications were made: • Support was introduced for the following features: – Configure VTI or GRE/IP in VRF mode without VRFs (terminate tunnels in global context) – Front door VRF – IPsec anti-replay window size – IPsec preferred peer – Persistent self-signed certificate – Easy VPN remote RSA signature storage • Support was removed for software-based cryptographic mode. • Support was removed for IPsec stateful failover using HSRP and SSP. • Tunnel capacity is increased to 16,000 tunnels. • Support was added for the following commands: – **clear crypto engine accelerator counter** command—Clears platform and network interface controller statistics. – **show crypto engine accelerator statistic** command—Displays platform and network interface controller statistics. • Support for Supervisor Engine 2 was removed. Cisco IOS Release 12.2(33)SXH is supported only by the Supervisor Engine 32 and Supervisor Engine 720.
Cisco IOS Release 12.2(18)SXF2	Support was introduced for the configuration of IP multicast over a GRE tunnel. Note the following changes from previous releases: • The **crypto engine subslot** command has been replaced by the **crypto engine slot** command.
Cisco IOS Release 12.2(18)SXE2	Support for the IPsec VPN SPA was introduced on the Cisco 7600 SSC-400 on the Catalyst 6500 Series switch.

Overview of the IPsec VPN SPA

The IPsec VPN SPA is a Gigabit Ethernet IP Security (IPsec) cryptographic SPA that you can install in a Catalyst 6500 Series switch to provide hardware acceleration for IPsec encryption and decryption, generic routing encapsulation (GRE), and Internet Key Exchange (IKE) key generation.

Note Software-based IPsec features are not supported in any Cisco IOS releases that support the IPsec VPN SPA.

The traditional software-based implementation of IPsec in Cisco IOS supports the entire suite of security protocols including Authentication Header (AH), Encapsulating Security Payload (ESP), and IKE. The resources consumed by these activities are significant and make it difficult to achieve line-rate transmission speeds over secure virtual private networks (VPNs). To address this problem, certain platforms with large VPN bandwidth requirements support bump-in-the-wire (BITW) IPsec hardware modules in conjunction with the hardware forwarding engines. These modules off-load policy enforcement, as well as bulk encryption and forwarding, from the route processor (RP) so that it is not required to look at each packet coming through the switch. This frees up resources that can be used for session establishment, key management, and other features. The IPsec VPN SPA provides a bump-in-the-wire (BITW) IPsec implementation using virtual LANs (VLANs) for a Catalyst 6500 Series switch.

Note BITW is an IPsec implementation that starts egress packet processing after the IP stack has finished with the packet and completes ingress packet processing before the IP stack receives the packet.

The IPsec VPN SPA can use multiple Fast Ethernet or Gigabit Ethernet ports on other Catalyst 6500 Series switch modules to connect to the Internet through WAN routers. The physical ports may be attached to the IPsec VPN SPA through a VLAN called the port VLAN. Packets that are received from the WAN routers pass through the IPsec VPN SPA for IPsec processing. The packets are output on a dedicated VLAN called the iinterface VLAN or inside VLAN. Depending on the configuration mode (VRF mode or crypto-connect mode), the interface VLAN or port VLAN may be configured explicitly or may be allocated implicitly by the system.

On the LAN side, traffic between the LAN ports can be routed or bridged on multiple Fast Ethernet or Gigabit Ethernet ports. Because the LAN traffic is not encrypted or decrypted, it does not pass through the IPsec VPN SPA.

The IPsec VPN SPA does not route, maintain routing information, or change the MAC header of a packet (except for the VLAN ID from one VLAN to another).

Overview of Basic IPsec and IKE Configuration Concepts

This section reviews some basic IPsec and IKE concepts that are used throughout the configuration of the IPsec VPN SPA, such as security associations (SAs), access control lists (ACLs), crypto maps, transform sets, and IKE policies. The information presented here is introductory and should not be considered complete.

Note For more detailed information on IPsec and IKE concepts and procedures, refer to the *Cisco IOS Security Configuration Guide.*

Information About IPsec Configuration

IPsec provides secure tunnels between two peers, such as two routers or switches. More accurately, these tunnels are sets of security associations (SAs) that are established between two IPsec peers. The SAs define which protocols and algorithms should be applied to sensitive packets and specify the keying material to be used by the two peers. SAs are unidirectional and are established per security protocol (Authentication Header (AH) or Encapsulating Security Payload (ESP)). Multiple IPsec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of SAs. For example, some data streams might be authenticated only while other data streams must both be encrypted and authenticated.

Note The use of the term “tunnel” in this subsection does not refer to using IPsec in tunnel mode.

With IPsec, you define what traffic should be protected between two IPsec peers by configuring ACLs and applying these ACLs to interfaces by way of crypto maps. (The ACLs used for IPsec are used only to determine which traffic should be protected by IPsec, not which traffic should be blocked or permitted through the interface. Separate ACLs define blocking and permitting at the interface.)

If you want certain traffic to receive one combination of IPsec protection (for example, authentication only) and other traffic to receive a different combination of IPsec protection (for example, both authentication and encryption), you must create two different crypto ACLs to define the two different types of traffic. These different ACLs are then used in different crypto map entries, which specify different IPsec policies.

Crypto ACLs associated with IPsec crypto map entries have four primary functions:

- Select outbound traffic to be protected by IPsec (permit = protect).
- Indicate the data flow to be protected by the new SAs (specified by a single permit entry) when initiating negotiations for IPsec security associations.
- Process inbound traffic in order to filter out and discard traffic that should have been protected by IPsec.
- Determine whether or not to accept requests for IPsec security associations on behalf of the requested data flows when processing IKE negotiation from the IPsec peer. Negotiation is performed only for ipsec-isakmp crypto map entries. In order to be accepted, if the peer initiates the IPsec negotiation, it must specify a data flow that is “permitted” by a crypto ACL associated with an ipsec-isakmp crypto map entry.

Crypto map entries created for IPsec combine the various parts used to set up IPsec SAs, including:

- Which traffic should be protected by IPsec (per a crypto ACL)
- The granularity of the flow to be protected by a set of SAs
- Where IPsec-protected traffic should be sent (the name of the remote IPsec peer)
- The local address to be used for the IPsec traffic
- What IPsec SA should be applied to this traffic (selecting from a list of one or more transform sets)
- Whether SAs are manually established or are established via IKE
- Other parameters that might be necessary to define an IPsec SA

Crypto map entries are searched in order. The switch attempts to match the packet to the access list specified in that entry.

Crypto map entries also include transform sets. A transform set is an acceptable combination of security protocols, algorithms, and other settings to apply to IPsec-protected traffic.

You can specify multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. During IPsec security association negotiations with IKE, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and will be applied to the protected traffic as part of both peers’ IPsec SAs. (With manually established SAs, there is no negotiation with the peer, so both sides must specify the same transform set.)

Note To minimize the possibility of packet loss during rekeying, we recommend using time-based rather than volume-based IPsec SA expiration. By setting the lifetime volume to the maximum value using the **set security-association lifetime kilobytes 536870912** command, you can usually force time-based SA expiration.

Information About IKE Configuration

IKE is a key management protocol standard that is used in conjunction with the IPsec standard.

IKE is a hybrid protocol that implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.)

In Cisco IOS Release 12.2(33)SXF and earlier releases, IPsec can be configured without IKE, but IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard. IKE is enabled by default.

You configure IKE by creating IKE policies at each peer using the **crypto isakmp policy** command. An IKE policy defines a combination of security parameters to be used during the IKE negotiation and mandates how the peers are authenticated.

You can create multiple IKE policies, each with a different combination of parameter values, but at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

If you do not configure any policies, your router uses the default policy, which is always set to the lowest priority, and which contains each parameter's default value.

There are five parameters to define in each IKE policy:

- Encryption algorithm
- Hash algorithm
- Authentication method
- Diffie-Hellman group identifier
- Security association lifetime

For more information about IKE, see the "Overview of IKE" section on page 24-2.

Configuring VPNs with the IPsec VPN SPA

To configure a VPN using the IPsec VPN SPA, you have two basic options: crypto-connect mode or Virtual Routing and Forwarding (VRF) mode. In either mode, you may also configure GRE tunneling to encapsulate a wide variety of protocol packet types, including multicast packets, inside the VPN tunnel.

Note Switching between crypto-connect mode and VRF mode requires a reload.

Note We recommend that you do not make changes to the VPN configuration while VPN sessions are active. To avoid system disruption, we recommend that you plan a scheduled maintenance time and clear all VPN sessions using the **clear crypto sessions** command before making VPN configuration changes.

Crypto-Connect Mode

Traditionally, VPNs are configured on the IPsec VPN SPA by attaching crypto maps to interface VLANs and then crypto-connecting a physical port to the interface VLAN. This method, known as crypto-connect mode, is similar to the method used to configure VPNs on routers running Cisco IOS software. When you configure VPNs on the IPsec VPN SPA using crypto-connect mode, you attach crypto maps to VLANs (using interface VLANs); when you configure VPNs on switches running Cisco IOS software, you configure individual interfaces.

Note With the IPsec VPN SPA, crypto maps are attached to individual interfaces but the set of interfaces allowed is restricted to interface VLANs.

Crypto-connect mode VPN configuration is described in Chapter 21, “Configuring VPNs in Crypto-Connect Mode.”

VRF Mode

VRF mode, also known as VRF-aware IPsec, allows you to map IPsec tunnels to VPN routing and forwarding instances (VRFs) using a single public-facing address. A VRF instance is a per-VPN routing information repository that defines the VPN membership of a customer site attached to the Provider Edge (PE) router. A VRF comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table. A separate set of routing and CEF tables is maintained for each VPN customer.

When you configure a VPN on the IPsec VPN SPA using VRF mode, the model of interface VLANs is preserved, but the **crypto connect vlan** command is not used. Instead, a route must be installed so that packets destined for that particular subnet in that particular VRF are directed to that interface VLAN.

When configuring a VPN using VRF mode, you have these additional tunneling options: tunnel protection (TP) using GRE, and Virtual Tunnel Interface (VTI). With either of these options, you can terminate tunnels in VRFs (normal VRF mode) or in the global context.

VRF mode VPN configuration is described in Chapter 22, “Configuring VPNs in VRF Mode.”

IPsec Feature Support

The tables in the following sections display supported and unsupported IPsec features of the VSPA in each VPN mode according to the software release:

- IPsec Features Common To All VPN Modes, page 20-7
- IPsec Features in Crypto-Connect Mode, page 20-11
- IPsec Features in VRF Mode, page 20-12

Note This document describes IPsec VPN SPA features and applications that have been tested and are supported. Features and applications that do not explicitly appear in this table and in the following chapters should be considered unsupported. Contact your Cisco account team before implementing a configuration that is not described in this document.

IPsec Features Common To All VPN Modes

Table 20-1 lists the supported and unsupported IPsec features common to all VPN modes.

Table 20-1 IPsec Feature Support By Release in All VPN Modes

Feature Name	Cisco IOS Software Release 12.2					
	SXE	SXF	SRA[1]	SRB, SRC, SRD	SXH	SXI
IPsec tunnels using software crypto	N	N	N	N	N	N
Enhanced GRE takeover (if the supervisor engine cannot process)	Y	Y	Y	Y	Y	Y
Multicast over GRE	N	Y	Y	Y	Y	Y
Multicast over multipoint GRE (mGRE) / DMVPN	N	N	N	N	N	N
Multicast Scalability Enhancement (single SPA mode)	N	Y	Y	Y	Y	N
Advanced Encryption Standard (AES)	Y	Y	Y	Y	Y	Y
ISAKMP keyring	Y	Y	Y	Y	Y	Y
SafeNet Client support	Y	Y	Y	Y	Y	N
Peer filtering (SafeNet Client support)	N	N	N	N	N	N
Certificate to ISAKMP profile mapping	Y	Y	Y	Y	Y	Y
Encrypted preshared key	Y	Y	Y	Y	Y	Y
IKE Aggressive Mode Initiation	N	N	N	N	N	N
Call Admission Control (CAC) for IKE	N	N	Y	Y	Y	Y
Dead Peer Detection (DPD) on-demand	Y	Y	Y	Y	Y	Y
DPD periodic message option	N	N	Y	Y	Y	Y
IPsec prefragmentation (Look-Ahead Fragmentation, or LAF)	Y	Y	Y	Y	Y	Y
Reverse Route Injection (RRI)	Y	Y	Y	Y	Y	Y
Reverse route with optional parameters	N	N	N	N	N	N
Adjustable IPsec anti-replay window size	N	Y	Y	Y	Y	Y
IPsec preferred peer	Y	Y	Y	Y	Y	Y
Per-crypto map (and global) IPsec security association (SA) idle timers	Y	Y	Y	Y	Y	Y
Distinguished name-based crypto maps	Y	Y	Y	Y	Y	Y
Sequenced access control lists (ACLs) (crypto ACLs)	Y	Y	Y	Y	Y	Y
Deny policy configuration enhancements (drop, jump, clear)	Y	Y	Y	Y	Y	Y
Disable volume lifetime per interface	N	N	N	N	N	Y

Table 20-1 IPsec Feature Support By Release in All VPN Modes (continued)

Feature Name	Cisco IOS Software Release 12.2					
	SXE	SXF	SRA[1]	SRB, SRC, SRD	SXH	SXI
IPsec VPN SPA quality of service (QoS) queueing	Y	Y	Y	Y	Y	Y
Multiple RSA key pair support	N	N	Y	Y	Y	Y
Protected private key storage	N	N	Y	Y	Y	Y
Trustpoint CLI	N	N	Y	Y	Y	Y
Query mode per trustpoint	N	N	N	N	N	Y
Local certificate storage location	N	N	Y	Y	Y	Y
Direct HTTP enroll with CA servers	Y	Y	Y	Y	Y	Y
Manual certificate enrollment (TFTP and cut-and-paste)	N	N	Y	Y	Y	Y
Certificate autoenrollment	N	N	Y	Y	Y	Y
Key rollover for Certificate Authority (CA) renewal	N	N	N	N	N	Y
Public-key infrastructure (PKI) query multiple servers	N	N	N	N	N	Y
Online Certificate Status Protocol (OCSP)	N	N	N	N	N	Y
Optional OCSP nonces	N	N	N	N	N	Y
Certificate security attribute-based access control	N	N	N	N	N	Y
PKI AAA authorization using entire subject name	N	N	N	N	N	Y
PKI local authentication using subject name	N	N	Y	Y	Y	Y
Source interface selection for outgoing traffic with certificate authority	N	N	N	N	N	Y
Persistent self-signed certificates as Cisco IOS CA server	N	N	N	N	N	N
Certificate chain verification	N	N	N	N	N	N
Multi-tier certificate support	Y	Y	Y	Y	Y	Y
Easy VPN Server enhanced features	N	N	N	N	N	N
Easy VPN Server basic features	Y	Y	Y	Y	Y	Y
Interoperate with Easy VPN Remote using preshared key	Y	Y	Y	Y	Y	Y
Interoperate with Easy VPN Remote using RSA signature	N	N	Y	Y	Y	Y
Central Policy Push (CPP)	N	N	Y	Y	N	N

Table 20-1 IPsec Feature Support By Release in All VPN Modes (continued)

Feature Name	Cisco IOS Software Release 12.2					
	SXE	SXF	SRA[1]	SRB, SRC, SRD	SXH	SXI
Stateless failover using the Hot Standby Router Protocol (HSRP)	Y	Y	Y	Y	Y	Y
Chassis-to-chassis stateful failover using HSRP and SSP in site-to-site IPsec using preshared keys with crypto maps	Y	Y	N	N	N	N
Chassis-to-chassis failover (IPsec stateful failover) with DMVPN, GRE/TP, VTI, Easy VPN, or PKI	N	N	N	N	N	N
Blade-to-Blade stateful failover	Y	Y	Y	Y	Y	Y
IPsec VPN Monitoring (IPsec Flow MIB)	Y	Y	Y	Y	Y	Y
IPsec VPN Accounting (start / stop / interim records)	Y	Y	Y	Y	Y	Y
Crypto Conditional Debug support	N	Y	Y	Y	Y	Y
show crypto engine accelerator statistic command	N	N	Y	Y	Y	Y
Other **show crypto engine** commands	N	N	N	N	N	N
clear crypto engine accelerator counter command	N	N	Y	Y	Y	Y
Crypto commands applied to a loopback interface	N	N	N	N	N	N
Policy Based Routing (PBR) on tunnel interface or interface VLAN	N	N	N	N	N	N
ACL on tunnel interface	N	N	N	N	N	Y
MQC QoS on tunnel interface (service policy)	N	N	N	N	N	Y
mls qos command on all tunnel interfaces: IPsec, GRE, mGRE	N	N	N	N	N	N
QoS pre-classify CLI	N	N	N	N	N	N
NAT (GRE taken over in crypto-connect mode) on interface VLAN with crypto maps	N	N	N	N	N	N
16 K tunnels (IKE and IPsec tunnels)	N	N	Y	Y	Y	Y
Switching between VRF and crypto-connect modes requires reboot	Y	Y	Y	Y	Y	Y
GRE keepalives on tunnel protection (TP) tunnels	N	N	N	N	N	N
GRE keepalives on mGRE/DMVPN tunnels	N	N	N	N	N	N

Table 20-1 IPsec Feature Support By Release in All VPN Modes (continued)

Feature Name	Cisco IOS Software Release 12.2					
	SXE	SXF	SRA[1]	SRB, SRC, SRD	SXH	SXI
IPsec Network Address Translation Transparency (NAT-T) (transport mode, ESP only)	Y	Y	Y	Y	Y	Y
Dynamic Multipoint VPN Phase 2 (DMVPN) (mGRE; TP & NHRP)	Y	Y	Y	Y	Y	Y
DMVPN Phase 3	N	N	N	N	N	N
DMVPN hub router behind a NAT gateway—tunnel mode	N	N	N	N	N	N
DMVPN hub router behind a NAT gateway—transport mode (not spoke-to-spoke)	N	N	N	N	Y	Y
DMVPN spoke router behind a NAT gateway—tunnel mode	N	N	N	N	N	N
DMVPN spoke router behind a NAT gateway—transport mode (not spoke-to-spoke)	Y	Y	Y	Y	Y	Y
Multicast transit traffic over DMVPN tunnels	N	N	N	N	N	N
Non-IP traffic over TP (DMVPN, point-to-point GRE, sVTI) tunnels	N	N	N	N	N	N
Support for the VPNSM	Y	Y	N	N	N	N
All serial PPP interfaces with crypto-connect mode must have **ip unnumber null 0** command	N	N	N	Y	Y	Y
Manual key	N	Y	N	N	N	N
Tunnel Endpoint Discovery	Y	Y	N	N	N	N
Transport adjacency and nested tunnels	N	N	N	N	N	N
Transit IPsec packets	N	Y	N	N	Y	Y
IPsec VPN SPA supported with virtual switching system (VSS)	N	N	N	N	N	N
IP header options through IPsec tunnels	N	N	N	N	N	N
Invalid SPI recovery	N	N	Y	Y	Y	Y
IPsec compression	N	N	N	N	N	N
Group Encrypted Transport VPN (GETVPN)	N	N	N	N	N	N
IPsec Passive Mode	N	N	N	N	N	N
Multilink or dialer interfaces	N	N	N	N	N	N

Table 20-1 IPsec Feature Support By Release in All VPN Modes (continued)

Feature Name	Cisco IOS Software Release 12.2					
	SXE	SXF	SRA[1]	SRB, SRC, SRD	SXH	SXI
Point-to-point frame relay	Y	Y	Y	Y	Y	Y
Multipoint frame relay	N	N	N	N	N	N

[1] The SR software releases are for the Cisco 7600 series router. These releases do not apply to the Catalyst 6500 series switches.

IPsec Features in Crypto-Connect Mode

Table 20-2 lists the supported and unsupported IPsec features in crypto-connect mode.

Table 20-2 IPsec Feature Support By Release in Crypto-Connect Mode

Feature Name	Cisco IOS Software Release 12.2					
	SXE	SXF	SRA[1]	SRB, SRC, SRD	SXH	SXI
Point-to-point GRE with tunnel protection and VTI	N	N	N	N	N	N
Path MTU discovery (PMTUD)	N	N	Y	Y	Y	Y
PMTUD with NAT-T	N	N	N	N	N	N
IPsec static virtual tunnel interface (sVTI)	N	N	N	N	N	N
The use of VRFs in conjunction with crypto features	N	N	N	N	N	N
IPX and Appletalk over point-to-point GRE	Y	Y	Y	Y	Y	Y
ip tcp adjust-mss command in GRE when taken over	N	N	N	N	N	Y

[1] The SR software releases are for the Cisco 7600 series router. These releases do not apply to the Catalyst 6500 series switches.

IPsec Features in VRF Mode

Table 20-3 lists the supported and unsupported IPsec features in VRF mode.

Table 20-3 IPsec Feature Support By Release in VRF Mode

Feature Name	Cisco IOS Software Release 12.2					
	SXE	SXF	SRA[1]	SRB, SRC, SRD	SXH	SXI
Global VRF	Y	Y	Y	Y	Y	Y
Front-door VRF (FVRF)	N	N	Y	Y	Y	Y
FVRF on an mGRE tunnel configured on a DMVPN hub	N	N	Y	Y	Y	Y
FVRF on an mGRE tunnel configured on a DMVPN spoke	N	N	N	N	N	N
Overlapping IP address space in VRFs	Y	Y	Y	Y	Y	Y
Secondary IP addresses on interfaces	N	N	N	N	N	N
MPLS over GRE/IPsec (tag switching on tunnel interfaces)	N	N	N	N	N	N
PE-PE encryption (IPsec only) over MPLS	N	N	N	N	N	N
PE-PE encryption (tunnel protection) over MPLS	N	N	N	N	N	N
MPLS PE-CE encryption (Tag2IP) with GRE/TP	N	N	N	Y	Y	Y
MPLS PE-CE encryption (Tag2IP) with sVTI	N	N	N	N	N	Y
MPLS PE-CE encryption (Tag2IP) with crypto map	N	N	N	N	N	N
Crypto maps in VRF-lite	Y	Y	Y	Y	Y	Y
Per-VRF AAA with RADIUS	N	N	N	Y	Y	Y
Per-VRF AAA with TACACS	N	N	N	Y	N	N
IPsec static virtual tunnel interface (sVTI)	N	N	Y	Y	Y	Y
Multicast over sVTI	N	N	N	N	N	Y
ip tcp adjust-mss command on sVTI or GRE	N	N	N	N	N	Y
Ingress and egress features (ACL, QOS) on sVTI, GRE/TP, and mGRE tunnel	N	N	N	N	N	Y
Ingress features (ACL, PBR, inbound service policy) on the outside interface	N	N	N	N	N	N
Outbound service policy on the outside interface	Y	Y	Y	Y	Y	Y

Table 20-3 IPsec Feature Support By Release in VRF Mode (continued)

Feature Name	Cisco IOS Software Release 12.2					
	SXE	SXF	SRA[1]	SRB, SRC, SRD	SXH	SXI
TP support in the global context	N	N	Y	Y	Y	Y
IPsec SA using crypto map created in transport mode	N	N	N	N	N	N
Path MTU discovery (PMTUD)	N	N	N	N	N	Y
Non-IP version 4 traffic over TP tunnels	N	N	N	N	N	N
IPv6 IPsec sVTI IPv6-in-IPv6	N	N	N	N	N	N

[1] The SR software releases are for the Cisco 7600 series router. These releases do not apply to the Catalyst 6500 series switches.

Software Requirements

The VSPA requires that one of the following crypto images is running on your switch:

- Supervisor Engine 720 (including 10G)
  - s72033-adventerprisek9_wan-mz
  - s72033-advipservicesk9_wan-mz
  - s72033-adventerprisek9_wan-vz
  - s72033-advipservicesk9_wan-vz
- Supervisor Engine 32 (including 10G)
  - s3223-adventerprisek9_wan-mz
  - s3223-advipservicesk9_wan-mz
  - s3223-adventerprisek9_wan-vz
  - s3223-advipservicesk9_wan-vz

Note The images ending in “-vz” require Cisco IOS Release 12.2(33)SXH or a later release.

Interoperability

The supervisor engine support varies based on the release. Table 20-4 lists the supervisor engine support for each release.

Table 20-4 Supervisor Engine Support for the IPsec VPN SPA by Release

		Cisco IOS Release 12.2		
Supervisor	**Description**	SXF2	SXH	SXI
WS-SUP720-3B	Supervisor 720 Fabric MSFC3 PFC3B	Y	Y	Y
WS-SUP720-3BXL	Supervisor 720 Fabric MSFC3 PFC3BXL	Y	Y	Y
VS-S720-10G-3C	Supervisor 720 with 2 ports 10GbE MSFC3 PFC3C	N	Y	Y
VS-S720-10G-3CXL	Supervisor 720 with 2 ports 10GbE MSFC3 PFC3CXL	N	Y	Y
WS-SUP32-GE-3B	Supervisor 32 with 8 GbE uplinks and PFC3B	Y	Y	Y
WS-SUP32-10GE-3B	Supervisor 32 with 2 ports 10GbE and PFC3B	N	Y	Y
WS-S32-GE-PISA	Supervisor 32 with PISA and 8 GbE uplinks	N	N	N
WS-S32-10GE-PISA	Supervisor 32 with PISA and 2 ports 10GbE	N	N	N
WS-X6K-S2-MSFC2	Supervisor Engine 2 with 2GbE and MSFC-2/PFC-2	N	N	N

The IPsec VPN SPA supports the following interoperability features:

- You may have an IPsec VPN SPA in the same chassis with the following service modules:
  - Firewall Services Module (WS-SVC-FWM-1-K9)
  - Network Analysis Module 2 (WS-SVC-NAM-2)

Table 20-5 lists the SIP and SSC support for each release.

Table 20-5 SIP and SSC Support for the IPsec VPN SPA by Release

	Cisco IOS Release 12.2		
Line Card or Module	**SXF**	**SXH**	**SXI**
7600-SIP-200	Y	Y	Y
7600-SIP-400	Y	Y	Y
7600-SIP-600	N	N[1]	Y
7600-SSC-400	Y	Y	Y

[1] Platform support for the 7600-SIP-600 is removed in Cisco IOS Release 12.2(33)SXH and restored in Cisco IOS Release 12.2(33)SXI and later releases.

The line card module support varies based on the release. Table 20-6 lists the Ethernet line card and module support for each release.

Table 20-6 Ethernet Line Card and Module Support for the IPsec VPN SPA by Release

Line Card or Module	Cisco IOS Release 12.2		
	SXF	SXH	SXI
SPA-1X10GE	N	N	SIP-600
SPA-10X1GE[1]	N	N	SIP-600
SPA-2X1GE	SIP-400	SIP-400	SIP-400
SPA-2XT3/E3	N	N	N
SPA-4X1FE-TX-V2	N	N	N
SPA-5X1GE[1]	N	N	SIP-600
SPA-5X1GE-V2	N	N	N
SPA-8X1FE-TX-V2	N	N	N
WS-X6148-GE-TX	Y	Y	Y
WS-X6148-RJ-21	Y	Y	Y
WS-X6148-RJ-21V	Y	Y	Y
WS-X6148-RJ-45	Y	Y	Y
WS-X6148-RJ-45V	Y	Y	Y
WS-X6408A-GBIC	Y	Y	Y
WS-X6416-GBIC	Y	Y	Y
WS-X6502-10GE	Y	Y	Y
WS-X6516-GBIC	Y	Y	Y
WS-X6516-GE-TX	Y	Y	Y
WS-X6516A-GBIC	Y	Y	Y
WS-X6548-GE-TX	Y	Y	Y
WS-X6548-RJ-45	Y	Y	Y
WS-X6704-10GE	Y	Y	Y
WS-X6708-10GE	N	Y	Y
WS-X6716-10GE	N	Y	Y
WS-X6748-GE-TX	Y	Y	Y
WS-X6748-SFP	Y	Y	Y

[1] Subinterfaces on SPA-5X1GE and SPA-10X1GE are not supported in any release.

Table 20-7 lists the ATM line card and module support for each release.

Table 20-7 ATM Line Card and Module Support for the IPsec VPN SPA by Release

Line Card or Module	Cisco IOS Release 12.2		
	SXF	SXH	SXI
SPA-1XCHSTM1/OC3	N	N	N
SPA-1XOC48-ATM	SIP-400	N	SIP-400
SPA-2XOC3-ATM	SIP-200 SIP-400	N	SIP-200 SIP-400
SPA-4XOC3-ATM	N	N	N

Table 20-8 lists the POS line card and module support for each release.

Table 20-8 POS Line Card and Module Support for the IPsec VPN SPA by Release

Line Card or Module	Cisco IOS Release 12.2		
	SXF	SXH	SXI
SPA-1XOC12-POS	N	SIP-400	SIP-400
SPA-1XOC48POS/RPR	N	N	N
SPA-2XOC3-POS	SIP-200 SIP-400	SIP-200 SIP-400	SIP-200 SIP-400
SPA-OC192POS-XFP	N	N	SIP-600

Table 20-9 lists the serial line cards and module support for each release.

Table 20-9 Serial Line Card and Module Support for the IPsec VPN SPA by Release

Line Card or Module	Cisco IOS Release 12.2		
	SXF	SXH	SXI
SPA-2XCT3/DS0	N	SIP-200	SIP-200 SIP-400
SPA-2XT3/E3	N	N	N
SPA-4XCT3/DS0	N	N	N
SPA-4XT3/E3	N	N	N
SPA-8XCHT1/E1	N	N	N
WS-6182-2PA	Y	N	N

Table 20-9 Serial Line Card and Module Support for the IPsec VPN SPA by Release (continued)

Line Card or Module	Cisco IOS Release 12.2		
	SXF	SXH	SXI
WS-6802-2PA	N	N	N
WS-X6582-2PA With the following PAs: PA-A3-OC3MM PA-POS-OC3MM PA-POS-2OC3 PA-MC-2T3+ PA-1FE-TX[1] PA-2FE-TX[1]	Y	Y	Y

[1] Subinterfaces on PA-1FE-TX and PA-2FE-TX are not supported in releases earlier than Cisco IOS Release 12.2(33)SXI2.

Table 20-10 lists the service module support for each release.

Table 20-10 Service Module Support for the IPsec VPN SPA by Release

Line Card or Module	Cisco IOS Release 12.2		
	SXF	SXH	SXI
WS-SVC-FWSM-1	Y	Y	Y
WS-SVC-IDSM2	N	N	N
WS-SVC-NAM2	N	N	Y

Table 20-11 lists the OSM line card support for each release.

Note Cisco IOS Release 12.2(33)SXH and later releases do not support OSM modules.

Table 20-11 OSM Line Card and Module Support for the IPsec VPN SPA by Release

Line Card or Module	Cisco IOS Release 12.2		
	SXF	SXH	SXI
OSM-2OC48/1DPT-SI	Y	N	N
OSM-2OC48/1DPT-SL	Y	N	N
OSM-2OC48/1DPT-SS	Y	N	N
OSM-8OC3-POS-MM	Y	N	N
OSM-8OC3-POS-SI	Y	N	N
OSM-8OC3-POS-SI+	Y	N	N
OSM-8OC3-POS-SL	Y	N	N
OSM-16OC3-POS-MM+	Y	N	N

Table 20-11 OSM Line Card and Module Support for the IPsec VPN SPA by Release (continued)

Line Card or Module	Cisco IOS Release 12.2		
	SXF	SXH	SXI
OSM-16OC3-POS-SI	Y	N	N
OSM-16OC3-POS-SI+	Y	N	N
OSM-16OC3-POS-SL	Y	N	N
OSM-2+4GE-WAN+	Y	N	N

Restrictions

Note For other SSC-specific features and restrictions see also Chapter 3, "Overview of the SIPs and SSC"in this guide.

The IPsec VPN SPA is subject to the following restrictions:

- The IPsec VPN SPA requires Cisco IOS Release 12.2(18)SXE2 or later releases.
- The IPsec VPN SPA is supported only on the Cisco 7600 SSC-400.
- A Supervisor Engine 720 (MSFC3 and PFC3) requires a minimum of 512 MB memory to operate with the IPsec VPN SPA.

  Note The IPsec VPN SPA MSFC DRAM requirements are as follows:

  – Up to 8,000 tunnels with 512-MB DRAM
  – Up to 16,000 tunnels with 1-GB DRAM

  These numbers are chosen to leave some memory available for routing protocols and other applications. However, your particular use of the MSFC may demand more memory than the quantities that are listed above. In an extreme case, you could have one tunnel but still require 512-MB DRAM for other protocols and applications running on the MSFC.

- A maximum of 10 IPsec VPN SPAs per chassis are supported.
- IPsec VPN SPA state information is not maintained between the active and standby supervisor engine during normal operation. During a supervisor engine switchover in an SSO environment, the IPsec VPN SPA will reboot.

Note In Cisco IOS Release 12.2(18)SXF2 and later releases, the **crypto engine subslot** command used in previous releases has been replaced with the **crypto engine slot** command (of the form **crypto engine slot** *slot/subslot* {**inside** | **outside**}). The **crypto engine subslot** command is no longer supported. In Cisco IOS Release 12.2(33)SXI and later releases, the **slot** *slot/subslot* is not specified when the **outside** keyword is used.

When upgrading, ensure that this command has been modified in your start-up configuration to avoid extended maintenance time.

Supported MIBs

The following MIB is supported in Cisco IOS Release 12.2(18)SXE2 for the Cisco 7600 SSC-400 and the IPsec VPN SPA on a Catalyst 6500 Series switch:

- CISCO-IPSEC-FLOW-MONITOR-MIB

Note Gigabit Ethernet port SNMP statistics (for example, ifHCOutOctets and ifHCInOctets) are not provided for the internal IPsec VPN SPA trunk ports because these ports are not externally operational ports and are used only for configuration.

For more information about MIB support on a Catalyst 6500 Series switch, refer to the *Cisco 7600 Series Router MIB Specifications Guide*, at the following URL:

http://www.cisco.com/en/US/docs/routers/7600/technical_references/7600_mib_guides/MIB_Guide_ver_6/mibgde6.html

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

IPsec VPN SPA Hardware Configuration Guidelines

The configuration guidelines for IPsec VPN SPA hardware are as follows:

- For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.
- Some CLI commands require you to specify the inside and outside ports of the VSPA in the format *slot/subslot/port*. Although the VSPA ports are not actual Gigabit Ethernet ports, and do not share all properties of external Gigabit Ethernet interfaces, they can be addressed for configuration as Gigabit Ethernet trunk ports, using port numbers as follows:
  - Port 1—Inside port, attached to interface VLAN
  - Port 2—Outside port, attached to port VLAN

  For example, to configure the outside port of a VSPA in the first subslot (subslot 0) of an Cisco 7600 SSC-400 in slot 6 of a Catalyst 6500 Series switch, enter the following command:

  ```
  Router(config)# interface GigabitEthernet6/0/2
  ```

- The **show crypto engine configuration** command does not show the IPsec VPN SPA subslot number when there is no crypto connection even if the adapter is installed in the chassis.
- When you remove an IPsec VPN SPA that has some ports participating in crypto connections, the crypto configuration remains intact. When you reinsert the same type of IPsec VPN SPA into the same slot, the crypto connections will be reestablished. To move the IPsec VPN SPA to a different slot, you must first manually remove the crypto connections before removing the IPsec VPN SPA. You can enter the **no crypto connect vlan** command from any interface when the associated physical port is removed.
- When you reboot an IPsec VPN SPA that has crypto connections, the existing crypto configuration remains intact. The crypto connections will be reestablished after the IPsec VPN SPA reboots. When a crypto connection exists but the associated interface VLAN is missing from the IPsec VPN SPA inside port, the crypto connection is removed after the IPsec VPN SPA reboots.
- When you remove a port VLAN or an interface VLAN with the **no interface vlan** command, the associated crypto connection is also removed.

Displaying the SPA Hardware Type

There are several commands on the Catalyst 6500 Series switch that provide IPsec VPN SPA hardware information.

- To verify the SPA hardware type that is installed in your switch, use the **show module** command.
- To display hardware information for the IPsec VPN SPA, use the **show crypto eli** command.

For more information about these commands, see the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX*.

Example of the show module Command

The following example shows output from the **show module** command on a Catalyst 6500 Series switch with an IPsec VPN SPA installed in subslot 0 of a Cisco 7600 SSC-400 that is installed in slot 4:

```
Router# show module 4
Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------
  4    0  2-subslot Services SPA Carrier-400     7600-SSC-400       JAB1104013N

Mod MAC addresses                       Hw    Fw           Sw           Status
--- ---------------------------------- ------ ------------ ------------ -------
  4  001a.a1aa.95f0 to 001a.a1aa.962f   2.0   12.2(33)SXH  12.2(33)SXH  Ok

Mod  Sub-Module                  Model              Serial       Hw     Status
---- --------------------------- ------------------ ----------- ------- -------
 4/0 2 Gbps IPSec SPA            SPA-IPSEC-2G       JAB1048075L  1.0    Ok

Mod  Online Diag Status
---- -------------------
  4  Pass
 4/0 Pass
```

Example of the show crypto eli Command

The following example shows output from the **show crypto eli** command on a Catalyst 6500 Series switch with IPsec VPN SPAs installed in subslots 0 and 1 of a Cisco 7600 SSC-400 that is installed in slot 3. The output displays how many IKE-SAs and IPsec sessions are active and how many Diffie-Hellman keys are in use for each IPsec VPN SPA.

```
Router# show crypto eli

Hardware Encryption : ACTIVE
Number of hardware crypto engines = 2

CryptoEngine SPA-IPSEC-2G[3/0] details: state = Active
Capability      :
    IPSEC: DES, 3DES, AES, RSA

IKE-Session   :     0 active, 16383 max, 0 failed
DH            :     0 active,  9999 max, 0 failed
IPSec-Session :     0 active, 65534 max, 0 failed

CryptoEngine SPA-IPSEC-2G[3/1] details: state = Active
Capability      :
    IPSEC: DES, 3DES, AES, RSA

IKE-Session   :     1 active, 16383 max, 0 failed
DH            :     0 active,  9999 max, 0 failed
IPSec-Session :     2 active, 65534 max, 0 failed
```

CHAPTER 21

Configuring VPNs in Crypto-Connect Mode

This chapter provides information about configuring IPsec VPNs in crypto-connect mode, one of the two VPN configuration modes supported by the IPsec VPN SPA. For information on the other VPN mode, Virtual Routing and Forwarding (VRF) mode, see Chapter 22, "Configuring VPNs in VRF Mode."

This chapter includes the following topics:

For general information on configuring IPsec VPNs with the IPsec VPN SPA, see the "Overview of Basic IPsec and IKE Configuration Concepts" section on page 20-3.

Note The procedures in this chapter assume you have familiarity with security configuration concepts, such as VLANs, ISAKMP policies, preshared keys, transform sets, access control lists, and crypto maps. For detailed information on configuring these features, refer to the following Cisco IOS documentation:

Cisco IOS Security Configuration Guide, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

Cisco IOS Security Command Reference, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

For additional information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* and the related Cisco IOS Release 12.2 software configuration guide and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Tip To ensure a successful configuration of your VPN using the IPsec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Configuring Ports in Crypto-Connect Mode

Before beginning your crypto-connect mode port configurations, you should read the following subsections:

- Understanding Port Types in Crypto-Connect Mode, page 21-2

- Crypto-Connect Mode Configuration Guidelines and Restrictions, page 21-4

Then perform the procedures in the following subsections:

- Configuring the IPsec VPN SPA Inside Port and Outside Port, page 21-6
- Configuring an Access Port, page 21-7
- Configuring a Routed Port, page 21-11
- Configuring a Trunk Port, page 21-14
- Configuring IPsec VPN SPA Connections to WAN Interfaces, page 21-19
- Displaying the VPN Running State, page 21-20

Note The configuration procedures in this section do not provide GRE tunneling support. For information on how to configure GRE tunneling support in crypto-connect mode, see the "Configuring GRE Tunneling in Crypto-Connect Mode" section on page 21-20.

Note The procedures in this section do not provide detailed information on configuring the following Cisco IOS features: IKE policies, preshared key entries, Cisco IOS ACLs, and crypto maps. For detailed information on configuring these features, refer to the following Cisco IOS documentation:

Cisco IOS Security Configuration Guide, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

Cisco IOS Security Command Reference, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

Understanding Port Types in Crypto-Connect Mode

To configure IPsec VPNs in crypto-connect mode, you should understand the following concepts:

- Switch Outside Ports and Inside Ports, page 21-2
- IPsec VPN SPA Outside Port and Inside Port, page 21-3
- Port VLAN and Interface VLAN, page 21-3
- Access Ports, Trunk Ports, and Routed Ports, page 21-4

Switch Outside Ports and Inside Ports

The Fast Ethernet or Gigabit Ethernet ports on the Catalyst 6500 Series switch that connect to the WAN routers are referred to as switch outside ports. These ports connect the LAN to the Internet or to remote sites. Cryptographic policies are applied to the switch outside ports.

The Fast Ethernet or Gigabit Ethernet ports on the Catalyst 6500 Series switch that connect to the LAN are referred to as switch inside ports.

The IPsec VPN SPA sends encrypted packets to the switch outside ports and decrypted packets to the Policy Feature Card (PFC) for Layer 3 forwarding to the switch inside ports.

IPsec VPN SPA Outside Port and Inside Port

The IPsec VPN SPA appears to the CLI as a SPA with two Gigabit Ethernet ports. The IPsec VPN SPA has no external connectors; the Gigabit Ethernet ports connect the IPsec VPN SPA to the switch backplane and Switch Fabric Module (SFM) (if installed).

One Gigabit Ethernet port handles all the traffic going to and coming from the switch outside ports. This port is referred to as the IPsec VPN SPA outside port. The other Gigabit Ethernet port handles all traffic going to and coming from the LAN or switch inside ports. This port is referred to as the IPsec VPN SPA inside port.

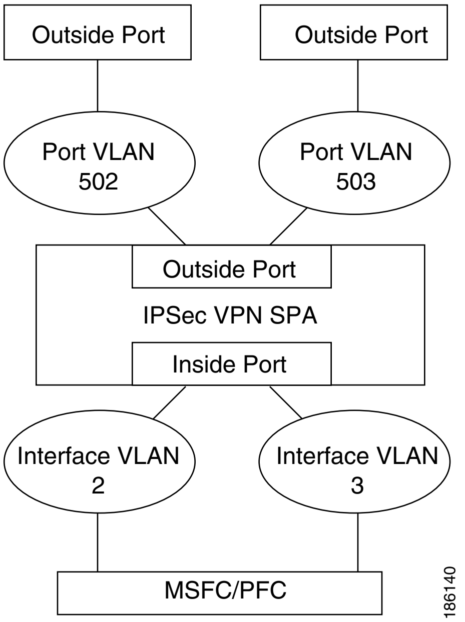
Port VLAN and Interface VLAN

Your VPN configuration can have one or more switch outside ports. To handle the packets from multiple switch outside ports, you must direct the packets from multiple switch outside ports to the IPsec VPN SPA outside port by placing the switch outside ports in a VLAN with the outside port of the IPsec VPN SPA. This VLAN is referred to as the port VLAN. The port VLAN is a Layer 2-only VLAN. You do not configure Layer 3 addresses or features on this VLAN; the packets within the port VLAN are bridged by the PFC.

Before the switch can forward the packets using the correct routing table entries, the switch needs to know which interface a packet was received on. For each port VLAN, you must create another VLAN so that the packets from every switch outside port are presented to the switch with the corresponding VLAN ID. This VLAN contains only the IPsec VPN SPA inside port and is referred to as the interface VLAN. The interface VLAN is a Layer 3-only VLAN. You configure the Layer 3 address and Layer 3 features, such as ACLs and the crypto map, to the interface VLAN.

You associate the port VLAN and the interface VLAN together using the **crypto engine** *slot* command on the interface VLAN followed by the **crypto connect vlan** command on the port VLAN. Figure 21-1 shows an example of the port VLAN and interface VLAN configurations.

Figure 21-1 Port VLAN and Interface VLAN Configuration Example



Port VLAN 502 and port VLAN 503 are the port VLANs that are associated with two switch outside ports.

Interface VLAN 2 and interface VLAN 3 are the interface VLANs that correspond to port VLAN 502 and port VLAN 503, respectively.

You configure the IP address, ACLs, and crypto map that apply to one switch outside port on interface VLAN 2. You configure the features that apply to another switch outside port on interface VLAN 3.

Packets coming from the WAN through the switch outside port belonging to VLAN 502 are directed by the PFC to the IPsec VPN SPA outside port. The IPsec VPN SPA decrypts the packets and changes the VLAN to interface VLAN 2 and then presents the packet to the switch through the IPsec VPN SPA inside port. The PFC then routes the packet to the proper destination.

Packets going from the LAN to the outside ports are first routed by the PFC. Based on the route, the PFC routes the packets to one of the interface VLANs and directs the packet to the IPsec VPN SPA inside port. The IPsec VPN SPA applies the cryptographic policies that are configured on the corresponding interface VLAN, encrypts the packet, changes the VLAN ID to the corresponding port VLAN, and sends the packet to the switch outside port through the IPsec VPN SPA outside port.

Access Ports, Trunk Ports, and Routed Ports

When you configure VPNs on the IPsec VPN SPA using crypto-connect mode, you attach crypto maps to interface VLANs. Using the **crypto connect vlan** command, you then attach an interface VLAN either to a Layer 2 port VLAN associated with one or more physical ports, or directly to a physical port. The physical ports can be ATM, POS, serial, or Ethernet ports.

When you crypto-connect an interface VLAN to a port VLAN that is attached to one or more Ethernet ports configured in switchport mode, the Ethernet ports can be configured as either access ports or trunk ports:

- Access ports—Access ports are switch ports that have an external or VLAN Trunk Protocol (VTP) VLAN associated with them. You can associate more than one port to a defined VLAN.
- Trunk ports—Trunk ports are switch ports that carry many external or VTP VLANs, on which all packets are encapsulated with an 802.1Q header.

When you crypto-connect an interface VLAN to a physical Ethernet port without defining a port VLAN, a hidden port VLAN is automatically created and associated with the port. In this configuration, the Ethernet port is a routed port:

- Routed ports—By default, every Ethernet port is a routed port until it is configured as a switch port. A routed port may or may not have an IP address assigned to it, but its configuration does not include the **switchport** command.

Crypto-Connect Mode Configuration Guidelines and Restrictions

Follow these guidelines and restrictions to prevent IPsec VPN SPA misconfigurations when configuring VPN ports in crypto-connect mode:

- Ethernet ports installed in a Cisco 7600 SIP-400 in the chassis cannot be configured as switch ports.
- Removing a line in a crypto ACL causes all crypto maps using that ACL to be removed and reattached to the IPsec VPN SPA. This action causes intermittent connectivity problems for all the security associations (SAs) derived from the crypto maps that reference that ACL.
- A loopback interface can be used as tunnel source address.

- Do not attach a crypto map set to a loopback interface. However, you can maintain an IPsec security association database independent of physical ingress and egress interfaces with the IPsec VPN SPA by entering the **crypto map local-address** command.

  If you apply the same crypto map set to each secure interface and enter the **crypto map local-address** command with the interface as a loopback interface, you will have a single security association database for the set of secure interfaces. If you do not enter the **crypto map local-address** command, the number of IKE security associations is equal to the number of interfaces attached.
- You can attach a crypto map to an interface VLAN that is associated with a GRE/IPsec tunnel, but do not attach a crypto map to an interface VLAN that is associated with a DMVPN tunnel.
- A crypto map local address (for example, the interface VLAN address if the crypto map is applied to the interface VLAN) can share the same address as the GRE/IPsec tunnel source address, but it cannot share the same address as a DMVPN tunnel source address.
- You can attach the same crypto map to multiple interfaces only if the interfaces are all bound to the same crypto engine.
- If you configure a crypto map with an empty ACL (an ACL that is defined but has no lines) and attach the crypto map to an interface, all traffic goes out of the interface in the clear (unencrypted) state.
- Do not convert existing crypto-connected port characteristics. When the characteristics of a crypto-connected access port or a routed port change (switch port to routed port or vice versa), the associated crypto connection is deleted.
- Do not remove the interface VLAN or port VLAN from the VLAN database. All interface VLANs and port VLANs must be in the VLAN database. When you remove these VLANs from the VLAN database, the running traffic stops.

  When you enter the **crypto connect vlan** command and the interface VLAN or port VLAN is not in the VLAN database, this warning message is displayed:

  ```
  VLAN id 2 not found in current VLAN database. It may not function correctly unless
  VLAN 2 is added to VLAN database.
  ```
- When replacing a crypto map on an interface, always enter the **no crypto map** command before reapplying a crypto map on the interface.
- After a supervisor engine switchover, the installed SPAs reboot and come back online. During this period, the IPsec VPN SPA’s established security associations (SAs) are temporarily lost and are reconstructed after the SPA comes back online. The reconstruction is through IKE (it is not instantaneous).
- Crypto ACLs support only the EQ operator. Other operators, such as GT, LT, and NEQ, are not supported.

Note When configuring a permit policy for multiple ports with the EQ operator, you must use multiple lines as in this example:

```
permit ip any any port eq 300
permit ip any any port eq 400
permit ip any any port eq 600
```

In Cisco IOS Release 12.2(33)SXH1 and later releases, when configuring a deny policy for multiple ports with the EQ operator, you can use commas to declare the ports as in this example:

```
deny ip any any port eq 300,400,600
```

- Noncontiguous subnets in a crypto ACL, as in the following example, are not supported:

```
deny ip   10.0.5.0   0.255.0.255   10.0.175.0   0.255.0.255
deny ip   10.0.5.0   0.255.0.255   10.0.176.0   0.255.0.255
```

- ACL counters are not supported for crypto ACLs.
- An egress ACL is not applied to packets generated by the route processor. An ingress ACL is not applied to packets destined for the route processor.
- Do not apply an IP ACL to the crypto-connect interface or port VLAN. Instead, you can apply IP ACLs to the interface VLAN, as in the following example:

```
interface GigabitEthernet1/2
 ! switch outside port
 switchport
 switchport access vlan 502
 switchport mode access
 ip access-group TEST_INBOUND in  <--- do not apply IP ACL here
!
interface Vlan2
 ! interface VLAN
 ip address 11.0.0.2 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0
 ip access-group TEST_INBOUND in  <--- apply IP ACL here
!
interface Vlan502
 ! port VLAN
 no ip address
 crypto connect vlan 2
 ip access-group TEST_INBOUND in  <--- do not apply IP ACL here
!
```

Note An IP ACL on the interface VLAN will not block inbound encrypted traffic from reaching the VSPA, but can prevent traffic from being routed further after decryption.

Supported and Unsupported Features in Crypto-Connect Mode

A list of the supported and unsupported features in crypto-connect mode can be found in the "IPsec Feature Support" section on page 20-6.

Configuring the IPsec VPN SPA Inside Port and Outside Port

In most cases, you do not explicitly configure the IPsec VPN SPA inside and outside ports. Cisco IOS software configures these ports automatically.

IPsec VPN SPA Inside and Outside Port Configuration Guidelines and Restrictions

When configuring the IPsec VPN SPA inside and outside ports, follow these guidelines:

- Do not change the port characteristics of the IPsec VPN SPA inside or outside port unless it is necessary to set the trusted state. Cisco IOS software configures the ports automatically.

Note Although the default trust state of the inside port is trusted, certain global settings may cause the state to change. To preserve the ToS bytes for VPN traffic in both directions, configure the **mls qos trust** command on both the inside and outside ports to set the interface to the trusted state. For information on the **mls qos trust** command, see the "Module QoS Configuration Guidelines and Restrictions" section on page 25-17.

If you accidentally change the inside port characteristics, enter the following commands to return the port characteristics to the defaults:

```
Router(config-if)# switchport
Router(config-if)# no switchport access vlan
Router(config-if)# switchport trunk allowed vlan 1,1002-1005
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# mtu 9216
Router(config-if)# flow control receive on
Router(config-if)# flow control send off
Router(config-if)# span portfast trunk
```

- Do not configure allowed VLANs on the inside trunk port. Cisco IOS software configures the VLAN list on the inside port automatically based on the **crypto engine slot** command. These VLANs are visible in the port configuration using the **show run** command.
- Do not configure allowed VLANs on the outside trunk port. Cisco IOS software configures these VLANs automatically as hidden VLANs. These VLANs are not visible in the port configuration using the **show run** command.
- Do not remove a VLAN from the IPsec VPN SPA inside port. The running traffic stops when you remove an interface VLAN from the IPsec VPN SPA inside port while the crypto connection to the interface VLAN exists. The crypto connection is not removed and the **crypto connect vlan** command still shows up in the **show running-config** command display. If you enter the **write memory** command with this running configuration, your startup-configuration file would be misconfigured.

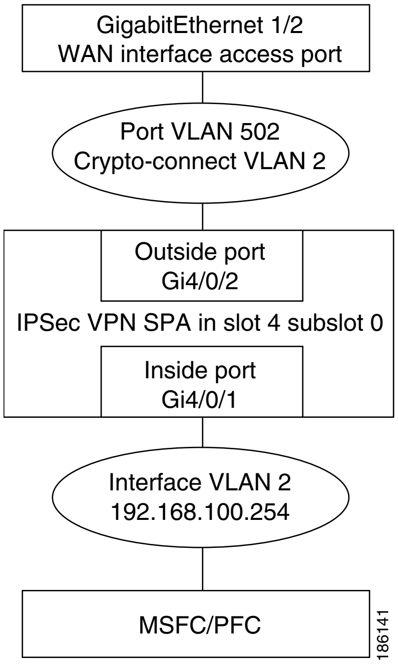
Note It is not possible to remove an interface VLAN from the IPsec VPN SPA inside port while the crypto connection to the interface VLAN exists. You must first remove the crypto connection.

- Do not remove a VLAN from the IPsec VPN SPA outside port. The running traffic stops when you remove a port VLAN from the IPsec VPN SPA outside port while the crypto connection to the interface VLAN exists. The crypto connection is not removed and the **crypto connect vlan** command still shows up in the **show running-config** command display. Removing a VLAN from the IPsec VPN SPA outside port does not affect anything in the startup-configuration file because the port VLAN is automatically added to the IPsec VPN SPA outside port when the **crypto connect vlan** command is entered.

Configuring an Access Port

This section describes how to configure the IPsec VPN SPA with an access port connection to the WAN router (see Figure 21-2).

Figure 21-2 Access Port Configuration Example



Note Ethernet ports installed in a Cisco 7600 SIP-400 in the chassis cannot be configured as switch ports.

To configure an access port connection to the WAN router, perform the following task beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)# crypto isakmp policy priority`
	`...`
	`Router(config-isakmp) # exit`
		Defines an ISAKMP policy and enters ISAKMP policy configuration mode.
		• *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
		For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.

Step 2	`Router(config)# crypto isakmp key keystring address peer-address`	Configures a preshared authentication key. • *keystring*—Preshared key. • *peer-address*—IP address of the remote peer. For details on configuring a preshared key, see the *Cisco IOS Security Configuration Guide*.

	Command	Purpose
Step 3	Router(config)# **crypto ipsec transform-set** *transform-set-name* *transform1*[*transform2*[*transform3*]] ... Router(config-crypto-tran)# **exit**	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. For accepted *transformx* values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference*.
Step 4	Router(config)# **access list** *access-list-number* {**deny** | **permit**} **ip** *source source-wildcard destination destination-wildcard*	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Address of the host from which the packet is being sent. • *source-wildcard*—Wildcard bits to be applied to the source address. • *destination*—Address of the host to which the packet is being sent. • *destination-wildcard*—Wildcard bits to be applied to the destination address. For details on configuring an access list, see the *Cisco IOS Security Configuration Guide*.
Step 5	Router(config)# **crypto map** *map-name seq-number* **ipsec-isakmp** ... Router(config-crypto-map)# **exit**	Creates or modifies a crypto map entry and enters the crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations. For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.
Step 6	Router(config)# **vlan** *inside-vlan-id*	Adds the VLAN ID into the VLAN database. • *inside-vlan-id*—VLAN identifier.
Step 7	Router(config)# **vlan** *outside-vlan-id*	Adds the VLAN ID into the VLAN database. • *outside-vlan-id*—VLAN identifier.

	Command	Purpose
Step 8	Router(config)# **interface vlan** *inside-vlan-id*	Enters interface configuration mode for the specified VLAN interface. • *inside-vlan-id*—VLAN identifier.
Step 9	Router(config-if)# **description inside_interface_vlan_for_crypto_map**	(Optional) Adds a comment to help identify the interface.
Step 10	Router(config-if)# **ip address** *address mask*	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 11	Router(config-if)# **crypto map** *map-name*	Applies a previously defined crypto map set to the interface. • *map-name*—Name that identifies the crypto map set. Enter the *map-name* value you created in Step 5.
Step 12	Router(config-if)# **no shutdown**	Enables the interface as a Layer 3 inside interface VLAN.
Step 13	Router(config-if)# **crypto engine slot** *slot/subslot*	Assigns the crypto engine to the crypto interface VLAN. • *slot/subslot*—Enter the slot and subslot where the IPsec VPN SPA is located.
Step 14	Router(config)# **interface vlan** *outside-vlan-id*	Enters interface configuration mode for the specified VLAN interface. • *outside-vlan-id*—VLAN identifier.
Step 15	Router(config-if)# **description outside_access_vlan**	(Optional) Adds a comment to help identify the interface.
Step 16	Router(config-if)# **no shutdown**	Enables the interface as an outside access port VLAN.
Step 17	Router(config-if)# **crypto connect vlan** *inside-vlan-id*	Connects the outside access port VLAN to the inside interface VLAN and enters crypto-connect mode. • *inside-vlan-id*—VLAN identifier.
Step 18	Router(config-if)# **interface gigabitethernet** *slot/subslot/port*	Enters interface configuration mode for the secure port.
Step 19	Router(config-if)# **description outside_secure_port**	(Optional) Adds a comment to help identify the interface.
Step 20	Router(config-if)# **switchport**	Configures the interface for Layer 2 switching.
Step 21	Router(config-if)# **switchport access vlan** *outside-vlan-id*	Specifies the default VLAN for the interface. • *outside-vlan-id*—VLAN identifier.
Step 22	Router(config-if)# **exit**	Exits interface configuration mode.

For access port configuration examples, see the "Access Port in Crypto-Connect Mode Configuration Example" section on page 21-28.

Verifying the Access Port Configuration

To verify an access port configuration, enter the **show crypto vlan** command.

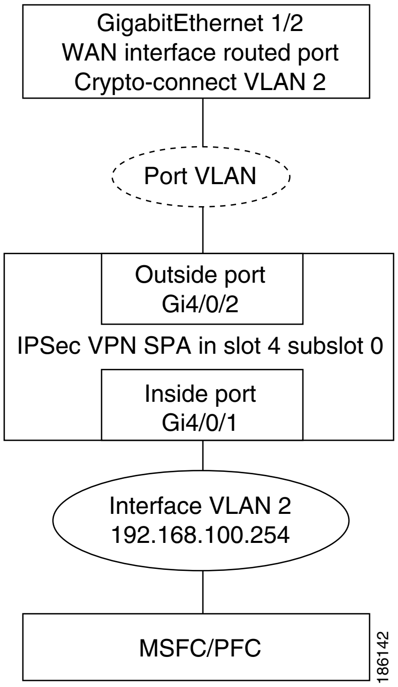
```
Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port Gi4/0/1 connected to VLAN 502 with crypto
map set MyMap
```

Configuring a Routed Port

This section describes how to configure the IPsec VPN SPA with a routed port connection to the WAN router (see Figure 21-3).

Note When a routed port without an IP address is crypto-connected to an interface VLAN, a hidden port VLAN is created automatically. This port VLAN is not explicitly configured by the user and does not appear in the running configuration.

Figure 21-3 Routed Port Configuration Example



Routed Port Configuration Guidelines

When configuring a routed port using the IPsec VPN SPA, follow these configuration guidelines:

- When a routed port has a crypto connection, IP ACLs cannot be attached to the routed port. Instead, you can apply IP ACLs to the attached interface VLAN.
- Unlike an access port or trunk port, the routed port does not use the **switchport** command in its configuration.

To configure a routed port connection to the WAN router, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto isakmp policy** *priority* ... Router(config-isakmp) # **exit**	Defines an ISAKMP policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest. For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.
Step 2	Router(config)# **crypto isakmp key** *keystring* **address** *peer-address*	Configures a preshared authentication key. • *keystring*—Preshared key. • *peer-address*—IP address of the remote peer. For details on configuring a preshared key, see the *Cisco IOS Security Configuration Guide*.
Step 3	Router(config)# **crypto ipsec transform-set** *transform-set-name* *transform1*[*transform2*[*transform3*]] ... Router(config-crypto-tran)# **exit**	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. For accepted *transformx* values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference*.
Step 4	Router(config)# **access list** *access-list-number* {**deny** | **permit**} **ip** *source source-wildcard destination destination-wildcard*	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Address of the host from which the packet is being sent. • *source-wildcard*—Wildcard bits to be applied to the source address. • *destination*—Address of the host to which the packet is being sent. • *destination-wildcard*—Wildcard bits to be applied to the destination address. For details on configuring an access list, see the *Cisco IOS Security Configuration Guide*.

	Command	Purpose
Step 5	Router(config)# **crypto map** *map-name seq-number* **ipsec-isakmp** ... Router(config-crypto-map)# **exit**	Creates or modifies a crypto map entry and enters the crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**— Indicates that IKE will be used to establish the IPsec security associations. For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.
Step 6	Router(config)# **vlan** *inside-vlan-id*	Adds the VLAN ID into the VLAN database. • *inside-vlan-id*—VLAN identifier.
Step 7	Router(config)# **interface vlan** *inside-vlan-id*	Enters interface configuration mode for the specified VLAN interface. • *inside-vlan-id*—VLAN identifier.
Step 8	Router(config-if)# **description inside_interface_vlan_for_crypto_map**	(Optional) Adds a comment to help identify the interface.
Step 9	Router(config-if)# **ip address** *address mask*	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 10	Router(config-if)# **crypto map** *map-name*	Applies a previously defined crypto map set to the interface. • *map-name*—Name that identifies the crypto map set. Enter the *map-name* value you created in Step 5.
Step 11	Router(config-if)# **no shutdown**	Enables the interface as a Layer 3 crypto interface VLAN.
Step 12	Router(config-if)# **crypto engine slot** *slot/subslot*	Assigns the crypto engine to the crypto interface VLAN. • *slot/subslot*—Enter the slot and subslot where the IPsec VPN SPA is located.
Step 13	Router(config-if)# **interface gigabitethernet** *slot/subslot/port*	Enters interface configuration mode for the secure port.
Step 14	Router(config-if)# **description outside_secure_port**	(Optional) Adds a comment to help identify the interface.
Step 15	Router(config-if)# **crypto connect vlan** *inside-vlan-id*	Connects the routed port to the crypto interface VLAN and enters crypto-connect mode. • *inside-vlan-id*—VLAN identifier.
Step 16	Router(config-if)# **exit**	Exits interface configuration mode.

For routed port configuration examples, see the "Routed Port in Crypto-Connect Mode Configuration Example" section on page 21-30.

Verifying a Routed Port Configuration

To verify a route port configuration, enter the **show crypto vlan** command. In the following example, Gi 1/2 is the crypto-connected port:

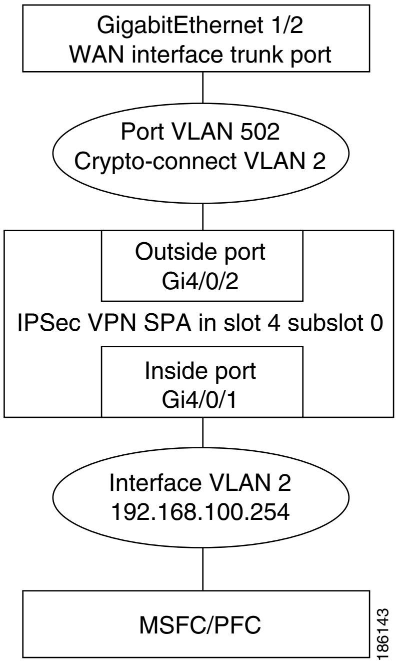
```
Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port Gi4/0/1 connected to Gi1/2 with crypto map
set MyMap
```

Configuring a Trunk Port

Caution When you configure an Ethernet port as a trunk port, all the VLANs are allowed on the trunk port by default. This default configuration does not work well with the IPsec VPN SPA and causes network loops. To avoid this problem, you must explicitly specify only the desirable VLANs.

This section describes how to configure the IPsec VPN SPA with a trunk port connection to the WAN router (see Figure 21-4).

Figure 21-4 Trunk Port Configuration Example



Note Ethernet ports installed in a Cisco 7600 SIP-400 in the chassis cannot be configured as switch ports.

Trunk Port Configuration Guidelines

When configuring a trunk port using the IPsec VPN SPA, follow these configuration guidelines:

- When you configure a trunk port for cryptographic connection, do not use the "all VLANs allowed" default. You must explicitly specify all the desirable VLANs using the **switchport trunk allowed vlan** command.
- Due to an incorrect startup configuration or through the default trunk port configuration, an interface VLAN might be associated with a trunk port. When you try to remove the interface VLAN from the VLAN list, you might receive an error message similar to the following:

```
Command rejected:VLAN 2 is crypto connected to V502.
```

To remove the interface VLAN from the VLAN list, enter the following commands:

```
Router# configure terminal
Router(config)# interface g1gabitethernet1/2
Router(config-if)# no switchport mode trunk
Router(config-if)# switchport trunk allowed vlan 1
Router(config-if)# switchport mode trunk
Router(config-if)# switchport trunk allowed vlan 1,502,1002-1005
```

Note VLANs in the VLAN list must not include any interface VLANs.

- To ensure that no interface VLANs are associated when you put an Ethernet port into the trunk mode, enter the following commands in the exact order given:

```
Router# configure terminal
Router(config)# interface g1gabitethernet1/2
Router(config)# no shut
Router(config-if)# switchport
Router(config-if)# switchport trunk allowed vlan 1
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# switchport trunk allowed vlan 1,502,1002-1005
```

Note VLANs in the VLAN list must not include any interface VLANs.

- A common mistake when configuring a trunk port occurs when you use the **add** option as follows:

```
Router(config-if)# switchport trunk allowed vlan add 502
```

If the **switchport trunk allowed vlan** command has not already been used, the **add** option does not make VLAN 502 the only allowed VLAN on the trunk port; all VLANs are still allowed after entering the command because all the VLANs are allowed by default. After you use the **switchport trunk allowed vlan** command to add a VLAN, you can then use the **switchport trunk allowed vlan add** command to add additional VLANs.

- To remove unwanted VLANs from a trunk port, use the **switchport trunk allowed vlan remove** command.

Caution Do not enter the **switchport trunk allowed vlan all** command on a secured trunk port. In addition, do not set the IPsec VPN SPA inside and outside ports to "all VLANs allowed."

To configure a trunk port connection to the WAN switch, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto isakmp policy** *priority* ... Router(config-isakmp) # **exit**	Defines an ISAKMP policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest. For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.
Step 2	Router(config)# **crypto isakmp key** *keystring* **address** *peer-address*	Configures a preshared authentication key. • *keystring*—Preshared key. • *peer-address*—IP address of the remote peer. For details on configuring a preshared key, see the *Cisco IOS Security Configuration Guide*.
Step 3	Router(config)# **crypto ipsec transform-set** *transform-set-name* *transform1*[*transform2*[*transform3*]] ... Router(config-crypto-tran)# **exit**	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. For accepted *transformx* values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference*.
Step 4	Router(config)# **access list** *access-list-number* {**deny** | **permit**} **ip** *source source-wildcard destination destination-wildcard*	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Address of the host from which the packet is being sent. • *source-wildcard*—Wildcard bits to be applied to the source address. • *destination*—Address of the host to which the packet is being sent. • *destination-wildcard*—Wildcard bits to be applied to the destination address. For details on configuring an access list, see the *Cisco IOS Security Configuration Guide*.

	Command	Purpose
Step 5	Router(config)# **crypto map** *map-name seq-number* **ipsec-isakmp** ... Router(config-crypto-map)# **exit**	Creates or modifies a crypto map entry and enters the crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations. For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.
Step 6	Router(config)# **vlan** *inside-vlan-id*	Adds the VLAN ID into the VLAN database. • *inside-vlan-id*—VLAN identifier.
Step 7	Router(config)# **vlan** *outside-vlan-id*	Adds the VLAN ID into the VLAN database. • *outside-vlan-id*—VLAN identifier.
Step 8	Router(config)# **interface vlan** *inside-vlan-id*	Enters interface configuration mode for the specified VLAN interface. • *inside-vlan-id*—VLAN identifier.
Step 9	Router(config-if)# **description inside_interface_vlan_for_crypto_map**	(Optional) Adds a comment to help identify the interface.
Step 10	Router(config-if)# **ip address** *address mask*	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 11	Router(config-if)# **crypto map** *map-name*	Applies a previously defined crypto map set to the interface. • *map-name*—Name that identifies the crypto map set. Enter the *map-name* value you created in Step 5.
Step 12	Router(config-if)# **no shutdown**	Enables the interface as a Layer 3 crypto interface VLAN.
Step 13	Router(config-if)# **crypto engine slot** *slot/subslot*	Assigns the crypto engine to the crypto interface VLAN. • *slot/subslot*—Enter the slot and subslot where the IPsec VPN SPA is located.
Step 14	Router(config)# **interface vlan** *outside-vlan-id*	Adds the specified VLAN interface as an outside trunk port VLAN and enters interface configuration mode for the specified VLAN interface. • *outside-vlan-id*—VLAN identifier.
Step 15	Router(config-if)# **description outside_trunk_port_vlan**	(Optional) Adds a comment to help identify the interface.

	Command	Purpose
Step 16	`Router(config-if)#` **`crypto connect vlan`** *`inside-vlan-id`*	Connects the outside trunk port VLAN to the inside (crypto) interface VLAN and enters crypto-connect mode. • *inside-vlan-id*—VLAN identifier.
Step 17	`Router(config-if)#` **`no shutdown`**	Enables the interface as a Layer 3 crypto interface VLAN.
Step 18	`Router(config-if)#` **`interface gigabitethernet`** *`slot/subslot/port`*	Enters interface configuration mode for the secure port.
Step 19	`Router(config-if)#` **`description outside_secure_port`**	(Optional) Adds a comment to help identify the interface.
Step 20	`Router(config-if)#` **`switchport`**	Configures the interface for Layer 2 switching.
Step 21	`Router(config-if)#` **`no switchport access vlan`**	Resets the access VLAN to the appropriate default VLAN for the device.
Step 22	`Router(config-if)#` **`switchport trunk encapsulation dot1q`**	Sets the trunk encapsulation to 802.1Q.
Step 23	`Router(config-if)#` **`switchport mode trunk`**	Specifies a trunk VLAN Layer 2 interface.
Step 24	`Router(config-if)#` **`switchport trunk allowed vlan remove`** *`vlan-list`*	Removes the specified list of VLANs from those currently set to transmit from this interface. *vlan-list*—List of VLANs that transmit the interface in tagged format when in trunking mode. Valid values are from 1 to 4094.
Step 25	`Router(config-if)#` **`switchport trunk allowed vlan add`** *`outside-vlan-id`*	Adds the specified VLAN to the list of VLANs currently set to transmit from this interface. *outside-vlan-id*—VLAN identifier from Step 14.
Step 26	`Router(config-if)#` **`exit`**	Exits interface configuration mode.

For trunk port configuration examples, see the "Trunk Port in Crypto-Connect Mode Configuration Example" section on page 21-33.

Verifying the Trunk Port Configuration

To verify the VLANs allowed by a trunk port, enter the **show interfaces trunk** command. The following display shows that all VLANs are allowed:

```
Router# show interfaces GigabitEthernet 1/2 trunk

Port      Mode         Encapsulation  Status        Native vlan
Gi1/2     on           802.1q         trunking      1

Port      Vlans allowed on trunk
Gi1/2     1-4094

Port      Vlans allowed and active in management domain
Gi1/2     1-4,7-8,513,1002-1005

Port      Vlans in spanning tree forwarding state and not pruned
Gi1/2     1-4,7-8,513,1002-1005
```

Configuring IPsec VPN SPA Connections to WAN Interfaces

The configuration of IPsec VPN SPA connections to WAN interfaces is similar to the configuration of Ethernet-routed interfaces.

IPsec VPN SPA Connections to WAN Interfaces Configuration Guidelines and Restrictions

When configuring a connection to a WAN interface using an IPsec VPN SPA, follow these guidelines and note these restrictions:

- To configure an IPsec VPN SPA connection to a WAN interface, make a crypto connection from the WAN subinterface to the interface VLAN as follows:

```
Router(config)# interface Vlan101
Router(config-if)# ip address 192.168.101.1 255.255.255.0
Router(config-if)# no mop enabled
Router(config-if)# crypto map cwan
Router(config-if)# crypto engine slot 4/0

Router(config)# interface ATM6/0/0.101 point-to-point
Router(config-subif)# pvc 0/101
Router(config-subif)# crypto connect vlan 101
```

- You must configure a crypto connection on subinterfaces for ATM and Frame Relay.
- For ATM, there is no SVC support, no RFC-1483 bridging, and no point-to-multipoint support.
- For Frame Relay, there is no SVC support, no RFC-1490 bridging, and no point-to-multipoint support.
- For Point-to-Point Protocol (PPP) and Multilink PPP (MLPPP), you must make the physical interface passive for routing protocols, as follows:

```
Router(config)# router ospf 10
Router(config-router)# passive-interface multilink1
```

- For PPP and MLPPP, when the **crypto connect vlan** command is configured on an interface, an **ip unnumbered Null0** command is automatically added to the port configuration to support IPCP negotiation. If you configure a **no ip address** command on the WAN port in the startup configuration, the **no ip address** command will be automatically removed in the running configuration so that it does not conflict with the automatic configuration.
- For PPP and MLPPP, there is no Bridging Control Protocol (BCP) support.
- When enabled on an inside VLAN, OSPF will be configured in broadcast network mode by default, even when a point-to-point interface (such as T1, POS, serial, or ATM) is crypto-connected to the inside VLAN. In addition, if OSPF is configured in point-to-point network mode on the peer router (for example, a transit router with no crypto card), OSPF will not establish full adjacency. In this case, you can manually configure OSPF network point-to-point mode in the inside VLAN:

```
Router(config)# interface vlan inside-vlan
Router(config-if)# ip ospf network point-to-point
```

For IPsec VPN SPA connections to WAN interfaces configuration examples, see the "IPsec VPN SPA Connections to WAN Interfaces Configuration Examples" section on page 21-35

Displaying the VPN Running State

Use the **show crypto vlan** command to display the VPN running state. The following examples show the **show crypto vlan** command output for a variety of IPsec VPN SPA configurations.

In the following example, the interface VLAN belongs to the IPsec VPN SPA inside port:

```
Router# show crypto vlan

  Interface VLAN 2 on IPSec Service Module port Gi4/0/1 connected to Fa8/3
```

In the following example, VLAN 2 is the interface VLAN and VLAN 2022 is the hidden VLAN:

```
Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port Gi4/0/1 connected to VLAN 2022 with crypto
map set coral2
```

In the following example, the interface VLAN is missing on the IPsec VPN SPA inside port, the IPsec VPN SPA is removed from the chassis, or the IPsec VPN SPA was moved to a different subslot:

```
Router# show crypto vlan

  Interface VLAN 2 connected to VLAN 502 (no IPSec Service Module attached)
```

Configuring GRE Tunneling in Crypto-Connect Mode

This section contains the following GRE configuration topics:

- Understanding GRE Tunneling in Crypto-Connect Mode, page 21-20
- Configuring the GRE Takeover Criteria, page 21-22
- Configuring IP Multicast over a GRE Tunnel, page 21-25

Understanding GRE Tunneling in Crypto-Connect Mode

Generic Routing Encapsulation (GRE) is a tunneling protocol that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to switches at remote points over an IP network.

Note The IPsec VPN SPA is able to accelerate packet processing for up to 2048 GRE tunnels per chassis. Any tunnels not taken over by the IPsec VPN SPA, or any tunnels in excess of 2048, are handled in platform hardware or by the route processor. The switch supports any number of GRE tunnels, but adding more IPsec VPN SPAs does not increase the 2048 tunnels per-chassis maximum that will be handled by IPsec VPN SPAs. If you configure more than 2048 tunnels per chassis, you could overload the route processor. Monitor the route processor CPU utilization when configuring more than 2048 tunnels per chassis.

Note Beginning with Cisco IOS Release 12.2(18)SXF, the GRE fragmentation behavior of the VPN module is changed to be consistent with the fragmentation behavior of the route processor. If GRE encapsulation is performed by the VPN module, prefragmentation of outbound packets will be based on the IP MTU of the tunnel interface. After GRE encapsulation is performed by the VPN module, depending on the IPsec prefragmentation settings, further fragmentation may occur. The IPsec fragmentation behavior is unchanged in this release, and is based on the IPsec MTU configuration of the egress interface.

GRE Tunneling Configuration Guidelines and Restrictions

When configuring point-to-point GRE tunneling in crypto-connect mode using the IPsec VPN SPA, follow these guidelines:

- In a Catalyst 6500 Series switch, GRE encapsulation and decapsulation is traditionally performed by the route processor or the supervisor engine hardware. When routing indicates that encapsulated packets for a GRE tunnel will egress through an interface VLAN that is attached to an IPsec VPN SPA inside port, the IPsec VPN SPA attempts to take over the GRE tunnel interface only if the supervisor engine is unable to process the GRE tunnel interface in hardware. If the supervisor engine cannot process the GRE tunnel interface in hardware, the IPsec VPN SPA will determine if it can take over the interface. By seizing the tunnel, the IPsec VPN SPA takes the GRE encapsulation and decapsulation duty from the route processor. No explicit configuration changes are required to use this feature; configure GRE as you normally would. As long as routing sends the GRE-encapsulated packets over an interface VLAN, the IPsec VPN SPA will seize the GRE tunnel.
- If the same source address is used for more than one GRE tunnel, the supervisor engine hardware will not take over the tunnel. The IPsec VPN SPA will take over the tunnel if it meets the criteria discussed in the previous bullet item.
- Point-to-point GRE with tunnel protection is not supported in crypto-connect mode, but DMVPN is supported.
- If routing information changes and the GRE-encapsulated packets no longer egress through an interface VLAN, the IPsec VPN SPA yields the GRE tunnel. After the IPsec VPN SPA yields the tunnel, the route processor resumes encapsulation and decapsulation, which increases CPU utilization on the route processor.

⚠ **Caution** Ensure that your GRE tunnel configuration does not overload the route processor.

- A delay of up to 10 seconds occurs between routing changes and the IPsec VPN SPA seizing the GRE tunnel.
- The crypto map must only be applied to the interface VLAN and not to the tunnel interface.
- The following options are supported on the tunnel interface: ACLs, service policy, TTL, and ToS.
- The following options are not supported on the tunnel interface: checksum enabled, sequence check enabled, tunnel key, IP security options, policy-based routing (PBR), traffic shaping (can be applied to the crypto engine configuration within the tunnel interface configuration), QoS preclassification, and NAT.
- GRE tunneling of all non-IPv4 packets is done by the route processor even if the tunnel is seized by the IPsec VPN SPA.
- In crypto-connect mode, to avoid fragmentation after encryption, set the tunnel IP MTU to be equal to or less than the egress interface MTU minus the GRE and IPsec overheads.
- When applied to the GRE tunnel interface, the **ip tcp adjust-mss** command is ignored. Apply the command to the ingress LAN interface instead.

To configure a GRE tunnel, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface tunnel`** *`number`*	Creates the tunnel interface if it does not exist and enters interface configuration mode. • *number*—Number of the tunnel interface to be configured.
Step 2	`Router(config-if)#` **`ip address`** *`address`*	Sets the IP address of the tunnel interface. • *address*—IP address.
Step 3	`Router(config-if)#` **`tunnel source`** `{`*`ip-address`* `|` *`type number`*`}`	Configures the tunnel source. The source is the switch where traffic is received from the customer network. • *ip-address*—IP address to use as the source address for packets in the tunnel. • *type number*—Interface type and number; for example, VLAN1.
Step 4	`Router(config-if)#` **`tunnel destination`** `{`*`hostname`* `|` *`ip-address`*`}`	Sets the IP address of the destination of the tunnel interface. The destination address is the switch that transfers packets into the receiving customer network. • *hostname*—Name of the host destination. • *ip-address*—IP address of the host destination expressed in decimal in four-part, dotted notation.
Step 5	`Router(config-if)#` **`exit`**	Exits interface configuration mode.

Verifying the GRE Tunneling Configuration

To verify that the IPsec VPN SPA has seized the GRE tunnel, enter the **show crypto vlan** command:

```
Router# show crypto vlan

Interface VLAN 101 on IPSec Service Module port 7/1/1 connected to AT4/0/0.101
    Tunnel101 is accelerated via IPSec SM in subslot 7/1
Router#
```

For complete configuration information about GRE tunneling, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12s_tos.html

For GRE tunneling configuration examples, see the "GRE Tunneling in Crypto-Connect Mode Configuration Example" section on page 21-39.

Configuring the GRE Takeover Criteria

You can configure the takeover criteria for Generic Routing Encapsulation (GRE) processing by using the **crypto engine gre supervisor** or **crypto engine gre vpnblade** commands. These two commands allow you to specify whether the GRE processing should be done by the supervisor engine hardware or the route processor or the IPsec VPN SPA.

Note The GRE takeover criteria commands are supported only in Cisco IOS Release 12.2(18)SXE5 and later. In releases prior to Cisco IOS Release 12.2SXE1, the crypto-related GRE tunnels are always taken over by the VPN SPA. In Cisco IOS Release 12.2SXE1, the GRE tunnels are taken over by the VPN SPA only if the supervisor engine hardware cannot do the processing.

To configure a switch to process GRE using the supervisor engine hardware or the route processor (RP), use the **crypto engine gre supervisor** command. When this command is specified, GRE processing by the supervisor engine hardware takes precedence over processing by the route processor (unless the tunnels are from duplicate sources); the RP only takes over GRE processing if the supervisor engine hardware cannot do the processing. If this command is configured, duplicate source GREs will be processed by the route processor.

To configure a switch to process GRE using the IPsec VPN SPA, use the **crypto engine gre vpnblade** command. If the IPsec VPN SPA cannot take over the GRE processing, the GRE processing will be handled either by supervisor engine hardware (which has precedence) or the route processor.

Both of these commands can be configured globally or at an individual tunnel.

Individual tunnel configuration takes precedence over the global configuration. For example, when the **crypto engine gre supervisor** command is configured at the global configuration level, the command will apply to all tunnels except those tunnels that have been configured individually using either a **crypto engine gre supervisor** command or a **crypto engine gre vpnblade** command.

At any time, only one of the two commands (**crypto engine gre supervisor** or **crypto engine gre vpnblade**) can be configured globally or individually at a tunnel. If either command is already configured, configuring the second command will overwrite the first command, and only the configuration applied by the second command will be used.

GRE Takeover Configuration Guidelines and Restrictions

When configuring GRE takeover on the IPsec VPN SPA, follow these guidelines and restrictions:

- For a GRE tunnel to be taken over by the IPsec VPN SPA, it must first satisfy the following criteria:
  - The GRE tunnel interface must be up.
  - The route to the tunnel destination must go through the IPsec VPN SPA.
  - The Address Resolution Protocol (ARP) entry for the next hop must exist.
  - The tunnel mode must be GRE.
  - The only supported options are **tunnel ttl** and **tunnel tos**. If any of the following command options are configured, then the tunnel will not be taken over:
    - **tunnel key**
    - **tunnel sequence-datagrams**
    - **tunnel checksum**

    All other options configured are ignored.
- If the GRE tunnels have the same source and destination addresses, then the IPsec VPN SPA will, at most, take over only one of them, and the determination of which specific tunnel is taken over is random.
- The IPsec VPN SPA will not take over GRE processing if any of the following features are configured on the tunnel interface:
  - DMVPN

  - NAT
- In crypto-connect mode, the IPsec VPN SPA will not take over GRE processing when the interface VLAN has no crypto map attached. The crypto map must be applied to the interface VLAN and not to the tunnel interface.
- If the IPsec VPN SPA cannot take over the GRE processing, the GRE processing will be handled either by the supervisor engine hardware (which has precedence) or the route processor.
- When neither the **crypto engine gre supervisor** command nor the **crypto engine gre vpnblade** command is specified globally or individually for a tunnel, the IPsec VPN SPA will only attempt to take over GRE processing if the following conditions apply:
  - The supervisor engine hardware does not take over GRE processing.
  - Protocol Independent Multicast (PIM) is configured on the tunnel.
  - Multiple tunnels share the same tunnel source interface and more than one tunnel is up. (If only one tunnel is up, the supervisor engine hardware can still perform the GRE processing.)
- When a new configuration file is copied to the running configuration, the new configuration will overwrite the old configuration for the **crypto engine gre vpnblade** and **crypto engine gre supervisor** commands. If the new configuration does not specify a GRE takeover criteria globally or for an individual tunnel, the existing old configuration will be used.

Configuring the GRE Takeover Criteria Globally

To configure the GRE takeover criteria globally (so that it affects all tunnels except those tunnels that have been configured individually using either a **crypto engine gre supervisor** command or a **crypto engine gre vpnblade** command), perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto engine gre supervisor`	Configures a router to process GRE using the supervisor engine hardware or the route processor.
	or	
	`Router(config)# crypto engine gre vpnblade`	Configures a router to process GRE using the IPsec VPN SPA.

Configuring the GRE Takeover Criteria at an Individual Tunnel

To configure the GRE takeover criteria at an individual tunnel (so that it affects only a specific tunnel), perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`interface tunnel`** *`number`*	Creates the tunnel interface if it does not exist and enters interface configuration mode. • *number*—Number of the tunnel interface to be configured.

Step 2	`Router(config-if)# `**`crypto engine gre supervisor`**
	`or`
	`Router(config-if)# `**`crypto engine gre vpnblade`**
		Configures a router to process GRE using the supervisor engine hardware or the route processor.
		or
		Configures a router to process GRE using the IPsec VPN SPA.

For GRE takeover criteria configuration examples, see the "GRE Takeover Criteria Configuration Examples" section on page 21-42.

Configuring IP Multicast over a GRE Tunnel

IP multicast is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to multiple recipients. GRE is a tunneling protocol developed by Cisco and commonly used with IPsec that encapsulates a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP network.

In some network scenarios, you might want to configure your network to use GRE tunnels to send Protocol Independent Multicast (PIM) and multicast traffic between routers. Typically, this occurs when the multicast source and receiver are separated by an IP cloud that is not configured for IP multicast routing. In such network scenarios, configuring a tunnel across an IP cloud with PIM-enabled transports multicast packets toward the receiver. The configuration of IP multicast over a GRE tunnel using the IPsec VPN SPA involves three key steps:

- Configuring single-SPA mode (if supported) for multicast traffic
- Configuring multicast globally
- Configuring PIM at the tunnel interfaces

IP Multicast over a GRE Tunnel Configuration Guidelines and Restrictions

When configuring IP multicast over a GRE tunnel, follow these guidelines:

- When the **hw-module slot subslot only** command is executed, it automatically resets the Cisco 7600 SSC-400 card and displays the following prompt on the console:

  ```
  Module n will be reset? Confirm [n]:
  ```

  The prompt will default to N (no). You must type Y (yes) to activate the reset action.

Note The **hw-module slot subslot only** command is not supported in Cisco IOS Release 12.2(33)SXI and later releases.

- When in single-SPA mode, if you manually plug in a second SPA, or if you attempt to reset the SPA (by entering a **no hw-module subslot shutdown** command, for example), a message is displayed on the router console that refers you to the customer documentation.
- If PIM is configured, and the GRE tunnel interface satisfies the rest of the tunnel takeover criteria, the GRE processing of the multicast packets will be taken over by the IPsec VPN SPA.
- GRE processing of IP multicast packets will be taken over by the IPsec VPN SPA if the GRE tunnel interface satisfies the following tunnel takeover criteria:
  - The tunnel is up.
  - There are no other tunnels with the same source destination pair.
  - The tunnel is not an mGRE tunnel.
  - PIM is configured on the tunnel.
  - None of the following features are configured on the tunnel: tunnel key, tunnel sequence-datagrams, tunnel checksum, tunnel udlr address-resolution, tunnel udlr receive-only, tunnel udlr send-only, ip proxy-mobile tunnel reverse, or NAT. If any of these options are specified, the IPsec VPN SPA will not seize the GRE tunnel.
- When a tunnel is configured for multicast traffic, the **crypto engine gre supervisor** command should not be applied to the tunnel.

Configuring Single-SPA Mode for IP Multicast Traffic

Note Single-SPA mode is not supported in Cisco IOS Release 12.2(33)SXI and later releases.

Before you configure IP multicast on the IPsec VPN SPA, you should change the mode of the Cisco 7600 SSC-400 card to allocate full buffers to the specified subslot using the Before you configure IP multicast on the IPsec VPN SPA, you should change the mode of the Cisco 7600 SSC-400 card to allocate full buffers to the specified subslot using the **hw-module slot subslot only** command. If this command is not used, the total amount of buffers available is divided between the two subslots on the Cisco 7600 SSC-400 card.

To allocate full buffers to the specified subslot, use the **hw-module slot subslot only** command as follows:

```
Router(config)# hw-module slot slot subslot subslot only
```

slot specifies the slot where the Cisco 7600 SSC-400 card is located.

subslot specifies the subslot where the IPsec VPN SPA is located.

If the **hw-module slot subslot only** command is not used, the total amount of buffers available is divided between the two subslots on the Cisco 7600 SSC-400 card.

Configuring IP Multicast Globally

You must enable IP multicast routing globally before you can enable PIM on the router interfaces.

To enable IP multicast routing globally, use the **ip multicast-routing** command.

Configuring PIM at the Tunnel Interfaces

You must enable PIM on all participating router interfaces before IP multicast will function.

To enable PIM, use the **ip pim** command as follows:

```
Router(config-if)# ip pim {dense-mode | sparse-mode | sparse-dense-mode}
```

dense-mode enables dense mode of operation.

sparse-mode enables sparse mode of operation.

sparse-dense-mode enables the interface in either sparse mode or dense mode of operation, depending on which mode the multicast group operates in.

For IP multicast over GRE tunnels configuration examples, see the "IP Multicast over a GRE Tunnel Configuration Example" section on page 21-42.

Verifying the IP Multicast over a GRE Tunnel Configuration

To verify the IP multicast over a GRE tunnel configuration, enter the **show crypto vlan** and **show ip mroute** commands.

To verify that the tunnel has been taken over by the IPsec VPN SPA, enter the **show crypto vlan** command:

```
Router# show crypto vlan

Interface VLAN 100 on IPSec Service Module port Gi7/0/1 connected to Po1 with crypto map
set map_t3
Tunnel15 is accelerated via IPSec SM in subslot 7/0
```

To verify that the IP multicast traffic is hardware-switched, enter the **show ip mroute** command and look for the **H** flag:

```
Router# show ip mroute 230.1.1.5

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 230.1.1.5), 01:23:45/00:03:16, RP 15.15.1.1, flags: SJC
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Tunnel15, Forward/Sparse-Dense, 00:25:47/00:03:16
(120.1.0.3, 230.1.1.5), 01:23:46/00:03:25, flags: T
Incoming interface: GigabitEthernet8/1, RPF nbr 0.0.0.0, RPF-MFD
Outgoing interface list:
Tunnel15, Forward/Sparse-Dense, 00:25:47/00:03:16, H
```

For IP multicast over GRE tunnels configuration examples, see the "IP Multicast over a GRE Tunnel Configuration Example" section on page 21-42.

Configuration Examples

This section provides examples of the following configurations:

- Access Port in Crypto-Connect Mode Configuration Example, page 21-28
- Routed Port in Crypto-Connect Mode Configuration Example, page 21-30
- Trunk Port in Crypto-Connect Mode Configuration Example, page 21-33
- IPsec VPN SPA Connections to WAN Interfaces Configuration Examples, page 21-35
- GRE Tunneling in Crypto-Connect Mode Configuration Example, page 21-39
- GRE Takeover Criteria Configuration Examples, page 21-42
- IP Multicast over a GRE Tunnel Configuration Example, page 21-42

Note The following examples use commands at the level of Cisco IOS Release 12.2(33)SXH.

As of Cisco IOS Release 12.2(33)SXH, the **crypto engine subslot** command used in previous releases has been replaced with the **crypto engine slot** command (of the form **crypto engine slot** *slot/subslot* {**inside** | **outside**}). The **crypto engine subslot** command is no longer supported. When upgrading, ensure that this command has been modified in your start-up configuration to avoid extended maintenance time.

Access Port in Crypto-Connect Mode Configuration Example

This section provides an example of the access port configuration with switch 1 shown in Figure 21-2 on page 21-8:

Switch 1 (Access Port)

```
!
hostname router-1
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.1
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
  !switch inside port
  ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 502
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk

interface Vlan2
 !interface vlan
 ip address 11.0.0.2 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan502
 !port vlan
 no ip address
 crypto connect vlan 2
!
ip classless
ip route 12.0.0.0 255.0.0.0 11.0.0.1
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
!
end
```

Switch 2 (Access Port)

```
!
hostname router-2
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.2
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal1
 match address 101
!
```

```
!
interface GigabitEthernet1/1
  !switch inside port
  ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 502
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 !interface vlan
 ip address 11.0.0.1 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan502
 !port vlan
 no ip address
 crypto connect vlan 2
!
ip classless
ip route 13.0.0.0 255.0.0.0 11.0.0.2
!
access-list 101 permit ip host 12.0.0.2 host 13.0.0.2
!
end
```

Routed Port in Crypto-Connect Mode Configuration Example

This section provides an example of the routed port configuration with switch 1 shown in Figure 21-3 on page 21-11:

Switch 1 (Routed Port)

```
!
hostname router-1
```

```
!
vlan 2
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 crypto isakmp key 12345 address 11.0.0.2
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 no ip address
 crypto connect vlan 2
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 !interface vlan
 ip address 11.0.0.1 255.255.255.0
 no mop enabled
 crypto map testtag
 crypto engine slot 4/0
!
ip classless
ip route 13.0.0.0 255.0.0.0 11.0.0.2
!
access-list 101 permit ip host 12.0.0.2 host 13.0.0.2
!
end
```

Switch 2 (Routed Port)

```
!
hostname router-2
!
vlan 2
!
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.1
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 no ip address
 crypto connect vlan 2
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 !interface vlan
 ip address 11.0.0.2 255.255.255.0
 no mop enabled
 crypto map testtag
 crypto engine slot 4/0
!
ip classless
ip route 12.0.0.0 255.0.0.0 11.0.0.1
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
```

```
!
end
```

Trunk Port in Crypto-Connect Mode Configuration Example

This section provides an example of the trunk port configuration with switch 1 shown in Figure 21-4 on page 21-14:

Switch 1 (Trunk Port)

```
!
hostname router-1
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.2
!
!
crypto ipsec transform-set proposal1  esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
  !switch inside port
  ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 502
 switchport mode trunk
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 !interface vlan
 ip address 11.0.0.1 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan 502
 !port vlan
 no ip address
 crypto connect vlan 2
!
ip classless
ip route 13.0.0.0 255.0.0.0 11.0.0.2
!
access-list 101 permit ip host 12.0.0.2 host 13.0.0.2
!
end
```

Switch 2 (Trunk Port)

```
!
hostname router-2
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.1
!
!
crypto ipsec transform-set proposal1  esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
  !switch inside port
  ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 502
 switchport mode trunk
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
```

```
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk

interface Vlan2
 !interface vlan
 ip address 11.0.0.2 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan502
 !port vlan
 no ip address
 crypto connect vlan 2
!
ip classless
ip route 12.0.0.0 255.0.0.0 11.0.0.1
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
!
end
```

IPsec VPN SPA Connections to WAN Interfaces Configuration Examples

The following are configuration examples of IPsec VPN SPA connections to WAN interfaces:

- IPsec VPN SPA Connection to an ATM Port Adapter Configuration Example, page 21-35
- IPsec VPN SPA Connection to a POS Port Adapter Configuration Example, page 21-36
- IPsec VPN SPA Connection to a Serial Port Adapter Configuration Example, page 21-37

IPsec VPN SPA Connection to an ATM Port Adapter Configuration Example

The following example shows the configuration of an IPsec VPN SPA connection to an ATM port adapter:

```
!
hostname router-1
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key 12345 address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set proposal esp-3des esp-sha-hmac
!
crypto map testtag_1 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal
 match address acl_1
```

```
!
interface GigabitEthernet1/1
 ip address 12.0.0.2 255.255.255.0
!
interface ATM2/0/0
 no ip address
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM2/0/0.1 point-to-point
 atm pvc 20 0 20 aal5snap
 no atm enable-ilmi-trap
 crypto connect vlan 2
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 11.0.0.1 255.255.255.0
 crypto map testtag_1
 crypto engine slot 4/0
!
ip classless
ip route 13.0.0.1 255.255.255.255 11.0.0.2
!
ip access-list extended acl_1
 permit ip host 12.0.0.1 host 13.0.0.1
!
```

IPsec VPN SPA Connection to a POS Port Adapter Configuration Example

The following example shows the configuration of an IPsec VPN SPA connection to a POS port adapter:

```
!

hostname router-1
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key 12345 address 0.0.0.0 0.0.0.0
```

```
!
!
crypto ipsec transform-set proposal esp-3des esp-sha-hmac
!
crypto map testtag_1 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal
 match address acl_1
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 12.0.0.2 255.255.255.0
!
interface POS2/0/0
 no ip address
 encapsulation frame-relay
 clock source internal
!
interface POS2/0/0.1 point-to-point
 frame-relay interface-dlci 16
 crypto connect vlan 2
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 11.0.0.1 255.255.255.0
 crypto map testtag_1
 crypto engine slot 4/0
!
ip classless
ip route 13.0.0.1 255.255.255.255 11.0.0.2
!
ip access-list extended acl_1
 permit ip host 12.0.0.1 host 13.0.0.1
```

IPsec VPN SPA Connection to a Serial Port Adapter Configuration Example

The following example shows the configuration of an IPsec VPN SPA connection to a serial port adapter:

```
!
hostname router-1
!
```

```
controller T3 2/1/0
 t1 1 channel-group 0 timeslots 1
 t1 2 channel-group 0 timeslots 1
 t1 3 channel-group 0 timeslots 1
 t1 4 channel-group 0 timeslots 1
 t1 5 channel-group 0 timeslots 1
 t1 6 channel-group 0 timeslots 1
 t1 7 channel-group 0 timeslots 1
 t1 8 channel-group 0 timeslots 1
 t1 9 channel-group 0 timeslots 1
 t1 10 channel-group 0 timeslots 1
 t1 11 channel-group 0 timeslots 1
 t1 12 channel-group 0 timeslots 1
 t1 13 channel-group 0 timeslots 1
 t1 14 channel-group 0 timeslots 1
 t1 15 channel-group 0 timeslots 1
 t1 16 channel-group 0 timeslots 1
 t1 17 channel-group 0 timeslots 1
 t1 18 channel-group 0 timeslots 1
 t1 19 channel-group 0 timeslots 1
 t1 20 channel-group 0 timeslots 1
 t1 21 channel-group 0 timeslots 1
 t1 22 channel-group 0 timeslots 1
 t1 23 channel-group 0 timeslots 1
 t1 24 channel-group 0 timeslots 1
 t1 25 channel-group 0 timeslots 1
 t1 26 channel-group 0 timeslots 1
 t1 27 channel-group 0 timeslots 1
 t1 28 channel-group 0 timeslots 1
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key 12345 address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set proposal esp-3des esp-sha-hmac
!
crypto map testtag_1 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal
 match address acl_1
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 12.0.0.2 255.255.255.0
!
interface Serial2/1/0/1:0
 ip unnumbered Null0
 encapsulation ppp
 no fair-queue
 no cdp enable
 crypto connect vlan 2
!
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 11.0.0.1 255.255.255.0
 crypto map testtag_1
 crypto engine slot 4/0
!
ip classless
ip route 13.0.0.1 255.255.255.255 11.0.0.2
!
ip access-list extended acl_1
 permit ip host 12.0.0.1 host 13.0.0.1
```

GRE Tunneling in Crypto-Connect Mode Configuration Example

This section provides an example of GRE tunneling configurations.

Switch 1 (GRE Tunneling)

The following example shows the configuration of GRE tunneling for switch 1:

```
!
hostname router-1
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.2
!
!
crypto ipsec transform-set proposal1 ah-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal1
 match address 101
!
!
!
!
interface Tunnel1
 ip address 1.0.0.1 255.255.255.0
 tunnel source Vlan2
 tunnel destination 11.0.0.2
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 502
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 11.0.0.1 255.255.255.0
 no mop enabled
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan502
 no ip address
 crypto connect vlan 2
!
!
ip classless
ip route 13.0.0.0 255.0.0.0 Tunnel1
!
!
access-list 101 permit gre host 11.0.0.1 host 11.0.0.2
!
```

Switch 2 (GRE Tunneling)

```
!
hostname router-2
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.1
!
!
crypto ipsec transform-set proposal1 ah-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
```

```
 set peer 11.0.0.1
 set transform-set proposal1
 match address 101
!
!
!
!
interface Tunnel1
 ip address 1.0.0.2 255.255.255.0
 tunnel source Vlan2
 tunnel destination 11.0.0.1
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 502
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 11.0.0.2 255.255.255.0
 no mop enabled
 crypto map testtag
 crypto engine slot 4/0
!
interface Vlan502
 no ip address
 crypto connect vlan 2
!
ip classless
ip route 12.0.0.0 255.0.0.0 Tunnel1
!
access-list 101 permit gre host 11.0.0.2 host 11.0.0.1
!
```

GRE Takeover Criteria Configuration Examples

The following examples show how to configure the GRE takeover criteria:

- GRE Takeover Criteria Global Configuration Example, page 21-42
- GRE Takeover Criteria Tunnel Configuration Example, page 21-42
- GRE Takeover Verification Example, page 21-42

GRE Takeover Criteria Global Configuration Example

The following example shows that the GRE takeover criteria has been set globally and the supervisor engine hardware or RP always does the GRE processing:

```
Router(config)# crypto engine gre supervisor
```

GRE Takeover Criteria Tunnel Configuration Example

The following example shows that the GRE takeover criteria has been set individually for tunnel interface 3 and the IPsec VPN SPA always does the GRE processing for this tunnel:

```
Router(config)# interface tunnel 3
Router(config-if)# crypto engine gre vpnblade
```

GRE Takeover Verification Example

The following example shows how to verify that the tunnel has been taken over by the IPsec VPN SPA:

```
Router(config)# show crypto vlan 100

Interface VLAN 100 on IPSec Service Module port GigabitEthernet4/0/1 connected to POS8/0/0
with crypto map set MAP_TO_R2
    Tunnel1 is accelerated via IPSec SM in subslot 4/0
```

The following example shows that the tunnel has not been taken over by the IPsec VPN SPA:

```
Router(config)# show crypto vlan 100

Interface VLAN 100 on IPSec Service Module port GigabitEthernet4/0/1 connected to POS8/0/0
with crypto map set MAP_TO_R2
```

IP Multicast over a GRE Tunnel Configuration Example

The following example shows how to configure IP multicast over GRE:

```
hostname router-1
!
vlan 2-1001
ip multicast-routing
!
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key 12345 address 0.0.0.0 0.0.0.0
```

```
!
!
crypto ipsec transform-set proposal esp-3des
!
!
crypto map cm_spoke1_1 10 ipsec-isakmp
 set peer 11.1.1.1
 set transform-set proposal
 match address spoke1_acl_1
!
!
interface Tunnel1
 ip address 20.1.1.1 255.255.255.0
 ip mtu 9216
 ip pim sparse-mode
 ip hold-time eigrp 1 3600
 tunnel source 1.0.1.1
 tunnel destination 11.1.1.1
 crypto engine slot 4/0
!
interface GigabitEthernet1/1
 !switch inside port
 mtu 9216
 ip address 50.1.1.1 255.0.0.0
 ip pim sparse-mode
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,252,1002-1005
 switchport mode trunk
 mtu 9216
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,252,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 mtu 9216
 ip address 1.0.1.1 255.255.255.0
 crypto map cm_spoke1_1
 crypto engine slot 4/0
!
interface Vlan252
 mtu 9216
 no ip address
 crypto connect vlan 2
!
router eigrp 1
 network 20.1.1.0 0.0.0.255
 network 50.1.1.0 0.0.0.255
 no auto-summary
 no eigrp log-neighbor-changes
!
ip classless
ip route 11.1.1.0 255.255.255.0 1.0.1.2
!
ip pim bidir-enable
ip pim rp-address 50.1.1.1
!
ip access-list extended spoke1_acl_1
 permit gre host 1.0.1.1 host 11.1.1.1
!
```

CHAPTER 22

Configuring VPNs in VRF Mode

This chapter provides information about configuring IPsec VPNs in Virtual Routing and Forwarding (VRF) mode, one of the two VPN configuration modes supported by the IPsec VPN SPA. For information on the other VPN mode, crypto-connect mode, see Chapter 21, "Configuring VPNs in Crypto-Connect Mode."

This chapter includes the following topics:

For general information on configuring IPsec VPNs with the IPsec VPN SPA, see the "Overview of Basic IPsec and IKE Configuration Concepts" section on page 20-3.

Note The procedures in this chapter assume you have familiarity with security configuration concepts, such as VLANs, ISAKMP policies, preshared keys, transform sets, access control lists, and crypto maps. For detailed information on configuring these features, refer to the following Cisco IOS documentation:

Cisco IOS Security Configuration Guide, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

Cisco IOS Security Command Reference, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

For additional information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* and the related Cisco IOS Release 12.2 software configuration guide and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Tip To ensure a successful configuration of your VPN using the IPsec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Configuring VPNs in VRF Mode

VRF mode, also known as VRF-Aware IPsec, allows you to map IPsec tunnels to VPN routing and forwarding instances (VRFs) using a single public-facing address.

A VRF instance is a per-VPN routing information repository that defines the VPN membership of a customer site attached to the Provider Edge (PE) router. A VRF comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table. A separate set of routing and CEF tables is maintained for each VPN customer.

Each IPsec tunnel is associated with two VRF domains. The outer encapsulated packet belongs to one VRF domain, called the front door VRF (FVRF), while the inner, protected IP packet belongs to another domain called the inside VRF (IVRF). Stated another way, the local endpoint of the IPsec tunnel belongs to the FVRF while the source and destination addresses of the inside packet belong to the IVRF, the unprotected (LAN) side.

Note Front door VRF (FVRF) is only supported as of Cisco IOS Release 12.2(33)SXH and later.

One or more IPsec tunnels can terminate on a single interface. The FVRF of all these tunnels is the same and is set to the VRF that is configured on that interface. The IVRF of these tunnels can be different and depends on the VRF that is defined in the ISAKMP profile that is attached to a crypto map entry.

With VRF mode, packets belonging to a specific VRF are routed through the IPsec VPN SPA for IPsec processing. Through the CLI, you associate a VRF with an interface VLAN that has been configured to point to the IPsec VPN SPA. An interface VLAN must be created for each VRF. Packets traveling from an MPLS cloud to the Internet that are received from an inside VRF are routed to an interface VLAN, and then to the IPsec VPN SPA for IPsec processing. The IPsec VPN SPA modifies the packets so that they are placed on a special Layer 3 VLAN for routing to the WAN-side port after they leave the IPsec VPN SPA.

Packets traveling in the inbound direction from a protected port on which the **crypto engine slot** command has been entered are redirected by a special ACL to the IPsec VPN SPA, where they are processed according to the Security Parameter Index (SPI) contained in the packet's IPsec header. Processing on the IPsec VPN SPA ensures that the decapsulated packet is mapped to the appropriate interface VLAN corresponding to the inside VRF. This interface VLAN has been associated with a specific VRF, so packets are routed within the VRF to the correct inside interface.

Note Tunnel protection is supported in VRF mode. For information on configuring tunnel protection, see the "Configuring VPNs in VRF Mode with Tunnel Protection (GRE)" section on page 22-12 and the "VRF Mode Tunnel Protection Configuration Example" section on page 22-33.

When configuring a VPN using VRF mode, you have these additional tunneling options: tunnel protection (TP) using GRE, and Virtual Tunnel Interface (VTI). With either of these options, you can terminate tunnels in VRFs (normal VRF mode) or in the global context.

The following subsections describe how to configure a VPN in VRF mode on the IPsec VPN SPA:

- Understanding VPN Configuration in VRF Mode, page 22-3
- VRF Mode Configuration Guidelines and Restrictions, page 22-4
- Configuring VPNs in VRF Mode without Tunnel Protection, page 22-6
- Configuring VPNs in VRF Mode with Tunnel Protection (GRE), page 22-12

Note For additional information on configuring VPNs in VRF mode, refer to the Cisco IOS documentation at this URL:
http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_vrf_aware_ipsec_ps6350_TSD_Products_Configuration_Guide_Chapter.html

Understanding VPN Configuration in VRF Mode

In the traditional crypto-connect mode, a VPN is configured by attaching crypto maps to interface VLANs and then crypto-connecting a physical port to the interface VLAN. When configuring a VPN in VRF mode using the IPsec VPN SPA, the model of interface VLANs is preserved, but the **crypto connect vlan** CLI command is not used. When a packet comes into an interface on a specific VRF, the packet must get to the proper interface VLAN. A route must be installed so that packets destined for that particular subnet in that particular VRF are directed to that interface VLAN. This function can be achieved through the following configuration options:

- Configuring an IP address on the interface VLAN that is in the same subnet as the packets' destination IP address. For example, packets are trying to reach subnet 10.1.1.x and their destination IP address is 10.1.1.1 as follows:

```
int vlan 100
 ip vrf forwarding coke
 ip address 10.1.1.254  255.255.255.0 <-- same subnet as 10.1.1.x that we are trying
to reach.
 crypto map mymap
 crypto engine slot 4/1
```

- Configuring a static route as follows:

```
ip route vrf coke 10.1.1.0 255.255.255.0 vlan 100
```

- Configuring routing protocols. You configure BGP, OSPF, or other routing protocols so that remote switches broadcast their routes.

Note Do not configure routing protocols unless you are using tunnel protection.

- Configuring Reverse Route Injection (RRI). You configure RRI so that a route gets installed when the remote end initiates an IPsec session (as in remote access situations).

With VRF mode, the switch sees the interface VLAN as a point-to-point connection; the packets are placed directly onto the interface VLAN. Each VRF has its own interface VLAN.

When a crypto map is attached to an interface VLAN and the **ip vrf forwarding** command has associated that VLAN with a particular VRF, the software creates a point-to-point connection so that all routes pointing to the interface VLAN do not attempt to run the Address Resolution Protocol (ARP). Through normal routing within the VRF, packets to be processed by the IPsec VPN SPA are sent to the interface VLAN. You may configure features on the interface VLAN. The IP address of the interface VLAN must be on the same subnet as the desired destination subnet for packets to be properly routed.

When you enter the **ip vrf forwarding** command on an inside interface, all packets coming in on that interface are routed correctly within that VRF.

When you enable the **crypto engine mode vrf** command and enter the **crypto engine slot outside** command on an interface, a special ACL is installed that forces all incoming Encapsulating Security Payload (ESP)/Authentication Header (AH) IPsec packets addressed to a system IP address to be sent to the IPsec VPN SPA WAN-side port. NAT Traversal (NAT-T) packets are also directed to the IPsec VPN SPA by the special ACL.

Note You must enter the **vrf** *vrf_name* command from within the context of an ISAKMP profile. This command does not apply to the VRF-aware crypto infrastructure; it applies only to generic crypto processing. When the ISAKMP profile is added to a crypto map set, the VRF becomes the default VRF for all of the crypto maps in the list. Individual crypto maps may override this default VRF by specifying another policy profile that contains a different VRF. If no profile is applied to a crypto map tag, it inherits the VRF from the interface if you have configured the interface with the **ip vrf forwarding** command.

All packets destined for a protected outside interface received in this VRF context are placed on the associated interface VLAN. Similarly, all decapsulated ingress packets associated with this VRF are placed on the appropriate interface VLAN so that they may be routed in the proper VRF context.

VRF Mode Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring a VPN for the IPsec VPN SPA using VRF mode:

Note After enabling or disabling VRF mode using the [**no**] **crypto engine mode vrf** command, you must reload the supervisor engine. In addition, MPLS tunnel recirculation must be enabled for VRF mode. That is, you must add the **mls mpls tunnel-recir** command before entering the **crypto engine mode vrf** command.

- The procedure for configuring a VPN in VRF mode varies based on whether you are using tunnel protection or not.
- Unlike IPsec VPN SPA crypto-connect mode configurations, when configuring VPNs in VRF mode, you do not use the **crypto connect vlan** command.
- In Cisco IOS Release 12.2(33)SXH and later releases, the **crypto engine subslot** command used in previous releases has been replaced with the **crypto engine slot** command (of the form **crypto engine slot** *slot/subslot* {**inside** | **outside**}). The **crypto engine subslot** command is no longer supported. In Cisco IOS Release 12.2(33)SXI and later releases, it is not necessary to specify the **slot** *slot/subslot* information with the **outside** keyword. When upgrading, ensure that the **crypto engine** command has been modified in your start-up configuration to avoid extended maintenance time.
- As of Cisco IOS Release 12.2(33)SXH, the **ip vrf forwarding** command is no longer required when configuring GRE with tunnel protection.
- Crypto ACLs support only the EQ operator. Other operators, such as GT, LT, and NEQ, are not supported.

Note When configuring a permit policy for multiple ports with the EQ operator, you must use multiple lines as in this example:

```
permit ip any any port eq 300
permit ip any any port eq 400
permit ip any any port eq 600
```

In Cisco IOS Release 12.2(33)SXH1 and later releases, when configuring a deny policy for multiple ports with the EQ operator, you can use commas to declare the ports as in this example:

```
deny ip any any port eq 300,400,600
```

- Noncontiguous subnets in a crypto ACL, as in the following example, are not supported:

```
deny ip   10.0.5.0   0.255.0.255   10.0.175.0   0.255.0.255
deny ip   10.0.5.0   0.255.0.255   10.0.176.0   0.255.0.255
```

- ACL counters are not supported for crypto ACLs.
- An egress ACL is not applied to packets generated by the route processor. An ingress ACL is not applied to packets destined for the route processor.
- When you create an ISAKMP profile, note the following guidelines regarding the use of the **vrf** command:
  - You must use the **vrf** command if you are using the ISAKMP profile with a crypto map.
  - You are not required to use the **vrf** command if you are using the ISAKMP profile with tunnel protection.
  - You should not use the **vrf** command if you are using the ISAKMP profile with DMVPN.
- When the **ip vrf forwarding** command is applied to a VLAN, any previously existing IP address assigned to that VLAN is removed. To assign an IP address to the VLAN, enter the **ip address** command after the **ip vrf forwarding** command, not preceding it.
- Although more than one IPsec VPN SPA in a chassis is supported beginning with Cisco IOS Release 12.2(18) SXE, in VRF mode, there is no configuration difference between multiple IPsec VPN SPA operation and single IPsec VPN SPA operation. For multiple IPsec VPN SPA operation, the only change is to the output of the **show crypto vlan** command. The following is an example:

```
Interface Tu1 on IPSec Service Module port Gi7/1/1 connected to VRF vrf1
Interface VLAN 2 on IPSec Service Module port Gi7/1/1 connected to VRF vrf2
```

- Applying an ACL to the ingress interface will interfere with the packet flow in releases earlier than Cisco IOS Release 12.2(33) SXI. .

Note Do not apply an ACL during the configuration of VRF mode in releases earlier than Cisco IOS Release 12.2(33) SXI.

- The number of outside interfaces supported by the IPsec VPN SPA is determined by your system resources.
- A loopback interface can be used as tunnel source address.
- A crypto map local address (for example, the interface VLAN address if the crypto map is applied to the interface VLAN) can share the same address as the TP tunnel source address, but it cannot share the same address as a DMVPN tunnel source address.

- In VRF mode, crypto map interfaces that share the same local address must be bound to the same crypto engine.
- When two tunnels share the same tunnel source address, they will be taken over by the IPsec VPN SPA only if one of the following two conditions are met:
  - Both tunnels share the same FVRF.
  - The **crypto engine gre vpnblade** command is entered.
- You can configure the FVRF to be the same as the IVRF.
- In VRF mode, ingress ACLs are installed on crypto engine outside interfaces. In combination with other configured ACLs, these ACLs may cause the ACL-TCAM usage to become excessive. To reduce the TCAM usage, share the TCAM resources by entering the **mls acl tcam share-global** command in the configuration. You can view the ACL usage using the **show tcam counts** command.

Supported and Unsupported Features in VRF Mode

A list of the supported and unsupported features in VRF mode can be found in the "IPsec Feature Support" section on page 20-6. Additional details are as follows:

- Remote access into a VRF (provider edge [PE]) is supported with the following:
  - Reverse Route Injection (RRI) only with crypto maps
  - Proxy AAA (one VRF is proxied to a dedicated AAA)
- Customer edge-provider edge (CE-PE) encryption using tunnel protection is supported with the following:
  - Routing update propagation between CEs
  - IGP/eBGP routing update propagation between the PE and CEs

Configuring VPNs in VRF Mode without Tunnel Protection

To configure a VPN in VRF mode with crypto maps and without tunnel protection, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mls mpls tunnel-recir`	Enables tunnel-MPLS recirculation.
Step 2	`Router(config)# crypto engine mode vrf`	Enables VRF mode for the IPsec VPN SPA. **Note** After enabling or disabling VRF mode using the **crypto engine mode vrf** command, you must reload the supervisor engine.
Step 3	`Router(config)# ip vrf vrf-name`	Configures a VRF routing table and enters VRF configuration mode. • *vrf-name*—Name assigned to the VRF.

	Command	Purpose
Step 4	Router(config-vrf)# **rd** *route-distinguisher*	Creates routing and forwarding tables for a VRF. • *route-distinguisher*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1).
Step 5	Router(config-vrf)# **route-target export** *route-target-ext-community*	Creates lists of export route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 4.
Step 6	Router(config-vrf)# **route-target import** *route-target-ext-community*	Creates lists of import route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 4.
Step 7	Router(config-vrf)# **exit**	Exits VRF configuration mode.
Step 8	Router(config)# **crypto keyring** *keyring-name* [**vrf** *fvrf-name*]	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode. • *keyring-name*—Name of the crypto keyring. • *fvrf-name*—(Optional) Front door virtual routing and forwarding (FVRF) name to which the keyring will be referenced. fvrf-name must match the FVRF name that was defined during virtual routing and forwarding (VRF) configuration
Step 9	Router(config-keyring)# **pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname*} **key** *key*	Defines a preshared key to be used for IKE authentication. • *address* [*mask*]—IP address of the remote peer or a subnet and mask. • *hostname*—Fully qualified domain name of the peer. • *key*—Specifies the secret key.
Step 10	Router(config-keyring)# **exit**	Exits keyring configuration mode.

	Command	Purpose
Step 11	`Router(config)# crypto ipsec transform-set transform-set-name transform1[transform2[transform3]]`	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. Accepted values are described in the *Cisco IOS Security Command Reference*.
Step 12	`Router(config-crypto-trans)# exit`	Exits crypto transform configuration mode
Step 13	`Router(config)# crypto isakmp policy priority`	Defines an IKE policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
Step 14	`Router(config-isakmp)# authentication pre-share`	Specifies the authentication method with an IKE policy. • **pre-share**—Specifies preshared keys as the authentication method.
Step 15	`Router(config-isakmp)# lifetime seconds`	Specifies the lifetime of an IKE SA. • *seconds*—Number of seconds each SA should exist before expiring. Use an integer from 60 to 86,400 seconds. Default is 86,400 (one day).
Step 16	`Router(config-isakmp)# exit`	Exits ISAKMP policy configuration mode.
Step 17	`Router(config)# crypto isakmp profile profile-name`	Defines an ISAKMP profile and enters ISAKMP profile configuration mode. • *profile-name*—Name of the user profile.
Step 18	`Router(config-isa-prof)# vrf ivrf`	Defines the VRF to which the IPsec tunnel will be mapped. • *ivrf*—Name of the VRF to which the IPsec tunnel will be mapped. Enter the same value specified in Step 3.
Step 19	`Router(config-isa-prof)# keyring keyring-name`	Configures a keyring within an ISAKMP profile. • *keyring-name*—Keyring name. This name must match the keyring name that was defined in global configuration. Enter the value specified in Step 8.

	Command	Purpose
Step 20	`Router(config-isa-prof)# match identity address address [mask] [vrf]`	Matches an identity from a peer in an ISAKMP profile. • *address* [*mask*]—IP address of the remote peer or a subnet and mask. • [*vrf*]—(Optional) This argument is only required when configuring a front door VRF (FVRF). This argument specifies that the address is an FVRF instance.
Step 21	`Router(config-isa-prof)# exit`	Exits ISAKMP profile configuration mode.
Step 22	`Router(config)# access list access-list-number {deny | permit} ip host source host destination`	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Number of the host from which the packet is being sent. • *destination*—Number of the host to which the packet is being sent.
Step 23	`Router(config)# crypto map map-name seq-number ipsec-isakmp`	Creates or modifies a crypto map entry and enters the crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations.
Step 24	`Router(config-crypto-map)# set peer {hostname | ip-address}`	Specifies an IPsec peer in a crypto map entry. • {*hostname* | *ip-address*}—IPsec peer host name or IP address. Enter the value specified in Step 20.
Step 25	`Router(config-crypto-map)# set transform-set transform-set-name`	Specifies which transform sets can be used with the crypto map entry. • *transform-set-name*—Name of the transform set. Enter the value specified in Step 11.
Step 26	`Router(config-crypto-map)# set isakmp-profile profile-name`	Sets the ISAKMP profile name. • *profile-name*—Name of the ISAKMP profile. Enter the value entered in Step 17.

	Command	Purpose
Step 27	Router(config-crypto-map)# **match address** [*access-list-id* | *name*]	Specifies an extended access list for the crypto map entry. • *access-list-id*—Identifies the extended access list by its name or number. Enter the value specified in Step 22. • *name*—(Optional) Identifies the named encryption access list. This name should match the name argument of the named encryption access list being matched.
Step 28	Router(config-crypto-map)# **exit**	Exits crypto map configuration mode.
Step 29	Router(config)# **crypto map** *map-name* **local-address** *interface-id*	Specifies and names an identifying interface to be used by the crypto map for IPsec traffic. • *map-name*—Name that identifies the crypto map set. Enter the value specified in Step 23. • **local-address** *interface-id*—Name of interface that has the local address of the switch. **Note** The local address must belong to the FVRF. **Note** In VRF mode, the VPN feature supports up to 1024 local addresses. This limit is across the chassis (not per VPN module).
Step 30	Router(config)# **interface fastethernet** *slot/port*	Configures a Fast Ethernet interface and enters interface configuration mode.
Step 31	Router(config-if)# **ip vrf forwarding** *vrf-name*	Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF. Enter the value specified in Step 3.
Step 32	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 33	Router(config-if)# **no shutdown**	Enables the interface.
Step 34	Router(config-if)# **interface gigabitethernet** *slot/subslot/port*	Configures a Gigabit Ethernet interface. Match the value specified as the *interface-id* in Step 29.
Step 35	Router(config-if)# **ip vrf forwarding** *vrf-name*	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF.
Step 36	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for an interface. • *address*—IP address. • *mask*—Subnet mask.

	Command	Purpose
Step 37	`Router(config-if)# crypto engine slot slot/subslot outside`	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located. **Note** In Cisco IOS Release 12.2(33)SXI and later releases, do not specify **slot** *slot/subslot* with the **outside** keyword.
Step 38	`Router(config-if)# no shutdown`	Enables the interface.
Step 39	`Router(config-if)# exit`	Exits interface configuration mode.
Step 40	`Router(config)# interface vlan-id`	Configures a VLAN interface and enters interface configuration mode. • *vlan-id*—VLAN identifier.
Step 41	`Router(config-if)# ip vrf forwarding vrf-name`	Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF. Enter the value specified in Step 3.
Step 42	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 43	`Router(config-if)# crypto map map-name`	Applies a previously defined crypto map set to an interface. • *map-name*—Name that identifies the crypto map set. Enter the value specified in Step 232.
Step 44	`Router(config-if)# crypto engine slot slot/subslot inside`	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located.
Step 45	`Router(config-if)# exit`	Exits interface configuration mode.
Step 46	`Router(config)# ip route vrf vrf-name prefix mask interface-number`	Establishes static routes for a VRF. • *vrf-name*—Name of the VRF for the static route. Enter the value specified in Step 3. • *prefix*—IP route prefix for the destination, in dotted-decimal format. • *mask*—Prefix mask for the destination, in dotted decimal format. • *interface-number*—Number identifying the network interface to use. Enter the *vlan-id* value specified in Step 40.
Step 47	`Router(config)# end`	Returns to privileged EXEC mode.

For complete configuration information for VRF-Aware IPsec, refer to this URL:

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_vrf_aware_ipsec_ps6350_TSD_Products_Configuration_Guide_Chapter.html

For a configuration example, see the "VRF Mode Basic Configuration Example" section on page 22-23.

Configuring VPNs in VRF Mode with Tunnel Protection (GRE)

This section describes how to configure a VPN in VRF mode with tunnel protection (TP). Tunnel protection is GRE tunneling in VRF mode.

When you configure IPsec, a crypto map is attached to an interface to enable IPsec. With tunnel protection, there is no need for a crypto map or ACL to be attached to the interface. A crypto policy is attached directly to the tunnel interface. Any traffic routed by the interface is encapsulated in GRE and then encrypted using IPsec. The tunnel protection feature can be applied to point-to-point GRE.

VRF Mode Using Tunnel Protection Configuration Guidelines and Restrictions

When configuring tunnel protection on the IPsec VPN SPA, follow these guidelines and restrictions:

- Do not configure any options (such as sequence numbers or tunnel keys) that prevent the IPsec VPN SPA from seizing the GRE tunnel.
- Do not configure the GRE tunnel keepalive feature.
- When applied to the GRE tunnel interface, the **ip tcp adjust-mss** command is ignored. Apply the command to the ingress LAN interface instead. (CSCsl27876)
- Do not use crypto maps to protect GRE traffic in VRF mode.
- When a crypto map interface and a tunnel protection interface (either VTI or GRE/TP) share the same outside interface, they cannot share the same local source address.
- To avoid fragmentation after encryption, set the tunnel IP MTU to be equal to or less than the egress interface MTU minus the GRE and IPsec overheads. The egress interface MTU must be the smallest MTU of all the active crypto outside interfaces.

To configure a VPN in VRF mode using tunnel protection, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mls mpls tunnel-recir`	Enables tunnel-MPLS recirculation.
Step 2	`Router(config)# crypto engine mode vrf`	Enables VRF mode for the IPsec VPN SPA. **Note** After enabling or disabling VRF mode using the **crypto engine mode vrf** command, you must reload the supervisor engine.
Step 3	`Router(config)# ip vrf vrf-name`	Configures a VRF routing table and enters VRF configuration mode. • *vrf-name*—Name assigned to the VRF.
Step 4	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables for a VRF. • *route-distinguisher*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1).

	Command	Purpose
Step 5	Router(config-vrf)# **route-target export** *route-target-ext-community*	Creates lists of export route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 4.
Step 6	Router(config-vrf)# **route-target import** *route-target-ext-community*	Creates lists of import route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 4.
Step 7	Router(config-vrf)# **exit**	Exits VRF configuration mode.
Step 8	Router(config)# **crypto keyring** *keyring-name* [**vrf** *fvrf-name*]	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode. • *keyring-name*—Name of the crypto keyring. • *fvrf-name*—(Optional) Front door virtual routing and forwarding (FVRF) name to which the keyring will be referenced. *fvrf-name* must match the FVRF name that was defined during virtual routing and forwarding (VRF) configuration.
Step 9	Router(config-keyring)# **pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname*} **key** *key*	Defines a preshared key to be used for IKE authentication. • *address* [*mask*]—IP address of the remote peer or a subnet and mask. • *hostname*—Fully qualified domain name of the peer. • *key*—Specifies the secret key.
Step 10	Router(config-keyring)# **exit**	Exits keyring configuration mode.
Step 11	Router(config)# **crypto ipsec transform-set** *transform-set-name* *transform1*[*transform2*[*transform3*]]	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. Accepted values are described in the *Cisco IOS Security Command Reference*.
Step 12	Router(config-crypto-trans)# **exit**	Exits crypto transform configuration mode

	Command	Purpose
Step 13	`Router(config)#` **`crypto isakmp policy`** *`priority`*	Defines an IKE policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
Step 14	`Router(config-isakmp)#` **`authentication pre-share`**	Specifies the authentication method with an IKE policy. • **pre-share**—Specifies preshared keys as the authentication method.
Step 15	`Router(config-isakmp)#` **`lifetime`** *`seconds`*	Specifies the lifetime of an IKE SA. • *seconds*—Number of seconds each SA should exist before expiring. Use an integer from 60 to 86,400 seconds. Default is 86,400 (one day.)
Step 16	`Router(config-isakmp)#` **`exit`**	Exits ISAKMP policy configuration mode.
Step 17	`Router(config)#` **`crypto isakmp profile`** *`profile-name`*	Defines an ISAKMP profile and enters ISAKMP profile configuration mode • *profile-name*—Name of the user profile.
Step 18	`Router(config-isa-prof)#` **`keyring`** *`keyring-name`*	Configures a keyring within an ISAKMP profile. • *keyring-name*—Keyring name. This name must match the keyring name that was defined in global configuration. Enter the value specified in Step 8.
Step 19	`Router(config-isa-prof)#` **`match identity address`** *`address`* `[`*`mask`*`]`	Matches an identity from a peer in an ISAKMP profile. • *address* [*mask*]—IP address of the remote peer or a subnet and mask.
Step 20	`Router(config-isa-prof)#` **`exit`**	Exits ISAKMP profile configuration mode.
Step 21	`Router(config)#` **`access list`** *`access-list-number`* `{`**`deny`** `|` **`permit`**`}` **`ip host`** *`source`* **`host`** *`destination`*	Defines an extended IP access list. • *access-list-number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Number of the host from which the packet is being sent. • *destination*—Number of the host to which the packet is being sent.
Step 22	`Router(config)#` **`crypto ipsec profile`** *`profile-name`*	Defines an IPsec profile and enters IPsec profile configuration mode. • *profile-name*—Name of the user profile.

	Command	Purpose
Step 23	Router(config-ipsec-profile)# **set transform-set** *transform-set-name*	Specifies which transform sets can be used with the crypto map entry. • *transform-set-name*—Name of the transform set. Enter the value specified in Step 11.
Step 24	Router(config-ipsec-profile)# **set isakmp-profile** *profile-name*	Sets the ISAKMP profile name. • *profile-name*—Name of the ISAKMP profile. Enter the value entered in Step 17.
Step 25	Router(config-ipsec-profile)# **exit**	Exits IPsec profile configuration mode.
Step 26	Router(config)# **interface** *tunnel-number*	Configures a tunnel interface and enters interface configuration mode. • *tunnel-number*—Name assigned to the tunnel interface.
Step 27	Router(config-if)# **ip vrf forwarding** *vrf-name*	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF. Enter the value specified in Step 3.
Step 28	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 29	Router(config-if)# **tunnel source** *ip-address*	Sets the source address of a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel.
Step 30	Router(config-if)# **tunnel vrf** *vrf-name*	(Optional) Associates a VPN routing and forwarding instance (VRF) with a specific tunnel destination, interface or subinterface. This step is only required when configuring a front door VRF (FVRF). • *vrf-name*—Name assigned to the VRF.
Step 31	Router(config-if)# **tunnel destination** *ip*-address	Sets the destination address of a tunnel interface. • *ip-address*—IP address to use as the destination address for packets in the tunnel.
Step 32	Router(config-if)# **tunnel protection ipsec** *crypto-policy-name*	Associates a tunnel interface with an IPsec profile. • *crypto-policy-name*—The value as specified in Step 22.
Step 33	Router(config-if)# **crypto engine slot** *slot/subslot* **inside**	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located.
Step 34	Router(config-if)# **interface fastethernet** *slot/subslot*	Configures a Fast Ethernet interface.

	Command	Purpose
Step 35	Router(config-if)# **ip vrf forwarding** *vrf-name*	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF.
Step 36	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for an interface. • *address*—IP address. • *mask*—Subnet mask.
Step 37	Router(config-if)# **no shutdown**	Enables the interface.
Step 38	Router(config-if)# **interface** *type slot/subslot/port*	Configures the physical egress interface.
Step 39	Router(config-if)# **ip vrf forwarding** *vrf-name*	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF.
Step 40	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for an interface. • *address*—IP address. Enter the value specified in Step 29. • *mask*—Subnet mask.
Step 41	Router(config-if)# **crypto engine slot** *slot/subslot* **outside**	Assigns the crypto engine to the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located. **Note** In Cisco IOS Release 12.2(33)SXI and later releases, do not specify **slot** *slot/subslot* with the **outside** keyword.
Step 42	Router(config-if)# **no shutdown**	Enables the interface.
Step 43	Router(config-if)# **exit**	Exits interface configuration mode.

For a configuration example, see the "VRF Mode Tunnel Protection Configuration Example" section on page 22-33.

Configuring an IPsec Virtual Tunnel Interface

The IPsec Virtual Tunnel Interface (VTI) provides a routable interface type for terminating IPsec tunnels that greatly simplifies the configuration process when you need to provide protection for remote access, and provides a simpler alternative to using GRE tunnels and crypto maps with IPsec. In addition, the IPsec VTI simplifies network management and load balancing.

Note IPsec VTI is supported in Cisco IOS Release 12.2(33)SXH and later releases, and is not supported in crypto-connect mode.

Note the following details about IPsec VTI routing and traffic encryption:

- You can enable routing protocols on the tunnel interface so that routing information can be propagated over the virtual tunnel. The router can establish neighbor relationships over the virtual tunnel interface. Interoperability with standard-based IPsec installations is possible through the use of the IP ANY ANY proxy. The static IPsec interface will negotiate and accept IP ANY ANY proxies.
- The IPsec VTI supports native IPsec tunneling and exhibits most of the properties of a physical interface.
- In the IPsec VTI, encryption occurs in the tunnel. Traffic is encrypted when it is forwarded to the tunnel interface. Traffic forwarding is handled by the IP routing table, and dynamic or static IP routing can be used to route the traffic to the virtual tunnel interface. Using IP routing to forward the traffic to encryption simplifies the IPsec VPN configuration because the use of ACLs with a crypto map in native IPsec configurations is not required. When IPsec VTIs are used, you can separate applications of NAT, ACLs, and QoS, and apply them to clear text or encrypted text, or both. When crypto maps are used, there is no easy way to specify forced encryption features.

IPsec Virtual Tunnel Interface Configuration Guidelines and Restrictions

When configuring IPsec VTI, follow these guidelines and restrictions:

- A VTI tunnel can terminate either in a VRF (normal VRF mode) or in the global context (with no **ip vrf forwarding** command on the tunnel interface).
- Only static VTI is supported.
- Only strict IP ANY ANY proxy is supported.
- The IPsec transform set must be configured only in tunnel mode.
- The IKE security association (SA) is bound to the virtual tunnel interface. Because it is bound to the virtual tunnel interface, the same IKE SA cannot be used for a crypto map.
- When the **mls mpls tunnel-recir** command is applied in a VTI configuration, one reserved VLAN is allocated to each tunnel. As a result, there will be a maximum limit of 1000 VTI tunnels.
- In releases earlier than Cisco IOS Release 12.2(33)SXI, the following guidelines apply:
  - The IPsec virtual tunnel interface is limited to IP unicast, as opposed to GRE tunnels, which have a wider application for IPsec implementation.
  - Multicast over VTI is not supported except for control plane traffic such as routing protocol updates.
- In Cisco IOS Release 12.2(33)SXI and later releases, the following guidelines apply:
  - A static VTI tunnel interface supports multicast traffic.
  - ACLs can be applied to GRE and static VTI tunnel interfaces participating in multicast traffic.
  - Platform QoS features can be applied to GRE and static VTI tunnel interfaces participating in multicast traffic.

Configuring an IPsec Static Tunnel

To configure a static IPsec virtual tunnel interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **mls mpls tunnel-recir**	Enables tunnel-MPLS recirculation.
Step 2	Router(config)# **crypto engine mode vrf**	Enables VRF mode for the IPsec VPN SPA. **Note** After enabling or disabling VRF mode using the **crypto engine mode vrf** command, you must reload the supervisor engine.
Step 3	Router(config)# **ip vrf** *vrf-name*	Configures a VRF routing table and enters VRF configuration mode. • *vrf-name*—Name assigned to the VRF.
Step 4	Router(config-vrf)# **rd** *route-distinguisher*	Creates routing and forwarding tables for a VRF. • *route-distinguisher*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1).
Step 5	Router(config-vrf)# **route-target export** *route-target-ext-community*	Creates lists of export route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 4.
Step 6	Router(config-vrf)# **route-target import** *route-target-ext-community*	Creates lists of import route-target extended communities for the specified VRF. • *route-target-ext-community*—Specifies an autonomous system number (ASN) and an arbitrary number (for example, 101:3) or an IP address and an arbitrary number (for example, 192.168.122.15:1). Enter the *route-distinguisher* value specified in Step 4.
Step 7	Router(config-vrf)# **exit**	Exits VRF configuration mode.

	Command	Purpose
Step 8	`Router(config)#` **`crypto keyring`** *`keyring-name`* `[`**`vrf`** *`fvrf-name`*`]`	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode. • *keyring-name*—Name of the crypto keyring. • *fvrf-name*—(Optional) Front door virtual routing and forwarding (FVRF) name to which the keyring will be referenced. *fvrf-name* must match the FVRF name that was defined during virtual routing and forwarding (VRF) configuration.
Step 9	`Router(config-keyring)#` **`pre-shared-key`** `{`**`address`** *`address`* `[`*`mask`*`] |` **`hostname`** *`hostname`*`}` **`key`** *`key`*	Defines a preshared key to be used for IKE authentication. • *address* [*mask*]—IP address of the remote peer or a subnet and mask. • *hostname*—Fully qualified domain name of the peer. • *key*—Specifies the secret key.
Step 10	`Router(config-keyring)#` **`exit`**	Exits keyring configuration mode.
Step 11	`Router(config)#` **`crypto ipsec transform-set`** *`transform-set-name`* *`transform1`*`[`*`transform2`*`[`*`transform3`*`]]`	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform-set-name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. Accepted values are described in the *Cisco IOS Security Command Reference*.
Step 12	`Router(config-crypto-trans)#` **`exit`**	Exits crypto transform configuration mode
Step 13	`Router(config)#` **`crypto isakmp policy`** *`priority`*	Defines an IKE policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
Step 14	`Router(config-isakmp)#` **`authentication pre-share`**	Specifies the authentication method with an IKE policy. • **pre-share**—Specifies preshared keys as the authentication method.
Step 15	`Router(config-isakmp)#` **`lifetime`** *`seconds`*	Specifies the lifetime of an IKE SA. • *seconds*—Number of seconds each SA should exist before expiring. Use an integer from 60 to 86,400 seconds. Default is 86,400 (one day.)
Step 16	`Router(config-isakmp)#` **`exit`**	Exits ISAKMP policy configuration mode.

	Command	Purpose
Step 17	Router(config)# **crypto ipsec profile** *profile-name*	Defines an IPsec profile and enters IPsec profile configuration mode. The IPsec profile defines the IP Security (IPsec) parameters that are to be used for IPsec encryption between two IPsec routers. • *profile-name*—Name of the user profile.
Step 18	Router(config-ipsec-profile)# **set transform-set** *transform-set-name* [*transform-set-name2 ...transform-set-name6*]	Specifies which transform sets can be used with the crypto map entry. • *transform-set-name*—Name of the transform set.
Step 19	Router(config)# **interface** *type slot*/[*subslot*]/*port*	Configures an interface type. • *type*—Type of interface being configured. • *slot*/[*subslot*]/*port*—Number of the slot, subslot (optional), and port to be configured.
Step 20	Router(config-if)# **ip vrf forwarding** *vrf-name*	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF.
Step 21	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for an interface. • *address*—IP address. • *mask*—Subnet mask.
Step 22	Router(config-if)# **tunnel mode ipsec ipv4**	Defines the mode for the tunnel as IPsec and the transport as IPv4.
Step 23	Router(config-if)# **tunnel source** *ip-address*	Sets the source address of a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel.
Step 24	Router(config-if)# **tunnel destination** *ip-address*	Sets the destination address of a tunnel interface. • *ip-address*—IP address to use as the destination address for packets in the tunnel.
Step 25	Router(config-if)# **tunnel vrf** *vrf-name*	(Optional) Associates a VPN routing and forwarding instance (VRF) with a specific tunnel destination. This step is only required when configuring a front door VRF (FVRF). • *vrf-name*—Name assigned to the VRF.
Step 26	Router(config-if)# **tunnel protection ipsec profile** *name*	Associates a tunnel interface with an IPsec profile. • *name*—Name of the IPsec profile; this value must match the name specified in the **crypto ipsec profile** command in Step 1.
Step 27	Router(config-if)# **crypto engine slot** *slot/subslot* **inside**	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located.
Step 28	Router(config-if)# **interface** *type slot/subslot/port*	Configures the physical egress interface.

	Command	Purpose
Step 29	`Router(config-if)# ip vrf forwarding vrf-name`	(Optional) Associates a VRF with an interface or subinterface. • *vrf-name*—Name assigned to the VRF.
Step 30	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for an interface. • *address*—IP address. Enter the value specified in Step 23. • *mask*—Subnet mask.
Step 31	`Router(config-if)# crypto engine outside`	Assigns the crypto engine to the interface.
Step 32	`Router(config-if)# no shutdown`	Enables the interface.
Step 33	`Router(config-if)# exit`	Exits interface configuration mode.

Verifying the IPsec Virtual Tunnel Interface Configuration

To confirm that your IPsec virtual tunnel interface configuration is working properly, enter the **show interfaces tunnel**, **show crypto session**, and **show ip route** commands.

The **show interfaces tunnel** command displays tunnel interface information, the **show crypto session** command displays status information for active crypto sessions, and the **show ip route** command displays the current state of the routing table.

In this display the Tunnel 0 is up and the line protocol is up. If the line protocol is down, the session is not active.

```
Router1# show interfaces tunnel 0

Tunnel0 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.0.51.203/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
reliability 255/255, txload 103/255, rxload 110/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.0.149.203, destination 10.0.149.217
Tunnel protocol/transport IPSEC/IP, key disabled, sequencing disabled
Tunnel TTL 255
Checksumming of packets disabled, fast tunneling enabled
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Tunnel protection via IPSec (profile "P1")
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 1/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
30 second input rate 13000 bits/sec, 34 packets/sec
30 second output rate 36000 bits/sec, 34 packets/sec
191320 packets input, 30129126 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
59968 packets output, 15369696 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
```

```
Router1# show crypto session
Crypto session current status
Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.0.149.217 port 500
IKE SA: local 10.0.149.203/500 remote 10.0.149.217/500 Active
IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
Active SAs: 4, origin: crypto map

Router1# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C 10.0.35.0/24 is directly connected, Ethernet3/3
S 10.0.36.0/24 is directly connected, Tunnel0
C 10.0.51.0/24 is directly connected, Tunnel0
C 10.0.149.0/24 is directly connected, Ethernet3/0
```

For more complete information about IPsec Virtual Tunnel Interface, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gtIPSctm.html

For IPsec Virtual Tunnel Interface configuration examples, see the "IPsec Virtual Tunnel Interfaces Configuration Examples" section on page 22-36.

Configuring VTI in the Global Context

With Cisco IOS Release 12.2(33)SXH and later releases, you can configure IPsec VTI without having to configure VRFs. Although VRF mode must be configured globally using the **crypto engine mode vrf** command, tunnels can be terminated in the global context rather than in VRFs.

The configuration steps for VTI in the global context are similar to the steps for IPsec VTI shown in the "Configuring an IPsec Static Tunnel" section on page 22-18 with the exception that the **ip vrf forwarding** *vrf-name* command and the **tunnel vrf** *vrf-name* command are not required.

For a configuration example of IPsec VTI in the global context, see the "IPsec Virtual Tunnel Interfaces Configuration Examples" section on page 22-36.

Configuration Examples

The following sections provide examples of VRF mode configurations:

- VRF Mode Basic Configuration Example, page 22-23
- VRF Mode Remote Access Using Easy VPN Configuration Example, page 22-26
- VRF Mode PE Configuration Example, page 22-29
- VRF Mode CE Configuration Example, page 22-31
- VRF Mode Tunnel Protection Configuration Example, page 22-33
- IP Multicast in VRF Mode Configuration Example, page 22-34

- IPsec Virtual Tunnel Interfaces Configuration Examples, page 22-36

Note When the **ip vrf forwarding** command is applied to a VLAN, any previously existing IP address assigned to that VLAN is removed. To assign an IP address to the VLAN, enter the **ip address** command after the **ip vrf forwarding** command, not preceding it.

Note The following examples use commands at the level of Cisco IOS Release 12.2(33)SXH.

In Cisco IOS Release 12.2(33)SXH and later releases, the **crypto engine subslot** command used in previous releases has been replaced with the **crypto engine slot** command (of the form **crypto engine slot** *slot/subslot* {**inside** | **outside**}). The **crypto engine subslot** command is no longer supported. In Cisco IOS Release 12.2(33)SXI and later releases, do not specify the **slot** *slot/subslot* information with the **outside** keyword. When upgrading, ensure that this command has been modified in your start-up configuration to avoid extended maintenance time.

VRF Mode Basic Configuration Example

The following example shows a basic IPsec VPN SPA configuration using VRF mode:

Switch 1 Configuration

```
hostname router-1
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto engine mode vrf
!
vlan 2,3
!
crypto keyring key0
  pre-shared-key address 11.0.0.2 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
!
crypto isakmp profile prof1
   vrf ivrf
   keyring key0
   match identity address 11.0.0.2 255.255.255.255
!
!
crypto ipsec transform-set proposal1  esp-3des esp-sha-hmac
!
crypto map testtag local-address Vlan3
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set transform-set proposal1
 set isakmp-profile prof1
 match address 101
```

```
!
interface GigabitEthernet1/1
 !switch inside port
 ip vrf forwarding ivrf
 ip address 12.0.0.1 255.255.255.0
!
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 3
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip vrf forwarding ivrf
 ip address 13.0.0.252 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0 inside
!
interface Vlan3
 ip address 11.0.0.1 255.255.255.0
 crypto engine slot 4/0 outside
!
access-list 101 permit ip host 12.0.0.2 host 13.0.0.2
```

Switch 2 Configuration

```
hostname router-2
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto engine mode vrf
!
vlan 2,3
!
crypto keyring key0
  pre-shared-key address 11.0.0.1 key 12345
```

```
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
!
crypto isakmp profile prof1
   vrf ivrf
   keyring key0
   match identity address 11.0.0.1 255.255.255.255
!
!
crypto ipsec transform-set proposal1  esp-3des esp-sha-hmac
!
crypto map testtag local-address Vlan3
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 set isakmp-profile prof1
 match address 101
!
interface GigabitEthernet1/1
 !switch inside port
 ip vrf forwarding ivrf
 ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 3
 switchport mode access
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip vrf forwarding ivrf
 ip address 12.0.0.252 255.255.255.0
 crypto map testtag
 crypto engine slot 4/0 inside
!
interface Vlan3
 ip address 11.0.0.2 255.255.255.0
 crypto engine slot 4/0 outside
!
```

```
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
```

VRF Mode Remote Access Using Easy VPN Configuration Example

The following examples show VRF mode configurations for remote access using Easy VPN, first using RADIUS authentication, then using local authentication:

Using RADIUS Authentication

```
aaa group server radius acs-vrf1
 server-private 192.1.1.251 auth-port 1812 acct-port 1813 key allegro
 ip vrf forwarding vrf1
!
aaa authentication login test_list group acs-vrf1
aaa authorization network test_list group acs-vrf1
aaa accounting network test_list start-stop group acs-vrf1
!
ip vrf ivrf
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
!
crypto isakmp policy 5
 encr 3des
 authentication pre-share
 group 2

crypto isakmp client configuration group test
 key world
 pool pool1
!
crypto isakmp profile test_pro
   vrf ivrf
   match identity group test
   client authentication list test_list
   isakmp authorization list test_list
   client configuration address respond
   accounting test_list
crypto ipsec transform-set t3 esp-3des esp-sha-hmac
!
crypto dynamic-map remote 1
 set transform-set t3
 set isakmp-profile test_pro
 reverse-route
!
!
crypto map map-ra local-address GigabitEthernet2/1
crypto map map-ra 10 ipsec-isakmp dynamic remote
!
interface GigabitEthernet2/1
  mtu 9216
 ip address 120.0.0.254 255.255.255.0
 ip flow ingress
 logging event link-status
 mls qos trust ip-precedence
 crypto engine slot 1/0 outside
!
interface GigabitEthernet1/0/1
```

```
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,100,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet1/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!

interface Vlan100
 ip vrf forwarding vrf1
 ip address 120.0.0.100 255.255.255.0
 no mop enabled
 crypto map map-ra
 crypto engine slot 1/0 inside

ip local pool pool1 100.0.1.1 100.0.5.250
```

Using Local Authentication

```
username t1 password 0 cisco
aaa new-model
!
aaa authentication login test_list local
aaa authorization network test_list local
!
aaa session-id common
!
ip vrf ivrf
 rd 1:2
 route-target export 1:2
 route-target import 1:2


!
crypto isakmp policy 5
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp client configuration group test
 key world
 pool pool1
crypto isakmp profile test_pro
   vrf ivrf
   match identity group test
   client authentication list test_list
   isakmp authorization list test_list
   client configuration address respond
   accounting test_list
crypto ipsec transform-set t3 esp-3des esp-sha-hmac
!
crypto dynamic-map remote 10
 set transform-set t3
 set isakmp-profile test_pro
 reverse-route

!
!
crypto map map-ra local-address GigabitEthernet2/1
crypto map map-ra 11 ipsec-isakmp dynamic remote
!
!

!
interface GigabitEthernet2/1
  mtu 9216
 ip address 120.0.0.254 255.255.255.0
 ip flow ingress
 logging event link-status
 mls qos trust ip-precedence
 crypto engine slot 1/0 outside
!
!
interface GigabitEthernet1/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,100,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet1/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan100
 ip vrf forwarding ivrf
 ip address 120.0.0.100 255.255.255.0
 ip flow ingress
 crypto map map-ra
 crypto engine slot 1/0 inside
!
!
ip local pool pool1 100.0.1.1 100.0.5.250
```

VRF Mode PE Configuration Example

The following example shows a VRF mode configuration for a provider edge (PE):

```
!
version 12.2
!
hostname EXAMPLE-PE
!
no aaa new-model
ip subnet-zero
!
ip vrf vrf1
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto engine mode vrf
!
redundancy
 mode sso
 main-cpu
  auto-sync running-config
  auto-sync standard
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
power redundancy-mode combined
!
vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
crypto keyring key0
  pre-shared-key address 11.0.0.1 key mykey
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 lifetime 500
crypto isakmp profile prof1
   vrf vrf1
   keyring key0
   self-identity user-fqdn a@example.com
   match identity address 11.0.0.1 255.255.255.255
!
crypto ipsec transform-set proposal1 ah-sha-hmac esp-3des esp-sha-hmac
!
crypto map testtag local-address Vlan3
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set security-association lifetime seconds 1000
 set transform-set proposal1
 set pfs group2
 set isakmp-profile prof1
 match address 101
!
interface GigabitEthernet1/1
 no ip address
 shutdown
!
interface GigabitEthernet1/2
```

```
 switchport
 switchport access vlan 3
 switchport mode access
 no ip address
!
interface GigabitEthernet1/14
 ip vrf forwarding vrf1
 ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet6/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet6/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan none
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet7/1
 no ip address
 shutdown
!
interface GigabitEthernet7/2
 ip address 17.1.5.4 255.255.0.0
 media-type rj45
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan2
 ip vrf forwarding vrf1
 ip address 12.0.0.252 255.255.255.0
 crypto map testtag
 crypto engine subslot 6/0
!
interface Vlan3
 ip address 11.0.0.2 255.255.255.0
 crypto engine subslot 6/0
!
ip classless
ip route 223.255.254.0 255.255.255.0 17.1.0.1
!
no ip http server
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
!
control-plane
!
dial-peer cor custom
!
```

```
line con 0
 exec-timeout 0 0
line vty 0 4
 login
!
end
```

VRF Mode CE Configuration Example

The following example shows a VRF mode configuration for a customer edge (CE):

```
!
version 12.2
!
hostname EXAMPLE-CE
!
no aaa new-model
ip subnet-zero
!
redundancy
 mode sso
 main-cpu
  auto-sync running-config
  auto-sync standard
spanning-tree mode pvst
!
power redundancy-mode combined
!
vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 lifetime 500
crypto isakmp key mykey address 11.0.0.2
!
crypto ipsec transform-set proposal1 ah-sha-hmac esp-3des esp-sha-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.2
 set security-association lifetime seconds 1000
 set transform-set proposal1
 set pfs group2
 match address 101
!
interface GigabitEthernet1/1
 ip address 12.0.0.1 255.255.255.0
 load-interval 30
 no keepalive
!
interface GigabitEthernet1/2
 switchport
 switchport access vlan 3
 switchport mode access
 no ip address
!
interface GigabitEthernet5/2
 ip address 17.1.5.3 255.255.0.0
 media-type rj45
```

```
!
interface GigabitEthernet6/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet6/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 3
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet6/1/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan none
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet6/1/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan none
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan2
 ip address 11.0.0.1 255.255.255.0
 no mop enabled
 crypto map testtag
 crypto engine subslot 6/0
!
interface Vlan3
 no ip address
 crypto connect vlan 2
!
ip classless
ip route 13.0.0.0 255.0.0.0 11.0.0.2
ip route 223.255.254.0 255.255.255.0 17.1.0.1
!
no ip http server
```

```
!
access-list 101 permit ip host 12.0.0.2 host 13.0.0.2
!
control-plane
!
dial-peer cor custom
!
line con 0
 exec-timeout 0 0
line vty 0 4
 login
!
end
```

VRF Mode Tunnel Protection Configuration Example

The following example shows a VRF mode configuration with tunnel protection:

```
ip vrf coke
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto keyring key1
 pre-shared-key address 100.1.1.1 key happy-eddie
!
crypto isakmp policy 1
 authentication pre-share

crypto isakmp profile prof1
 keyring key1
 match identity address 100.1.1.1 255.255.255.255
!
crypto ipsec transform-set TR esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile tp
 set transform-set TR
 set isakmp-profile prof1
!
!
crypto engine mode vrf
!
interface Tunnel1
 ip vrf forwarding coke
 ip address 10.1.1.254 255.255.255.0
 tunnel source 172.1.1.1
 tunnel destination 100.1.1.1
 tunnel protection ipsec profile tp
 crypto engine slot 4/0 inside
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 flowcontrol receive on
 flowcontrol send off
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 cdp enable
 spanning-tree portfast trunk
```

```
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 no ip address
 flowcontrol receive on
 flowcontrol send off
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 cdp enable
 spanning-tree portfast trunk
!
interface GigabitEthernet6/1
 ip address 172.1.1.1 255.255.255.0
 crypto engine slot 4/0 outside
!
interface FastEthernet7/13
 ip vrf forwarding coke
 ip address 13.1.1.2 255.255.255.0
!
ip route 100.1.1.1 255.255.255.255 Tunnel1
```

IP Multicast in VRF Mode Configuration Example

Note If two IPsec VPN SPAs are present in the Cisco 7600 SSC-400, one will be shut down if the **hw-module slot** *X* **subslot** *Y* **only** command is in the configuration. In this case, the IPsec VPN SPA in subslot Y will be active, and the IPsec VPN SPA in the other subslot will be disabled.

The following example shows how to configure IP multicast over GRE:

```
hostname router-1
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
!
!
ip multicast-routing vrf ivrf
!
crypto engine mode vrf
!
!
hw-module slot 4 subslot 0 only
!
crypto keyring key1
  pre-shared-key address 11.0.0.0 255.0.0.0 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp profile isa_prof
   keyring key1
   match identity address 11.0.0.0 255.0.0.0
```

```
!
crypto ipsec transform-set proposal esp-3des
 mode transport
!
crypto ipsec profile vpnprof
 set transform-set proposal
 set isakmp-profile isa_prof
!
!
!
interface Tunnel1
 ip vrf forwarding ivrf
 ip address 20.1.1.1 255.255.255.0
 ip mtu 9216
 ip hold-time eigrp 1 3600
 ip pim sparse-mode
 tunnel source 1.0.1.1
 tunnel destination 11.1.1.1
 tunnel protection ipsec profile vpnprof
 crypto engine slot 4/0 inside
!
interface Loopback1
 ip address 1.0.1.1 255.255.255.0
!
interface GigabitEthernet1/1
 mtu 9216
 ip vrf forwarding ivrf
 ip address 50.1.1.1 255.0.0.0
 ip pim sparse-mode
!
interface GigabitEthernet1/2
 mtu 9216
 ip address 9.1.1.1 255.255.255.0
 crypto engine slot 4/0 outside
!
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
router eigrp 1
 !
 address-family ipv4 vrf ivrf
  autonomous-system 1
  network 20.1.1.0 0.0.0.255
  network 50.1.1.0 0.0.0.255
  no auto-summary
  no eigrp log-neighbor-changes
 exit-address-family
!
router ospf 1
 log-adjacency-changes
 network 1.0.0.0 0.255.255.255 area 0
 network 9.0.0.0 0.255.255.255 area 0
!
ip pim vrf ivrf rp-address 50.1.1.1
!
```

IPsec Virtual Tunnel Interfaces Configuration Examples

The following examples show VRF mode configurations that use VTI:

- IPsec Virtual Tunnel Interface FVRF Configuration Example, page 22-36
- IPsec Virtual Tunnel Interface in the Global Context Configuration Example, page 22-38
- IPsec Virtual Tunnel Interface Multicast Configuration Example, page 22-39

IPsec Virtual Tunnel Interface FVRF Configuration Example

The following example configuration shows an FVRF VTI configuration:

```
hostname router-1
!
!
ip vrf fvrf
 rd 2000:1
 route-target export 2000:1
 route-target import 2000:1
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto engine mode vrf
!
crypto keyring key1 vrf fvrf
  pre-shared-key address 11.1.1.1 key cisco47
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
!
crypto isakmp profile isa_prof
   keyring key1
   match identity address 11.1.1.1 255.255.255.255 fvrf

crypto ipsec transform-set proposal esp-3des esp-sha-hmac
!
!
crypto ipsec profile vpnprof
 set transform-set proposal
 set isakmp-profile isa_prof
!
!
!
```

```
!
!
interface Tunnel1
 ip vrf forwarding ivrf
 ip address 20.1.1.1 255.255.255.0
 ip pim sparse-mode
 ip ospf network broadcast
 ip ospf priority 2
 tunnel source 1.0.0.1
 tunnel destination 11.1.1.1
 tunnel mode ipsec ipv4
 tunnel vrf fvrf
 tunnel protection ipsec profile vpnprof
 crypto engine slot 4/0 inside
!
interface Loopback1
 ip vrf forwarding fvrf
 ip address 1.0.0.1 255.255.255.0
!
interface GigabitEthernet1/1
 !switch inside port
 ip vrf forwarding ivrf
 ip address 50.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
 ip vrf forwarding fvrf
 ip address 9.1.1.1 255.255.255.0
 crypto engine slot 4/0 outside
!
interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
router ospf 1 vrf ivrf
 log-adjacency-changes
 network 20.1.1.0 0.0.0.255 area 0
 network 21.1.1.0 0.0.0.255 area 0
 network 50.0.0.0 0.0.0.255 area 0
!
ip classless
ip route vrf fvrf 11.1.1.0 255.255.255.0 9.1.1.254
```

IPsec Virtual Tunnel Interface in the Global Context Configuration Example

The following example configuration shows IPsec VTI configuration in the global context:

```
!
crypto engine mode vrf
!
crypto keyring key1
  pre-shared-key address 14.0.0.2 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share

!
crypto isakmp profile prof1
   keyring key1
   match identity address 14.0.0.2 255.255.255.255
!
crypto ipsec transform-set t-set1 esp-3des esp-sha-hmac
!
crypto ipsec profile prof1
 set transform-set t-set1
 set isakmp-profile prof1
!
!
interface Tunnel1
 ip address 122.0.0.2 255.255.255.0
 tunnel source 15.0.0.2
 tunnel destination 14.0.0.2
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile prof1
 crypto engine slot 2/0 inside
!
interface Loopback2
 ip address 15.0.0.2 255.255.255.0
!

interface GigabitEthernet1/3
 ip address 172.2.1.1 255.255.255.0
 crypto engine slot 2/0 outside
!
interface GigabitEthernet2/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet2/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
```

```
!
ip route 14.0.0.0 255.0.0.0 172.2.1.2
ip route 172.0.0.0 255.0.0.0 172.2.1.2
```

IPsec Virtual Tunnel Interface Multicast Configuration Example

The following example shows how to configure multicast over VTI:

```
hostname router-1
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
!
!
ip multicast-routing vrf ivrf
!
crypto engine mode vrf
!
!
!
crypto keyring key1
  pre-shared-key address 11.0.0.0 255.0.0.0 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp profile isa_prof
   keyring key1
   match identity address 11.0.0.0 255.0.0.0
!
crypto ipsec transform-set proposal esp-3des
!
crypto ipsec profile vpnprof
 set transform-set proposal
 set isakmp-profile isa_prof
!
!
!
interface Tunnel1
 ip vrf forwarding ivrf
 ip address 20.1.1.1 255.255.255.0
 ip mtu 9216
 ip hold-time eigrp 1 3600
 ip pim sparse-mode
 tunnel source 1.0.1.1
 tunnel destination 11.1.1.1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile vpnprof
 crypto engine slot 4/0 inside
!
interface Loopback1
 ip address 1.0.1.1 255.255.255.0
!
interface GigabitEthernet1/1
 mtu 9216
 ip vrf forwarding ivrf
 ip address 50.1.1.1 255.0.0.0
 ip pim sparse-mode
!
interface GigabitEthernet1/2
 mtu 9216
 ip address 9.1.1.1 255.255.255.0
 crypto engine slot 4/0 outside
!
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
router eigrp 1
 !
 address-family ipv4 vrf ivrf
  autonomous-system 1
  network 20.1.1.0 0.0.0.255
  network 50.1.1.0 0.0.0.255
  no auto-summary
  no eigrp log-neighbor-changes
 exit-address-family
!
router ospf 1
 log-adjacency-changes
 network 1.0.0.0 0.255.255.255 area 0
 network 9.0.0.0 0.255.255.255 area 0
!
ip pim vrf ivrf rp-address 50.1.1.1
!
```

CHAPTER 23

Configuring IPsec VPN Fragmentation and MTU

This chapter provides information about configuring IPsec VPN fragmentation and the maximum transmission unit (MTU). It includes the following sections:

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Understanding IPsec VPN Fragmentation and MTU

This section includes the following topics:

Overview of Fragmentation and MTU

When a packet is nearly the size of the maximum transmission unit (MTU) of the physical egress port of the encrypting switch, and it is encapsulated with IPsec headers, it probably will exceed the MTU of the egress port. This condition causes the packet to be fragmented after encryption (post-fragmentation), which requires the IPsec peer to perform reassembly before decryption, degrading its performance. To minimize post-fragmentation, you can set the MTU in the upstream data path to ensure that most fragmentation occurs before encryption (prefragmentation). Prefragmentation for IPsec VPNs avoids performance degradation by shifting the reassembly task from the receiving IPsec peer to the receiving end hosts.

Note In this document, prefragmentation refers to fragmentation prior to any type of encapsulation, such as IPsec or GRE. IPsec prefragmentation refers to fragmentation prior to IPsec encryption.

To ensure prefragmentation in most cases, we recommend the following MTU settings:

- The crypto interface VLAN MTU associated with the IPsec VPN SPA should be set to be equal or less than the egress interface MTU.
- For GRE over IPsec, the IP MTU of the GRE tunnel interface should be set below the egress interface MTU by at least the overhead of IPsec encryption and the 24-byte GRE+IP header (20-byte IP header plus 4-byte GRE header). Because options such as tunnel key (RFC 2890) are not supported, the GRE+IP IP header will always be 24 bytes.

Note The crypto interface VLAN MTU, the egress interface MTU, and the IP MTU of the GRE tunnel interface are all Layer 3 parameters.

The following are additional guidelines for IPsec prefragmentation and MTU in crypto-connect mode:

- If a packet's DF (Don't Fragment) bit is set and the packet exceeds the MTU at any point in the data path, the packet will be dropped. To prevent a packet drop, clear the DF bit by using either policy-based routing (PBR) or the **crypto df-bit clear** command.
- In Cisco IOS Release 12(33)SXH, and earlier releases, the IPsec VPN SPA does not support path MTU discovery (PMTUD) on GRE tunnels using the **tunnel path-mtu-discovery** command. In Cisco IOS Release 12(33)SXI and later releases, PMTUD is supported on GRE tunnels.
- If GRE encapsulation is not taken over by the IPsec VPN SPA, and if the packets exceed the IP MTU of the GRE tunnel interface, the route processor will fragment and encapsulate the packets.

Note If the supervisor engine performs GRE encapsulation, the encapsulated packets will have the DF bit set.

The IPsec and GRE prefragmentation feature differs based on the Cisco IOS release, as described in Table 23-1.

Table 23-1 IPsec and GRE Prefragmentation based on Cisco IOS Release

Cisco IOS Release	Prefragmentation Feature
12.2(18)SXE	A single prefragmentation process occurs for both IPsec and GRE, based on the smaller of the IP MTU and the egress interface MTU. To prevent fragmentation or packet loss, configure the VLAN MTU as the largest predicted GRE packet size (IP length plus GRE overhead), and the egress interface MTU as the largest predicted GRE/IPsec packet size (IP length plus GRE overhead plus IPsec overhead).
12.2(18)SXF	GRE fragmentation and IPsec fragmentation are separate processes. If GRE encapsulation is performed by the IPsec VPN SPA, prefragmentation of outbound packets will be based on the IP MTU of the tunnel interface. After GRE encapsulation is performed by the IPsec VPN SPA, depending on the IPsec prefragmentation settings, further fragmentation may occur. The IPsec fragmentation behavior is unchanged from Cisco IOS Release 12.2(18)SXE, and is based on the IPsec MTU configuration of the egress interface.

Table 23-1 IPsec and GRE Prefragmentation based on Cisco IOS Release (continued)

12.2SXH	Path MTU discovery (PMTUD) is supported in crypto-connect mode.
12.2SXI and later	• Prefragmentation for IPsec is based on the IP MTU of the tunnel or the crypto interface VLAN, not the egress interface. • The IPsec VPN SPA will perform only prefragmentation or postfragmentation, but not both (although the RP may also perform fragmentation). • Postfragmentation of tunnel interfaces is not supported. • PMTUD is supported in crypto-connect and VRF modes. • The **ip tcp adjust-mss** command is supported in crypto-connect and VRF modes on GRE, GRE/TP, and sVTI tunnels.

For general information on fragmentation and MTU issues, see “Resolve IP Fragmentation, MTU, MSS, and PMTUD Issues with GRE and IPsec” at this URL:

http://www.cisco.com/en/US/tech/tk827/tk369/technologies_white_paper09186a00800d6979.shtml

IPsec Prefragmentation

In the IPsec prefragmentation process (also called Look-Ahead Fragmentation, or LAF), the encrypting switch can predetermine the encapsulated packet size from information available in transform sets, which are configured as part of the IPsec security association (SA). IPsec prefragmentation avoids reassembly by the receiving switch before decryption and helps improve overall IPsec traffic throughput by shifting the reassembly task to the end hosts.

A packet will be fragmented before encryption in the following situations:

- (in Cisco IOS Release 12.2(33)SXH and earlier releases) if it is predetermined that the encrypted packet will exceed the MTU of the output interface.
- (in Cisco IOS Release 12.2(33)SXI and later releases) if either of the following conditions is met:
  - the encrypted packet will exceed the MTU of the crypto interface VLAN
  - the clear packet exceeds the tunnel MTU.

Fragmentation in Cisco IOS Release 12.2(33)SXH and Earlier Releases

The fragmentation process differs depending on the IPsec VPN mode and whether GRE or VTI are used, as described in the following sections:

- Fragmentation in Crypto-Connect Mode, page 23-4
- Fragmentation of IPsec (Using Crypto Maps) Packets in VRF Mode, page 23-5
- Fragmentation of GRE Packets with Tunnel Protection in VRF Mode, page 23-7
- Fragmentation in VTIs, page 23-8

In the following fragmentation descriptions, we assume that the DF (Don’t Fragment) bit is not set for packets entering the flowchart. If a packet requires fragmentation and the DF bit is set, the packet will be dropped.

Fragmentation in Crypto-Connect Mode

The following are the relevant MTU settings for fragmentation of packets in crypto-connect mode:

- The MTU of the interface VLAN.

  Prefragmentation of non-GRE traffic by the RP will be based on this MTU.
- The IP MTU of the GRE tunnel.

  Prefragmentation of GRE traffic will be based on this MTU.
- The MTU of the physical egress interface.

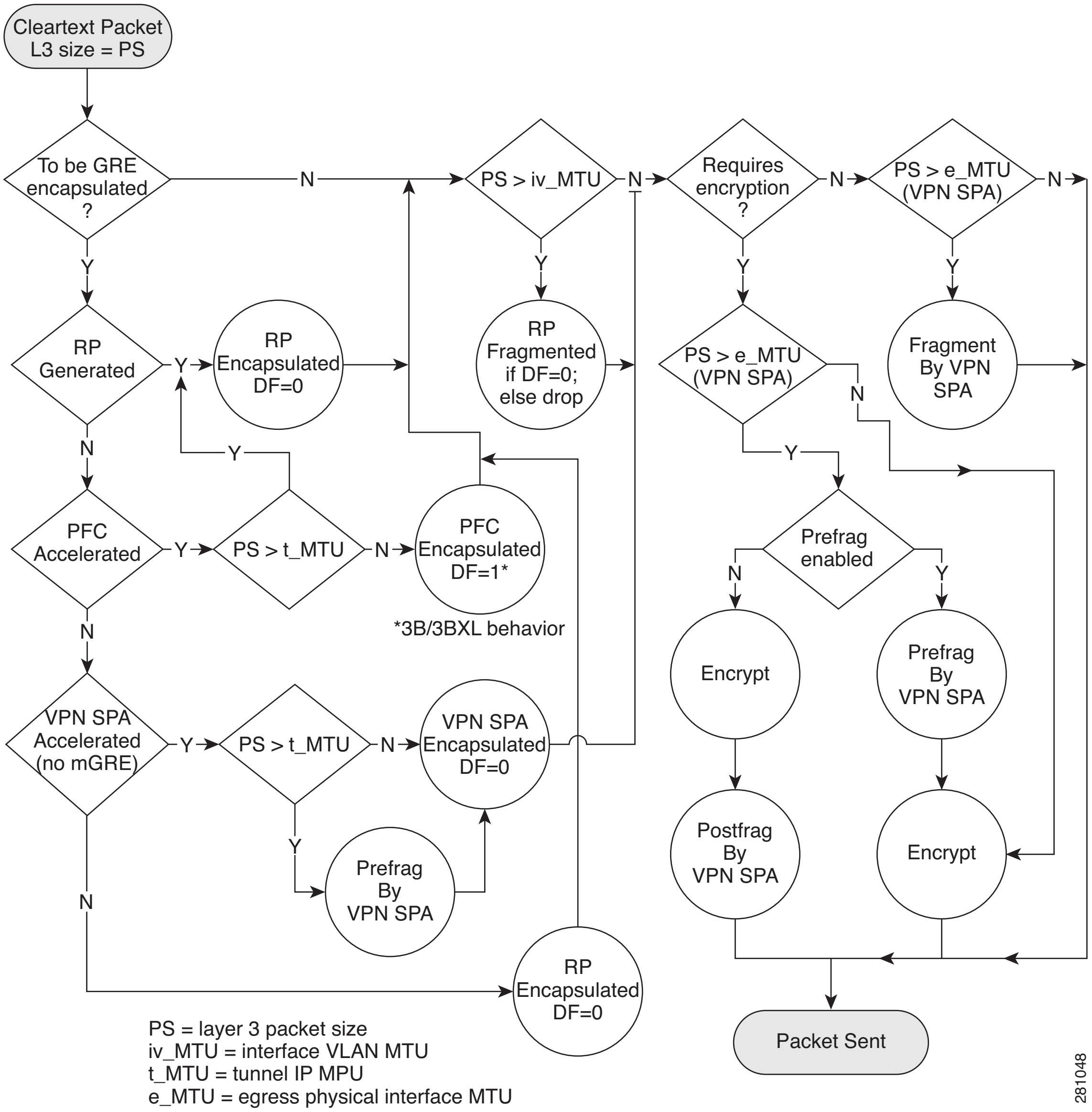
  Pre- and post-fragmentation by the IPsec VPN SPA will be based on this MTU.

Fragmentation will be performed as follows:

- If any packets to be sent to the IPsec VPN SPA exceed the MTU of the interface VLAN, the RP will perform prefragmentation before sending the packets to the IPsec VPN SPA.
- If packets to be GRE encapsulated exceed the IP MTU of the GRE tunnel:
  - The RP will perform prefragmentation when the tunnel is not taken over by the IPsec VPN SPA.
  - The IPsec VPN SPA will perform prefragmentation when the tunnel is taken over by the IPsec VPN SPA.
- If packets to be encrypted will exceed the MTU of the physical egress interface:
  - If IPsec prefragmentation is enabled, the IPsec VPN SPA will perform prefragmentation of the packets. The IPsec VPN SPA will not perform post-fragmentation.
  - If IPsec prefragmentation is disabled, the IPsec VPN SPA will perform post-fragmentation of the encrypted packets. The IPsec VPN SPA will not perform prefragmentation.
- If unencrypted egress packets will exceed the MTU of the physical egress interface, the IPsec VPN SPA will perform fragmentation of the packets.

Figure 23-1 shows the fragmentation process for packets in crypto-connect mode.

Figure 23-1 Fragmentation of Packets in Crypto-Connect Mode



Fragmentation of IPsec (Using Crypto Maps) Packets in VRF Mode

The following are the relevant MTU settings for fragmentation of IPsec traffic in VRF mode:

- The MTU of the interface VLAN.

  Prefragmentation by the RP will be based on this MTU.

- The MTU of the physical egress interface.

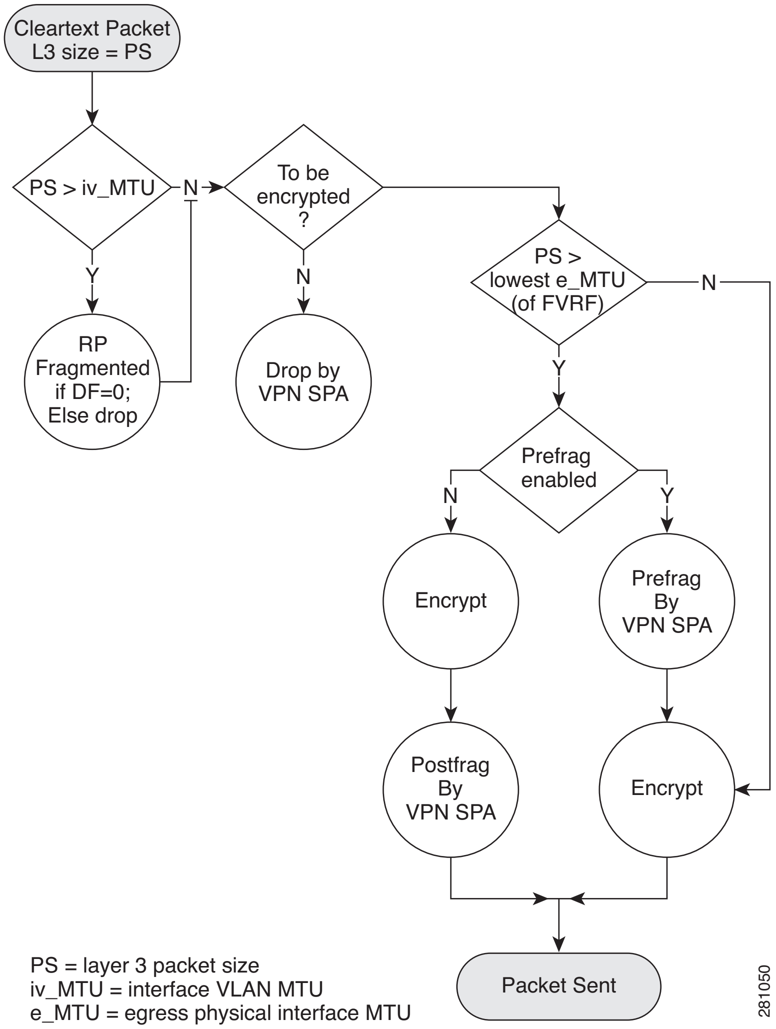
  Pre- and post-fragmentation by the IPsec VPN SPA will be based on this MTU.

Fragmentation will be performed as follows:

- If packets exceed the MTU of the interface VLAN, the RP will perform prefragmentation.
- If encrypted egress packets will exceed the lowest MTU of any physical egress interface on the FVRF:
  - If IPsec prefragmentation is enabled, the IPsec VPN SPA will perform prefragmentation of the packets. The IPsec VPN SPA will not perform post-fragmentation.
  - If IPsec prefragmentation is disabled, the IPsec VPN SPA will perform post-fragmentation of the encrypted packets. The IPsec VPN SPA will not perform prefragmentation.

The fragmentation process for IPsec packets in VRF mode is shown in Figure 23-2.

Figure 23-2 Fragmentation of IPsec Packets in VRF Mode



Fragmentation of GRE Packets with Tunnel Protection in VRF Mode

The following are the relevant MTU settings for fragmentation of GRE traffic with tunnel protection in VRF mode:

- The IP MTU of the GRE tunnel.

  Prefragmentation will be based on this MTU.

- The lowest MTU of any physical egress interface on the FVRF.

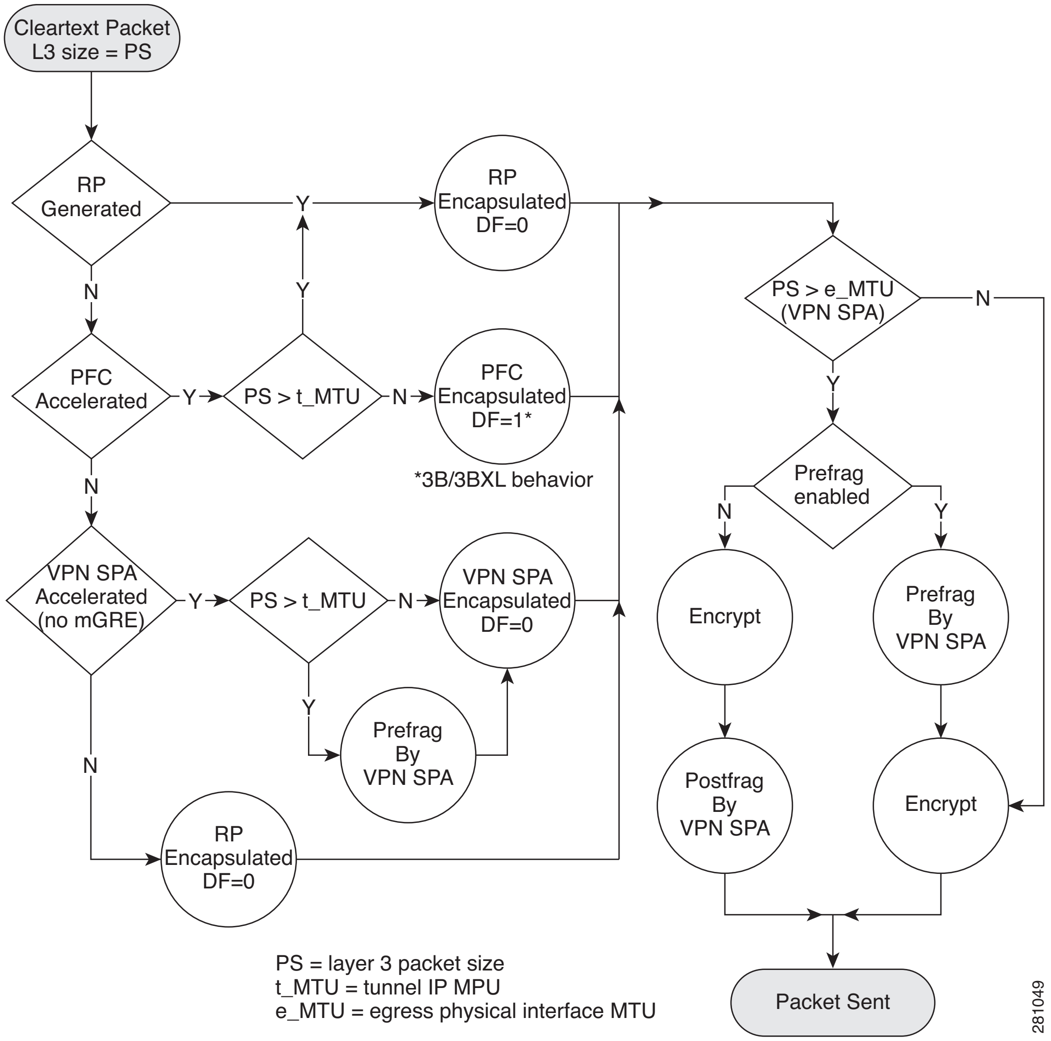
  Pre- and post-fragmentation by the IPsec VPN SPA will be based on this MTU.

Fragmentation will be performed as follows:

- If packets to be encapsulated exceed the IP MTU of the GRE tunnel:
  - The RP will perform prefragmentation when the tunnel is not taken over by the IPsec VPN SPA.
  - The IPsec VPN SPA will perform prefragmentation when the tunnel is taken over by the IPsec VPN SPA.
- If encrypted GRE-encapsulated packets will exceed the lowest MTU of any physical egress interface on the FVRF:
  - If IPsec prefragmentation is enabled, the IPsec VPN SPA will perform prefragmentation of the GRE-encapsulated packets. The IPsec VPN SPA will not perform post-fragmentation.
  - If IPsec prefragmentation is disabled, the IPsec VPN SPA will perform post-fragmentation of the encrypted GRE-encapsulated packets. The IPsec VPN SPA will not perform prefragmentation.

The fragmentation process for GRE packets with tunnel protection in VRF mode is shown in Figure 23-3.

Figure 23-3 Fragmentation of GRE Packets with Tunnel Protection in VRF Mode



Fragmentation in VTIs

The following are the relevant MTU settings for fragmentation of VTI packets:

- The IP MTU of the VTI tunnel interface.

  Prefragmentation will be based on this MTU.

Note We recommend that the IP MTU of the VTI tunnel interface be left at its default value. If you change it, be sure that it does not exceed the MTU of the physical egress interface minus the IPsec overhead.

- The MTU of the physical egress interface.

  Post-fragmentation by the IPsec VPN SPA will be based on this MTU.

Fragmentation will be performed as follows:

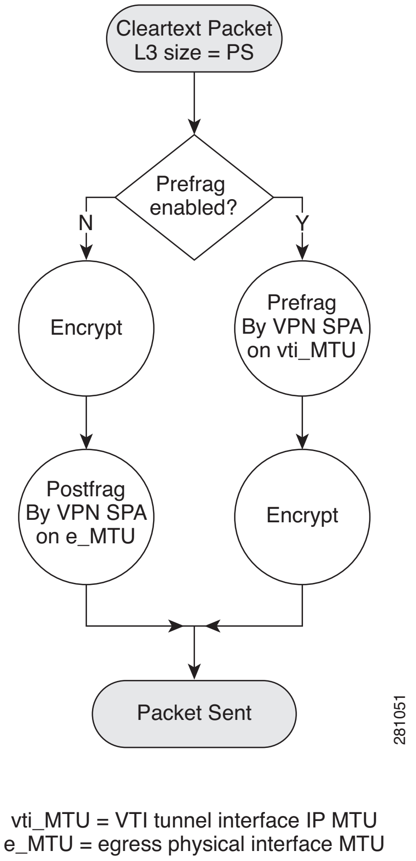
- If IPsec prefragmentation is enabled, the IPsec VPN SPA will perform prefragmentation of packets that exceed the IP MTU of the VTI tunnel interface. The IPsec VPN SPA will not perform post-fragmentation.

Note The RP will perform post-fragmentation of packets that exceed the MTU of the egress interface. This is considered a misconfiguration.

- If IPsec prefragmentation is disabled, the IPsec VPN SPA will perform post-fragmentation of packets that exceed the MTU of the egress interface. The IPsec VPN SPA will not perform prefragmentation.

The fragmentation process for VTI packets is shown in Figure 23-4.

Figure 23-4 Fragmentation of VTI Packets



Fragmentation in Cisco IOS Release 12.2(33)SXI and Later Releases

The fragmentation in Cisco IOS Release 12.2(33)SXI and later releases differs from earlier fragmentation in these significant ways:

- The IPsec VPN SPA will perform only a single fragmentation operation, either prefragmentation or postfragmentation, but not both.
- Fragmentation is based on the IP MTU of the tunnel or of the crypto interface VLAN, not the egress interface.
- Path MTU discovery (PMTUD) is supported in both crypto-connect and VRF modes.
- The **ip tcp adjust-mss** command is supported in all modes.

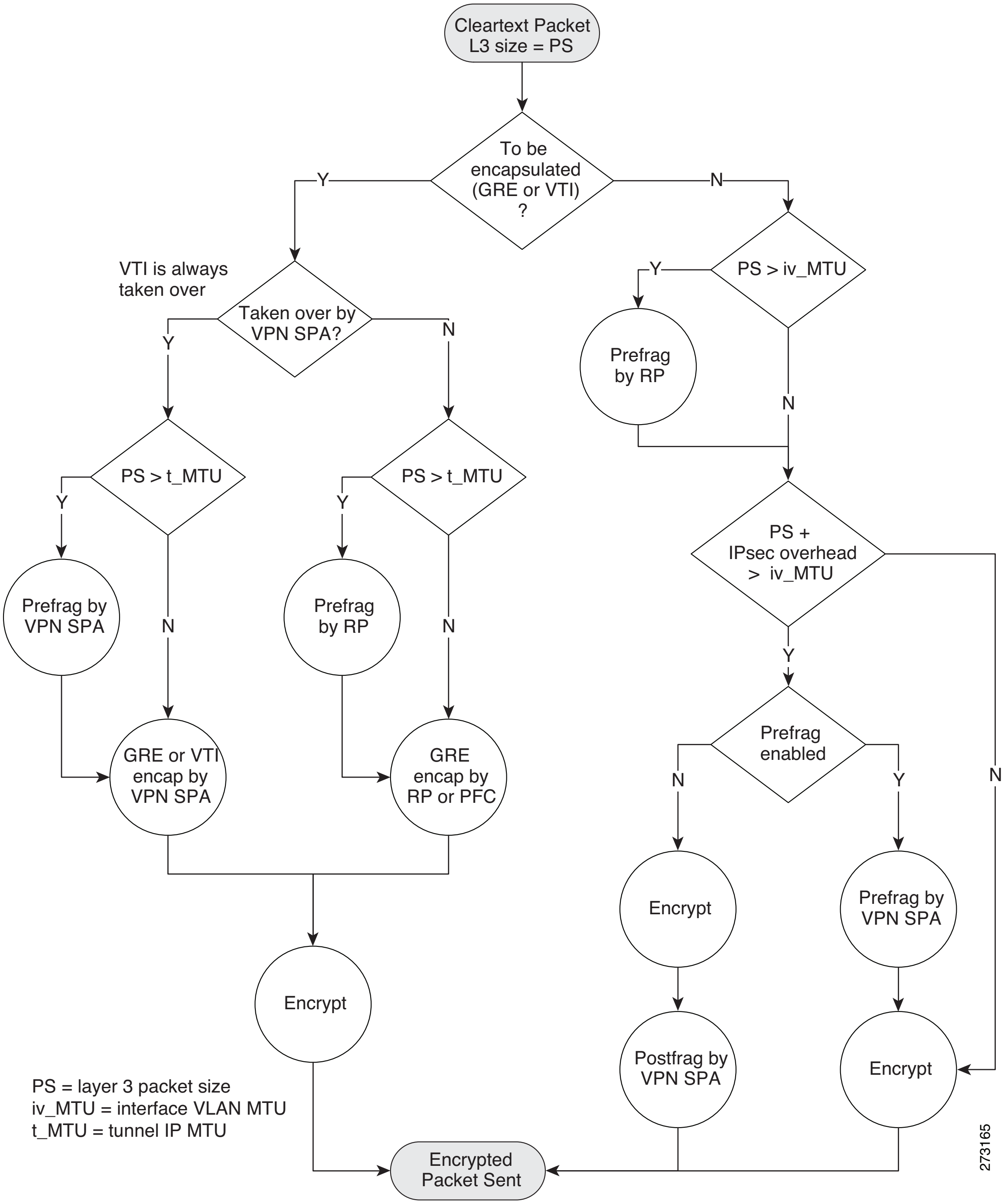
As in earlier releases, the fragmentation process differs depending on the IPsec VPN mode and whether GRE or VTI is used. The process is described in the following sections:

- Overview of the Fragmentation Process, page 23-11
- Fragmentation of IPsec Packets in Crypto-Connect Mode, page 23-12
- Fragmentation of GRE Packets in Crypto-Connect Mode, page 23-13
- Fragmentation of IPsec Packets in VRF Mode, page 23-14
- Fragmentation of GRE Packets in VRF Mode, page 23-14
- Fragmentation of IPsec Packets Using VTI, page 23-15

Overview of the Fragmentation Process

Figure 23-5 shows the fragmentation process for IPsec packets in all VPN modes.

Figure 23-5 Fragmentation of IPsec Packets in All VPN Modes



These notes apply to the fragmentation process described in Figure 23-5:

- The fragmentation process applies only when the DF (Don't Fragment) bit is not set for cleartext packets entering the flowchart. If a packet requires fragmentation and the DF bit is set, the packet will be dropped.
- VTI encapsulation is always taken over by the IPsec VPN SPA.
- GRE encapsulation of RP-generated packets is never taken over by the IPsec VPN SPA.
- GRE encapsulation of mGRE packets is never taken over by the IPsec VPN SPA.

Fragmentation of IPsec Packets in Crypto-Connect Mode

For fragmentation of packets in crypto-connect mode, the following are the MTU setting requirements and recommendations:

- The configured IP MTU of the interface VLAN
  - Prefragmentation of traffic by the IPsec VPN SPA is based on this MTU.
  - You must configure this MTU to be less than or equal to the minimum MTU of the physical egress interfaces configured on the port VLAN, or packets will be dropped.
- The configured MTU of the LAN interface
  - To avoid fragmentation by the RP, we recommend that you configure the MTU of the LAN interface to be less than or equal to the configured IP MTU of the interface VLAN.

In the following example, a 1500-byte cleartext packet will not be fragmented by the RP, because it is within the MTU of the interface VLAN. The cleartext packet will be fragmented by the IPsec VPN SPA, because the IPsec overhead would cause the encrypted packet to exceed the MTU of the interface VLAN.

A 1600-byte cleartext packet will first be fragmented by the RP, because the packet exceeds the MTU of the interface VLAN. The packet will then be fragmented again by the IPsec VPN SPA, because the IPsec overhead added by the encryption process would cause the encrypted packet to exceed the MTU of the interface VLAN.

```
interface GigabitEthernet1/1
  ! switch inside port
  mtu 9216
  ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
  ! switch outside port
  ! mtu 1500 by default
  switchport
  switchport access vlan 502
  switchport mode access
!
interface Vlan2
  ! interface vlan
  ! mtu 1500 by default
  ip address 11.0.0.2 255.255.255.0
  crypto map testtag
  crypto engine slot 4/0
!
interface Vlan502
  ! port vlan
  no ip address
  crypto connect vlan 2
!
```

Fragmentation of GRE Packets in Crypto-Connect Mode

For fragmentation of packets in crypto-connect mode, the following are the MTU setting requirements and recommendations:

- The configured IP MTU of the crypto interface VLAN
  - You must configure this MTU to be less than or equal to the minimum MTU of the physical egress interfaces configured on the port VLAN, or packets will be dropped.
- The configured MTU of the LAN interface
  - To avoid fragmentation by the RP, we recommend that you configure the MTU of the LAN interface to be less than or equal to the configured IP MTU of the crypto interface VLAN.
- The configured IP MTU of the GRE tunnel interface
  - Prefragmentation of traffic by the IPsec VPN SPA is based on this MTU.
  - You must set this MTU so that IPsec-encrypted GRE packets will not exceed the IP MTU of the crypto interface VLAN, or packets will be dropped. This requirement applies regardless of whether the GRE tunnel is taken over by the IPsec VPN SPA.

In the following example, if the tunnel is taken over by the IPsec VPN SPA, a 1600-byte cleartext packet will be fragmented by the IPsec VPN SPA, because the packet exceeds the IP MTU of the tunnel interface. The fragmented packet will then be GRE-encapsulated and IPsec-encrypted by the IPsec VPN SPA.

If the tunnel is not taken over by the IPsec VPN SPA, a 1600-byte cleartext packet will be fragmented by the RP, because the packet exceeds the IP MTU of the tunnel interface. The fragmented packet will then be GRE-encapsulated by the PFC and IPsec-encrypted by the IPsec VPN SPA.

```
interface Tunnel1
  ip mtu 1400
  ip address 1.0.0.1 255.255.255.0
  tunnel source Vlan2
  tunnel destination 11.0.0.2
!
interface GigabitEthernet1/1
  ! switch inside port
  mtu 9216
  ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
  ! switch outside port
  ! mtu 1500 by default
  switchport
  switchport access vlan 502
  switchport mode access
!
interface Vlan2
  ! mtu 1500 by default
  ip address 11.0.0.1 255.255.255.0
  no mop enabled
  crypto map testtag
  crypto engine slot 4/0
!
interface Vlan502
  no ip address
  crypto connect vlan 2
!
```

Fragmentation of IPsec Packets in VRF Mode

For fragmentation of packets in VRF mode, the following are the MTU setting requirements and recommendations:

- The MTU of the crypto interface VLAN.
  - Prefragmentation by the IPsec VPN SPA will be based on this MTU.
  - You must configure this MTU to be less than or equal to the minimum MTU of the physical egress interfaces, or packets will be dropped.
- The configured MTU of the LAN interface
  - To avoid fragmentation by the RP, we recommend that you configure the MTU of the LAN interface to be less than or equal to the configured IP MTU of the crypto interface VLAN.

In the following example, a 1500-byte cleartext packet will not be fragmented by the RP, because it is within the MTU of the interface VLAN. The cleartext packet will be fragmented by the IPsec VPN SPA, because the IPsec overhead would cause the encrypted packet to exceed the MTU of the interface VLAN.

A 1600-byte cleartext packet will first be fragmented by the RP, because the packet exceeds the MTU of the interface VLAN. The packet will then be fragmented again by the IPsec VPN SPA, because the IPsec overhead added by the encryption process would cause the encrypted packet to exceed the MTU of the interface VLAN.

```
interface GigabitEthernet1/1
  ! switch inside port
  mtu 9216
  ip vrf forwarding ivrf
  ip address 12.0.0.1 255.255.255.0
!
!
interface GigabitEthernet1/2
  ! switch outside port
  ! mtu 1500 by default
  ip address 11.0.0.1 255.255.255.0
  crypto engine slot 4/0 outside
!
interface Vlan2
  ! mtu 1500 by default
  ip vrf forwarding ivrf
  ip address 13.0.0.252 255.255.255.0
  crypto map testtag
  crypto engine slot 4/0 inside
!
```

Fragmentation of GRE Packets in VRF Mode

For fragmentation of packets in VRF mode, the following are the MTU setting requirements and recommendations:

- The MTU of the crypto interface VLAN.
  - You must configure this MTU to be less than or equal to the minimum MTU of the physical egress interfaces configured on the port VLAN, or packets will be dropped.
- The configured MTU of the LAN interface
  - To avoid fragmentation by the RP, we recommend that you configure the MTU of the LAN interface to be less than or equal to the configured IP MTU of the crypto interface VLAN.
- The configured IP MTU of the GRE tunnel interface

  - Prefragmentation by the IPsec VPN SPA will be based on this MTU.
  - You must set this MTU so that IPsec-encrypted GRE packets will not exceed the minimum MTU of the physical egress interfaces, or packets will be dropped. This requirement applies regardless of whether the GRE tunnel is taken over by the IPsec VPN SPA.

In the following example, if the tunnel is taken over by the IPsec VPN SPA, a 1600-byte cleartext packet will be fragmented by the IPsec VPN SPA, because the packet exceeds the IP MTU of the tunnel interface. The fragmented packet will then be GRE-encapsulated and IPsec-encrypted by the IPsec VPN SPA.

If the tunnel is not taken over by the IPsec VPN SPA, a 1600-byte cleartext packet will be fragmented by the RP, because the packet exceeds the IP MTU of the tunnel interface. The fragmented packet will then be GRE-encapsulated by the PFC and IPsec-encrypted by the IPsec VPN SPA.

```
interface Tunnel1
  ip mtu 1400
  ip vrf forwarding coke
  ip address 10.1.1.254 255.255.255.0
  tunnel source 172.1.1.1
  tunnel destination 100.1.1.1
  tunnel protection ipsec profile tp
  crypto engine slot 4/0 inside
!
interface GigabitEthernet6/1
  ! switch outside port
  ! mtu 1500 by default
  ip address 172.1.1.1 255.255.255.0
  crypto engine slot 4/0 outside
!
interface FastEthernet7/13
  ! switch inside port
  mtu 9216
  ip vrf forwarding coke
  ip address 13.1.1.2 255.255.255.0
!
```

Fragmentation of IPsec Packets Using VTI

The following are the relevant MTU settings for fragmentation of sVTI packets:

- The IP MTU of the VTI tunnel interface.
  - Prefragmentation by the IPsec VPN SPA will be based on this MTU.
  - Configuring this MTU is unnecessary because it is automatically adjusted to accommodate the IPsec overhead.

Note We recommend that the IP MTU of the VTI tunnel interface be left at its default value. If you change it, be sure that it does not exceed the MTU of the physical egress interface minus the IPsec overhead.

The fragmentation behavior using VTI is the same as the behavior shown in the "Fragmentation of GRE Packets in VRF Mode" section on page 23-14 for the case in which the tunnel is taken over by the IPsec VPN SPA.

Configuring IPsec Prefragmentation

IPsec prefragmentation can be configured globally or at the interface level. By default, IPsec prefragmentation is enabled globally. Enabling or disabling IPsec prefragmentation at the interface will override the global configuration.

IPsec Prefragmentation Configuration Guidelines

Note In Cisco IOS Release 12.2(33)SXI and later releases, tunnels support only IPsec prefragmentation; postfragmentation is not supported. The guidelines in this section apply only to an interface to which a crypto map is applied.

When configuring IPsec prefragmentation, follow these guidelines:

- To configure IPsec prefragmentation at the interface level, apply it on the interface to which the crypto map is applied.
- If an IPsec peer is experiencing high CPU utilization with large packet flows, verify that IPsec prefragmentation is enabled (the peer may be reassembling large packets).
- IPsec prefragmentation for IPsec VPNs operates in IPsec tunnel mode. It does not apply in IPsec transport mode.
- IPsec prefragmentation for IPsec VPNs functionality depends on the **crypto ipsec df-bit** configuration of the interface to which the crypto map is applied, and on the incoming packet "do not fragment" (DF) bit state. For general information about IPsec prefragmentation, see the following URL:

  http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftprefrg.html
- GRE+IP encapsulation adds 24 bytes to the packet size. When configuring for prefragmentation based on anticipated GRE overhead, use this value.
- IPsec encryption adds a number of bytes to the packet size depending on the configured IPsec transform set. When configuring for prefragmentation based on anticipated IPsec overhead, use the following table of worst-case IPsec overhead bytes for various IPsec transform sets:

IPsec Transform Set	IPsec Overhead, Maximum Bytes
esp-aes-(256 or 192 or 128) esp-sha-hmac or md5	73
esp-aes (256 or 192 or 128)	61
esp-3des, esp-des	45
esp-(des or 3des) esp-sha-hmac or md5	57
esp-null esp-sha-hmac or md5	45
ah-sha-hmac or md5	44

Configuring IPsec Prefragmentation Globally

IPsec prefragmentation is globally enabled by default. To enable or disable prefragmentation for IPsec VPNs at the global level, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto ipsec fragmentation before-encryption`	Enables prefragmentation for IPsec VPNs globally.
Step 2	`Router(config)# crypto ipsec fragmentation after-encryption`	Disables prefragmentation for IPsec VPNs globally.

Configuring IPsec Prefragmentation at the Interface

IPsec prefragmentation is globally enabled by default. To enable or disable prefragmentation for IPsec VPNs at the interface level, perform this task beginning in interface configuration mode for the interface to which the crypto map is attached:

	Command	Purpose
Step 1	`Router(config-if)# crypto ipsec fragmentation before-encryption`	Enables prefragmentation for IPsec VPNs on the interface.
Step 2	`Router(config-if)# crypto ipsec fragmentation after-encryption`	Disables prefragmentation for IPsec VPNs on the interface.

Note Enabling or disabling IPsec prefragmentation at the interface will override the global configuration.

Verifying the IPsec Prefragmentation Configuration

To verify that IPsec prefragmentation is enabled, consult the interface statistics on the encrypting switch and the decrypting switch. If fragmentation occurs on the encrypting switch, and no reassembly occurs on the decrypting switch, fragmentation is occurring before encryption, which means that the packets are not being reassembled before decryption and the feature is enabled.

To verify that the IPsec prefragmentation feature is enabled, enter the **show running-configuration** command on the encrypting switch. If the feature is enabled, no fragmentation feature will appear in the command output:

```
Router# show running-configuration

crypto isakmp policy 10
 authentication pre-share
crypto isakmp key abcd123 address 25.0.0.7
crypto ipsec transform-set fooprime esp-3des esp-sha-hmac
!!! the postfragmentation feature appears here if IPsec prefragmentation is disabled
crypto map bar 10 ipsec-isakmp
 set peer 25.0.0.7
 set transform-set fooprime
 match address 102
```

If IPsec prefragmentation has been disabled, the postfragmentation feature will appear in the command output:

```
Router# show running-configuration

crypto isakmp policy 10
 authentication pre-share
crypto isakmp key abcd123 address 25.0.0.7
crypto ipsec transform-set fooprime esp-3des esp-sha-hmac
crypto ipsec fragmentation after-encryption
crypto map bar 10 ipsec-isakmp
 set peer 25.0.0.7
 set transform-set fooprime
 match address 102
```

To display the configuration of the encrypting switch interface VLAN, enter the **show running-configuration interface** command. If the IPsec prefragmentation feature is enabled, a prefragmentation statement will appear in the command output:

```
Router# show running-configuration interface vlan2

interface Vlan2
 ip address 15.0.0.2 255.255.255.0
 crypto map testtag
 crypto engine slot 1/0
 crypto ipsec fragmentation before-encryption
```

If the IPsec prefragmentation feature has been disabled at the interface VLAN, a postfragmentation statement will appear in the command output:

```
Router# show running-configuration interface vlan2

interface Vlan2
 ip address 15.0.0.2 255.255.255.0
 crypto map testtag
 crypto engine slot 1/0
 crypto ipsec fragmentation after-encryption end
```

Configuring MTU Settings

The Cisco IOS software allows the configuration of the Layer 3 maximum transmission unit (MTU) of interfaces and VLANs. You should ensure that all MTU values are consistent to avoid unnecessary fragmentation of packets.

Note When configuring MTU, note that the **ip mtu** command applies only to IP protocol traffic. Other Layer 3 protocol traffic will observe the MTU configured by the **mtu** command.

MTU Settings Configuration Guidelines and Restrictions

When configuring MTU settings for an IPsec VPN SPA, follow these guidelines and note these restrictions:

- In Cisco IOS Release 12.2(33)SXH and earlier releases, the MTU value used by the IPsec VPN SPA for fragmentation decisions is based on the MTU value of the secure port as follows:

  - Routed ports—Use the MTU value of their associated secure port.
  - Access ports—Use the MTU value of the secure port associated with their interface VLAN.
  - Trunk ports—Use the MTU value of the secure port associated with their interface VLAN.
- In Cisco IOS Release 12.2(33)SXI and later releases, the MTU value used by the IPsec VPN SPA for fragmentation decisions is based on the IP MTU of the tunnel or of the crypto interface VLAN, not the egress interface. For information on the recommended MTU settings, see the "Fragmentation in Cisco IOS Release 12.2(33)SXI and Later Releases" section on page 23-10.
- If you have GRE tunneling configured, see the "Fragmentation in Cisco IOS Release 12.2(33)SXH and Earlier Releases" section on page 23-3 or the "Fragmentation in Cisco IOS Release 12.2(33)SXI and Later Releases" section on page 23-10 for information on the recommended MTU settings.

Note For additional information on fragmentation of packets, see the "Configuring IPsec Prefragmentation" section on page 23-16.

Changing the Physical Egress Interface MTU

You can configure either the Layer 3 MTU or the IP MTU of the physical egress interface. To change the MTU value on a physical egress interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface** *type*[1] *slot/port*	Enters interface configuration mode for the interface.
Step 2	Router(config-if)# **mtu** *bytes*	Configures the maximum transmission unit (MTU) size for the interface. • *bytes*—The range is 1500 to 9216; the default is 1500.

[1] *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Changing the Tunnel Interface IP MTU

You can configure the IP MTU of the tunnel interface, but you cannot configure the Layer 3 MTU. To change the IP MTU value on a tunnel, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface** *tunnel_name*	Enters interface configuration mode for the tunnel.
Step 2	Router(config-if)# **ip mtu** *bytes*	Configures the IP MTU size for the tunnel. • *bytes*—The minimum is 68; the maximum and the default depend on the interface medium.

Changing the Interface VLAN MTU

You can configure the Layer 3 MTU of the interface VLAN. To change the MTU value on an interface VLAN, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface vlan_ID`	Enters interface configuration mode for the VLAN.
Step 2	`Router(config-if)# mtu bytes`	Configures the MTU size for the interface VLAN. • *bytes*—The range is 64 to 9216; the default is 1500.

Verifying the MTU Size

To verify the MTU size for an interface, enter the **show interface** command or the **show ip interface** command, as shown in the following examples:

To display the MTU value for a secure port, enter the **show interface** command:

```
Router# show interface g1/1

GigabitEthernet1/1 is up, line protocol is up (connected)
Hardware is C6k 1000Mb 802.3, address is 000a.8ad8.1c4a (bia 000a.8ad8.1c4a)
MTU 9216 bytes, BW 1000000 Kbit, DLY 10 usec,
reliability 255/255, txload 1/255, rxload 1/255
...
```

To display the MTU size for an interface VLAN, enter the **show interface** command.

```
Router# show interface vlan2
Vlan2 is up, line protocol is up
  Hardware is EtherSVI, address is 000e.39ad.e700 (bia 000e.39ad.e700)
  Internet address is 192.168.1.1/16
  MTU 1000 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
...
```

To display the IP MTU value for a GRE tunnel, enter the **show ip interface** command:

```
Router# show ip interface tunnel 2

Tunnel2 is up, line protocol is up
Internet address is 11.1.0.2/16
Broadcast address is 255.255.255.255
Address determined by non-volatile memory
MTU is 1450 bytes
...
```

Configuration Examples

The following sections provide examples of IPsec prefragmentation configurations using commands at the level of Cisco IOS Release 12.2(33)SXI:

- Crypto-Connect Mode IPsec Prefragmentation Configuration Example, page 23-21

- VRF Mode with GRE using Tunnel Protection IPsec Prefragmentation Configuration Example, page 23-22

Crypto-Connect Mode IPsec Prefragmentation Configuration Example

The following example shows an IPsec prefragmentation configuration using crypto-connect mode:

```
!
hostname router-1
!
vlan 2,502
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key 12345 address 11.0.0.1
!
!
crypto ipsec transform-set proposal1 esp-3des esp-md5-hmac
!
crypto map testtag 10 ipsec-isakmp
 set peer 11.0.0.1
 set transform-set proposal1
 match address 101
!
!
interface GigabitEthernet1/1
  !switch inside port
  ! mtu 1500 by default
  ip address 13.0.0.1 255.255.255.0
!
interface GigabitEthernet1/2
 !switch outside port
  mtu 1000
  switchport
  switchport access vlan 502
  switchport mode access
!
interface GigabitEthernet4/0/1
  !IPsec VPN SPA inside port
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,2,1002-1005
  switchport mode trunk
  mtu 9216
  flowcontrol receive on
  flowcontrol send off
  spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
  !IPsec VPN SPA outside port
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,502,1002-1005
  switchport mode trunk
  mtu 9216
  flowcontrol receive on
  flowcontrol send off
  spanning-tree portfast trunk
!
interface Vlan2
```

```
 !interface vlan
  mtu 1000
  ip address 11.0.0.2 255.255.255.0
  crypto map testtag
  crypto engine slot 4/0
!
interface Vlan502
  !port vlan
  no ip address
  crypto connect vlan 2
!
ip classless
ip route 12.0.0.0 255.0.0.0 11.0.0.1
!
access-list 101 permit ip host 13.0.0.2 host 12.0.0.2
!
end
```

VRF Mode with GRE using Tunnel Protection IPsec Prefragmentation Configuration Example

The following example shows an IPsec prefragmentation configuration using VRF mode with GRE and tunnel protection:

```
!
hostname router-1
!
ip vrf coke
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto keyring key1
 pre-shared-key address 100.1.1.1 key happy-eddie
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp profile prof1
 keyring key1
 match identity address 100.1.1.1 255.255.255.255
!
crypto ipsec transform-set TR esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile tp
 set transform-set TR
 set isakmp-profile prof1
!
!
crypto engine mode vrf
!
interface Tunnel1
  ip mtu 1400
  ip vrf forwarding coke
  ip address 10.1.1.254 255.255.255.0
  tunnel source 172.1.1.1
  tunnel destination 100.1.1.1
  tunnel protection ipsec profile tp
  crypto engine slot 4/0 inside
!
```

```
interface GigabitEthernet4/0/1
  !IPsec VPN SPA inside port
  flowcontrol receive on
  flowcontrol send off
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,1002-1005
  switchport mode trunk
  cdp enable
  spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
  !IPsec VPN SPA outside port
  no ip address
  flowcontrol receive on
  flowcontrol send off
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,1002-1005
  switchport mode trunk
  cdp enable
  spanning-tree portfast trunk
!
interface GigabitEthernet6/1
  ! mtu 1500 by default
  ip address 172.1.1.1 255.255.255.0
  crypto engine slot 4/0 outside
!
interface FastEthernet7/13
  ip vrf forwarding coke
  ip address 13.1.1.2 255.255.255.0
!
ip route 100.1.1.1 255.255.255.255 Tunnel1
end
```

CHAPTER 24

Configuring IKE Features Using the IPsec VPN SPA

This chapter provides information about configuring Internet Key Exchange (IKE) related features using the IPsec VPN SPA on the Catalyst 6500 Series switch. It includes the following sections:

Note For detailed information on Internet Key Exchange (IKE), refer to the following Cisco IOS documentation:

Cisco IOS Security Configuration Guide, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

Cisco IOS Security Command Reference, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For additional information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Tip To ensure a successful configuration of your VPN using the IPsec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Overview of IKE

Internet Key Exchange (IKE) is a key management protocol standard that is used in conjunction with the IPsec standard. IPsec can be configured without IKE, but IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard.

Note For more detailed information on IKE, refer to the *Cisco IOS Security Configuration Guide.*

IKE automatically negotiates IPsec security associations (SAs) and enables IPsec secure communications without costly manual preconfiguration. Specifically, IKE provides the following benefits:

- Eliminates the need to manually specify all the IPsec security parameters in the crypto maps at both peers.

  Note Beginning in Cisco IOS Release 12.2SXH, manual keying is no longer supported.

- Allows you to specify a lifetime for the IPsec security association (SA).
- Allows encryption keys to change during IPsec sessions.
- Allows IPsec to provide anti-replay services.
- Permits certification authority (CA) support for a manageable, scalable IPsec implementation.
- Allows dynamic authentication of peers.

Because IKE negotiations must be protected, each IKE negotiation begins by agreement of both peers on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated. You must create an IKE policy at each peer participating in the IKE negotiation.

If you do not configure any IKE policies, your switch will use the default policy, which is always set to the lowest priority and contains the default value of each parameter.

After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these SAs apply to all subsequent IKE traffic during the negotiation.

You can configure multiple, prioritized policies on each peer, each with a different combination of parameter values. However, at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

Configuring Advanced Encryption Standard in an IKE Policy Map

The Advanced Encryption Standard (AES) is a privacy transform for IPsec and Internet Key Exchange (IKE) that has been developed to replace the Data Encryption Standard (DES). AES is designed to be more secure than DES. AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length. The algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.

To configure the AES encryption algorithm within an IKE policy map, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto isakmp policy`** *`priority`*	Defines an ISAKMP policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
Step 2	`Router(config-isakmp)#` **`encryption`** `{`**`aes`** `|` **`aes 192`** `|` **`aes 256`**`}`	Specifies the encryption algorithm within an IKE policy. • **aes**—Specifies 128-bit AES as the encryption algorithm. • **aes 192**—Specifies 192-bit AES as the encryption algorithm. • **aes 256**—Specifies 256-bit AES as the encryption algorithm.

Step 3	`...`
	`Router(config-isakmp)#` **`exit`**
		Specifies any other policy values appropriate to your configuration, and then exits ISAKMP policy configuration mode.
		For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.

Verifying the AES IKE Policy

To verify the configuration of the AES IKE policy, enter the **show crypto isakmp policy** command:

```
Router# show crypto isakmp policy

Protection suite of priority 1
encryption algorithm:  AES - Advanced Encryption Standard (256 bit keys).
hash algorithm:        Secure Hash Standard
authentication method: Pre-Shared Key
Diffie-Hellman group:  #1 (768 bit)
lifetime: 3600 seconds, no volume limit

Default protection suite
        encryption algorithm:   DES - Data Encryption Standard (56 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit
```

For an AES configuration example, see the "Advanced Encryption Standard Configuration Example" section on page 24-22.

Configuring ISAKMP Keyrings

A crypto keyring is a collection of preshared and RSA public keys. You can configure a keyring and then associate it with the Internet Security Association and Key Management Protocol (ISAKMP) profile. The crypto ISAKMP profile may contain zero, one, or more than one keyring.

The ISAKMP keyrings feature (also known as the SafeNet IPsec VPN Client Support feature) allows you to use the **local-address** command to limit the scope of an ISAKMP profile or ISAKMP keyring configuration to a local termination address or interface. The benefit of this feature is that different customers can use the same peer identities and ISAKMP keys by using different local termination addresses.

ISAKMP Keyrings Configuration Guidelines and Restrictions

When configuring ISAKMP keyrings, follow these guidelines and restrictions:

- The local address option works only for the primary address of an interface.
- If an IP address is provided, the administrator must ensure that the connection of the peer terminates to the address that is provided.
- If the IP address does not exist on the device, or if the interface does not have an IP address, the ISAKMP profile or ISAKMP keyring will be effectively disabled.

Limiting an ISAKMP Profile to a Local Termination Address or Interface

To configure an ISAKMP profile and limit it to a local termination address or interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto isakmp profile profile-name`	Defines an ISAKMP profile and enters ISAKMP profile configuration mode. • *profile-name*—Name of the ISAKMP profile.
Step 2	`Router(conf-isa-profile)# keyring keyring-name`	(Optional) Configures a keyring with an ISAKMP profile. • *keyring-name*—Name of the crypto keyring. **Note** A keyring is not needed inside an ISAKMP profile for local termination to work. Local termination works even if Rivest, Shamir, and Adelman (RSA) certificates are used.

	Command	Purpose
Step 3	`Router(conf-isa-profile)# `**`match identity address`** *`address`*	Matches an identity from a peer in an ISAKMP profile. • *address*—IP address of the remote peer.
Step 4	`Router(conf-isa-profile)# `**`local-address`** `{`*`interface-name`*` | `*`ip-address`*` [`*`vrf-tag`*`]}`	Limits the scope of an ISAKMP profile or an ISAKMP keyring configuration to a local termination address or interface. • *interface-name*—Name of the local interface. • *ip-address*—Local termination address. • *vrf-tag*—(Optional) Scope of the IP address will be limited to the VRF.

Limiting a Keyring to a Local Termination Address or Interface

To configure an ISAKMP keyring and limit its scope to a local termination address or interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`keyring`** *`keyring-name`*	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode. • *keyring-name*—Name of the crypto keyring.
Step 2	`Router(conf-keyring)# `**`local-address`** `{`*`interface-name`*` | `*`ip-address`*` [`*`vrf-tag`*`]}`	Limits the scope of an ISAKMP profile or an ISAKMP keyring configuration to a local termination address or interface. • *interface-name*—Name of the local interface. • *ip-address*—Local termination address. • *vrf-tag*—(Optional) Scope of the IP address will be limited to the VRF.
Step 3	`Router(conf-keyring)`**`# pre-shared-key address`** *`address`*	Defines a preshared key to be used for IKE authentication. • *address*—IP address.

For complete configuration information for SafeNet IPsec VPN Client Support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gt_scse.html

For ISAKMP keyrings configuration examples, see the "ISAKMP Keyrings Configuration Examples" section on page 24-22.

Configuring Certificate to ISAKMP Profile Mapping

The Certificate to ISAKMP Profile Mapping feature enables you to assign an Internet Security Association and Key Management Protocol (ISAKMP) profile to a peer on the basis of the contents of arbitrary fields in the certificate. In addition, this feature allows you to assign a group name to those peers that are assigned an ISAKMP profile.

Note Certificate to ISAKMP Profile Mapping is only supported as of Cisco IOS Release 12.2(33)SXH.

Certificate to ISAKMP Profile Mapping Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring Certificate to ISAKMP Profile Mapping:

- This feature will not be applicable if you use Rivest, Shamir, and Adelman (RSA)-signature or RSA-encryption authentication without certificate exchange. ISAKMP peers must be configured to do RSA-signature or RSA-encryption authentication using certificates.

Mapping the Certificate to the ISAKMP Profile

To map the certificate to the ISAKMP profile, perform the following task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto isakmp profile`** *`profile-name`*	Defines an ISAKMP profile and enters ISAKMP profile configuration mode • *profile-name*—Name of the user profile.
Step 2	`Router(config-isa-prof)#` **`match certificate`** *`certificate-map`*	Accepts the name of a certificate map. • *certificate-map*—Name of the certificate map.

Verifying the Certificate to ISAKMP Profile Mapping Configuration

To verify that the subject name of the certificate map has been properly configured, enter the **show crypto pki certificates** and the **debug crypto isakmp** commands.

The **show crypto pki certificates** command displays all current IKE security associations (SAs) at a peer. The **debug crypto isakmp** command displays messages about IKE events.

The following examples show that a certificate has been mapped to an ISAKMP profile. The examples include the configurations for the responder and initiator, the **show crypto pki certificates** command output verifying that the subject name of the certificate map has been configured, and the **debug crypto isakmp** command output showing that the certificate has gone through certificate map matching and been matched to the ISAKMP profile.

Responder Configuration

```
crypto pki certificate map cert_map 10
! The above line is the certificate map definition.
```

```
 subject-name co ou = green
! The above line shows that the subject name must have "ou = green."
!
crypto isakmp profile certpro
! The above line shows that this is the ISAKMP profile that will match if the certificate
of the peer matches cert_map (shown on third line below).
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
```

Initiator Configuration

```
crypto ca trustpoint LaBcA
 enrollment url http://10.76.82.20:80/cgi-bin/openscep
 subject-name ou=green,c=IN
! The above line ensures that the subject name "ou = green" is set.
 revocation-check none
```

Command Output for show crypto pki certificates for the Initiator

```
Router# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 21
  Certificate Usage: General Purpose
  Issuer:
    cn=blue-lab CA
    o=CISCO
    c=IN
  Subject:
    Name: Router.cisco.com
    c=IN
    ou=green
! The above line is a double check that "ou = green" has been set as the subject name.
    hostname=Router.cisco.com
  Validity Date:
    start date: 14:34:30 UTC Mar 31 2004
    end   date: 14:34:30 UTC Apr 1 2009
    renew date: 00:00:00 UTC Jan 1 1970
  Associated Trustpoints: LaBcA
```

Command Output for debug crypto isakmp for the Responder

```
Router# debug crypto isakmp

*Nov  6 19:31:25.010: ISAKMP:(0): SA request profile is prof2
*Nov  6 19:31:25.010: ISAKMP: Found a peer struct for 14.0.0.2, peer port 500
*Nov  6 19:31:25.010: ISAKMP: Locking peer struct 0x13884FB8, refcount 349 for
isakmp_initiator
*Nov  6 19:31:25.010: ISAKMP[I]: sa->swdb: Vlan3
*Nov  6 19:31:25.010: ISAKMP: local port 500, remote port 500
*Nov  6 19:31:25.010: ISAKMP: set new node 0 to QM_IDLE
*Nov  6 19:31:25.010: ISAKMP: Find a dup sa in the avl tree during calling isadb_insert sa
= 13C041E8
*Nov  6 19:31:25.010: ISAKMP:(0):Can not start Aggressive mode, trying Main mode.
*Nov  6 19:31:25.010: ISAKMP:(0):Profile has no keyring, aborting key search
*Nov  6 19:31:25.010: ISAKMP:(0): constructed NAT-T vendor-07 ID
*Nov  6 19:31:25.010: ISAKMP:(0): constructed NAT-T vendor-03 ID
*Nov  6 19:31:25.010: ISAKMP:(0): constructed NAT-T vendor-02 ID
*Nov  6 19:31:25.010: ISAKMP:(0):Input = IKE_MESG_FROM_IPSEC, IKE_SA_REQ_MM
*Nov  6 19:31:25.010: ISAKMP:(0):Old State = IKE_READY  New State = IKE_I_MM1
```

```
*Nov  6 19:31:25.010: ISAKMP:(0): beginning Main Mode exchange
*Nov  6 19:31:25.010: ISAKMP:(0): sending packet to 14.0.0.2 my_port 500 peer_port 500 (I)
MM_NO_STATE
*Nov  6 19:31:25.018: ISAKMP (0): received packet from 14.0.0.2 dport 500 sport 500 fvrf
(N) NEW SA
*Nov  6 19:31:25.018: ISAKMP: Found a peer struct for 14.0.0.2, peer port 500
*Nov  6 19:31:25.018: ISAKMP: Locking peer struct 0x13884FB8, refcount 350 for
crypto_isakmp_process_block
*Nov  6 19:31:25.018: ISAKMP[R]: sa->swdb: Vlan2
*Nov  6 19:31:25.018: ISAKMP: local port 500, remote port 500
*Nov  6 19:31:25.018: ISAKMP: Find a dup sa in the avl tree during calling isadb_insert sa
= 148C68D8
*Nov  6 19:31:25.018: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Nov  6 19:31:25.018: ISAKMP:(0):Old State = IKE_READY  New State = IKE_R_MM1

*Nov  6 19:31:25.018: ISAKMP:(0): processing SA payload. message ID = 0
*Nov  6 19:31:25.018: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.018: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
*Nov  6 19:31:25.018: ISAKMP (0): vendor ID is NAT-T v7
*Nov  6 19:31:25.018: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.018: ISAKMP:(0): vendor ID seems Unity/DPD but major 157 mismatch
*Nov  6 19:31:25.018: ISAKMP:(0): vendor ID is NAT-T v3
*Nov  6 19:31:25.018: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.018: ISAKMP:(0): vendor ID seems Unity/DPD but major 123 mismatch
*Nov  6 19:31:25.018: ISAKMP:(0): vendor ID is NAT-T v2
*Nov  6 19:31:25.038: ISAKMP:(0):Checking ISAKMP transform 1 against priority 1 policy
*Nov  6 19:31:25.038: ISAKMP:      encryption 3DES-CBC
*Nov  6 19:31:25.038: ISAKMP:      hash MD5
*Nov  6 19:31:25.038: ISAKMP:      default group 1
*Nov  6 19:31:25.038: ISAKMP:      auth RSA sig
*Nov  6 19:31:25.038: ISAKMP:      life type in seconds
*Nov  6 19:31:25.038: ISAKMP:      life duration (VPI) of  0x0 0x1 0x51 0x80
*Nov  6 19:31:25.042: ISAKMP:(0):atts are acceptable. Next payload is 3
*Nov  6 19:31:25.042: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.042: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
*Nov  6 19:31:25.042: ISAKMP (0): vendor ID is NAT-T v7
*Nov  6 19:31:25.042: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.042: ISAKMP:(0): vendor ID seems Unity/DPD but major 157 mismatch
*Nov  6 19:31:25.042: ISAKMP:(0): vendor ID is NAT-T v3
*Nov  6 19:31:25.042: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.042: ISAKMP:(0): vendor ID seems Unity/DPD but major 123 mismatch
*Nov  6 19:31:25.042: ISAKMP:(0): vendor ID is NAT-T v2
*Nov  6 19:31:25.042: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Nov  6 19:31:25.042: ISAKMP:(0):Old State = IKE_R_MM1  New State = IKE_R_MM1

*Nov  6 19:31:25.046: ISAKMP:(0): constructed NAT-T vendor-07 ID
*Nov  6 19:31:25.046: ISAKMP:(0): sending packet to 14.0.0.2 my_port 500 peer_port 500 (R)
MM_SA_SETUP
*Nov  6 19:31:25.046: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
*Nov  6 19:31:25.046: ISAKMP:(0):Old State = IKE_R_MM1  New State = IKE_R_MM2

*Nov  6 19:31:25.046: ISAKMP (0): received packet from 14.0.0.2 dport 500 sport 500 fvrf
(I) MM_NO_STATE
*Nov  6 19:31:25.046: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Nov  6 19:31:25.046: ISAKMP:(0):Old State = IKE_I_MM1  New State = IKE_I_MM2

*Nov  6 19:31:25.046: ISAKMP:(0): processing SA payload. message ID = 0
*Nov  6 19:31:25.046: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.046: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
*Nov  6 19:31:25.046: ISAKMP (0): vendor ID is NAT-T v7
*Nov  6 19:31:25.046: ISAKMP : Looking for xauth in profile prof2
*Nov  6 19:31:25.046: ISAKMP:(0):Checking ISAKMP transform 1 against priority 1 policy
*Nov  6 19:31:25.046: ISAKMP:      encryption 3DES-CBC
```

```
*Nov  6 19:31:25.046: ISAKMP:      hash MD5
*Nov  6 19:31:25.046: ISAKMP:      default group 1
*Nov  6 19:31:25.046: ISAKMP:      auth RSA sig
*Nov  6 19:31:25.050: ISAKMP:      life type in seconds
*Nov  6 19:31:25.050: ISAKMP:      life duration (VPI) of  0x0 0x1 0x51 0x80
*Nov  6 19:31:25.050: ISAKMP:(0):atts are acceptable. Next payload is 0
*Nov  6 19:31:25.050: ISAKMP:(0): processing vendor id payload
*Nov  6 19:31:25.050: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
*Nov  6 19:31:25.050: ISAKMP (0): vendor ID is NAT-T v7
*Nov  6 19:31:25.050: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Nov  6 19:31:25.050: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM2

*Nov  6 19:31:25.050: ISAKMP (0): constructing CERT_REQ for issuer
cn=mscavpn1,ou=isbu,o=cisco
*Nov  6 19:31:25.054: ISAKMP:(0): sending packet to 14.0.0.2 my_port 500 peer_port 500 (I)
MM_SA_SETUP
*Nov  6 19:31:25.054: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
*Nov  6 19:31:25.054: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM3

*Nov  6 19:31:25.058: ISAKMP (0): received packet from 14.0.0.2 dport 500 sport 500 fvrf
(R) MM_SA_SETUP
*Nov  6 19:31:25.062: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Nov  6 19:31:25.062: ISAKMP:(0):Old State = IKE_R_MM2  New State = IKE_R_MM3

*Nov  6 19:31:25.062: ISAKMP:(0): processing KE payload. message ID = 0
*Nov  6 19:31:25.062: ISAKMP:(0): processing NONCE payload. message ID = 0
*Nov  6 19:31:25.062: ISAKMP:(83727): processing CERT_REQ payload. message ID = 0
*Nov  6 19:31:25.062: ISAKMP:(83727): peer wants a CT_X509_SIGNATURE cert
*Nov  6 19:31:25.066: ISAKMP:(83727): peer want cert issued by cn=mscavpn1,ou=isbu,o=cisco
*Nov  6 19:31:25.066: ISAKMP:(83727): Choosing trustpoint MSCA as issuer
*Nov  6 19:31:25.066: ISAKMP:(83727): processing vendor id payload
*Nov  6 19:31:25.066: ISAKMP:(83727): vendor ID is DPD
*Nov  6 19:31:25.066: ISAKMP:(83727): processing vendor id payload
*Nov  6 19:31:25.066: ISAKMP:(83727): speaking to another IOS box!
*Nov  6 19:31:25.066: ISAKMP:(83727): processing vendor id payload
*Nov  6 19:31:25.066: ISAKMP:(83727): vendor ID seems Unity/DPD but major 230 mismatch
*Nov  6 19:31:25.066: ISAKMP:(83727): vendor ID is XAUTH
*Nov  6 19:31:25.066: ISAKMP (83727): His hash no match - this node outside NAT
*Nov  6 19:31:25.066: ISAKMP (83727): No NAT Found for self or peer
*Nov  6 19:31:25.066: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Nov  6 19:31:25.066: ISAKMP:(83727):Old State = IKE_R_MM3  New State = IKE_R_MM3

*Nov  6 19:31:25.066: ISAKMP (83727): constructing CERT_REQ for issuer
cn=mscavpn1,ou=isbu,o=cisco
*Nov  6 19:31:25.066: ISAKMP:(83727): sending packet to 14.0.0.2 my_port 500 peer_port 500
(R) MM_KEY_EXCH
*Nov  6 19:31:25.070: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
*Nov  6 19:31:25.070: ISAKMP:(83727):Old State = IKE_R_MM3  New State = IKE_R_MM4

*Nov  6 19:31:25.070: ISAKMP (0): received packet from 14.0.0.2 dport 500 sport 500 fvrf
(I) MM_SA_SETUP
*Nov  6 19:31:25.070: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Nov  6 19:31:25.070: ISAKMP:(0):Old State = IKE_I_MM3  New State = IKE_I_MM4

*Nov  6 19:31:25.070: ISAKMP:(0): processing KE payload. message ID = 0
*Nov  6 19:31:25.074: ISAKMP:(0): processing NONCE payload. message ID = 0
*Nov  6 19:31:25.098: ISKAMP: growing send buffer from 1024 to 3072

*Nov  6 19:31:25.118: ISAKMP (83727): received packet from 14.0.0.2 dport 500 sport 500
fvrf (R) MM_KEY_EXCH
*Nov  6 19:31:25.122: ISAKMP:(83727):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Nov  6 19:31:25.122: ISAKMP:(83727):Old State = IKE_R_MM4  New State = IKE_R_MM5

*Nov  6 19:31:25.122: ISAKMP:(83727): processing ID payload. message ID = 0
```

```
*Nov  6 19:31:25.122: ISAKMP (83727): ID payload
        next-payload : 6
        type         : 3
        USER FQDN    : a@vrf2.com
        protocol     : 17
        port         : 500
        length       : 18
*Nov  6 19:31:25.134: ISAKMP:(83727):: peer matches prof2 profile
*Nov  6 19:31:25.134: ISAKMP:(83727): processing CERT payload. message ID = 0
*Nov  6 19:31:25.134: ISAKMP:(83727): processing a CT_X509_SIGNATURE cert
*Nov  6 19:31:25.142: ISAKMP:(83727): peer's pubkey isn't cached
*Nov  6 19:31:25.158: %CRYPTO-6-IKMP_NO_ID_CERT_USER_FQDN_MATCH: ID of a@vrf2.com (type 3) 
and certificate user fqdn with empty
*Nov  6 19:31:25.158: ISAKMP (83727): adding peer's pubkey to cache
*Nov  6 19:31:25.158: ISAKMP:(83727): processing SIG payload. message ID = 0
*Nov  6 19:31:25.162: ISAKMP:(83727):SA authentication status:
        authenticated
*Nov  6 19:31:25.162: ISAKMP:(83727):SA has been authenticated with 14.0.0.2
*Nov  6 19:31:25.162: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Nov  6 19:31:25.162: ISAKMP:(83727):Old State = IKE_R_MM5  New State = IKE_R_MM5

*Nov  6 19:31:25.170: ISAKMP:(83727):SA is doing RSA signature authentication using id 
type ID_USER_FQDN
*Nov  6 19:31:25.170: ISAKMP (83727): ID payload
        next-payload : 6
        type         : 3
        USER FQDN    : a@vrf2.com
        protocol     : 17
        port         : 500
        length       : 18
*Nov  6 19:31:25.170: ISAKMP:(83727):Total payload length: 18
*Nov  6 19:31:25.182: ISAKMP (83727): constructing CERT payload for 
cn=HUB,ou=isbu,o=cisco,hostname=HUB.cisco.com,serialNumber=1234D
*Nov  6 19:31:25.182: ISKAMP: growing send buffer from 1024 to 3072
*Nov  6 19:31:25.186: ISAKMP:(83727): using the MSCA trustpoint's keypair to sign
*Nov  6 19:31:25.194: ISAKMP:(83727): sending packet to 14.0.0.2 my_port 500 peer_port 500 
(R) MM_KEY_EXCH
*Nov  6 19:31:25.198: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
*Nov  6 19:31:25.198: ISAKMP:(83727):Old State = IKE_R_MM5  New State = IKE_P1_COMPLETE

*Nov  6 19:31:25.198: ISAKMP:(83727):Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
*Nov  6 19:31:25.198: ISAKMP:(83727):Old State = IKE_P1_COMPLETE  New State = 
IKE_P1_COMPLETE

*Nov  6 19:31:25.238: ISAKMP (83727): received packet from 14.0.0.2 dport 500 sport 500 
fvrf (R) QM_IDLE
*Nov  6 19:31:25.238: ISAKMP: set new node -134314170 to QM_IDLE
*Nov  6 19:31:25.242: ISAKMP:(83727): processing HASH payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727): processing SA payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727):Checking IPSec proposal 1
*Nov  6 19:31:25.242: ISAKMP: transform 1, ESP_3DES
*Nov  6 19:31:25.242: ISAKMP:   attributes in transform:
*Nov  6 19:31:25.242: ISAKMP:      encaps is 1 (Tunnel)
*Nov  6 19:31:25.242: ISAKMP:      SA life type in seconds
*Nov  6 19:31:25.242: ISAKMP:      SA life duration (basic) of 3600
*Nov  6 19:31:25.242: ISAKMP:      SA life type in kilobytes
*Nov  6 19:31:25.242: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
*Nov  6 19:31:25.242: ISAKMP:      authenticator is HMAC-SHA
*Nov  6 19:31:25.242: ISAKMP:(83727):atts are acceptable.
*Nov  6 19:31:25.242: ISAKMP:(83727): processing NONCE payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727): processing ID payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727): processing ID payload. message ID = -134314170
*Nov  6 19:31:25.242: ISAKMP:(83727):QM Responder gets spi
```

```
*Nov  6 19:31:25.242: ISAKMP:(83727):Node -134314170, Input = IKE_MESG_FROM_PEER,
IKE_QM_EXCH
*Nov  6 19:31:25.242: ISAKMP:(83727):Old State = IKE_QM_READY  New State =
IKE_QM_SPI_STARVE
*Nov  6 19:31:25.242: ISAKMP:(83727): Creating IPSec SAs
*Nov  6 19:31:25.246:         inbound SA from 14.0.0.2 to 15.0.0.2 (f/i)  1/714
        (proxy 12.0.0.2 to 13.0.0.2)
*Nov  6 19:31:25.246:         has spi 0x917AD879 and conn_id 0
*Nov  6 19:31:25.246:         lifetime of 3600 seconds
*Nov  6 19:31:25.246:         lifetime of 4608000 kilobytes
*Nov  6 19:31:25.246:         outbound SA from 15.0.0.2 to 14.0.0.2 (f/i) 1/714
        (proxy 13.0.0.2 to 12.0.0.2)
*Nov  6 19:31:25.246:         has spi  0xC54A5A05 and conn_id 0
*Nov  6 19:31:25.246:         lifetime of 3600 seconds
*Nov  6 19:31:25.246:         lifetime of 4608000 kilobytes
*Nov  6 19:31:25.246:  ISAKMP: Failed to find peer index node to update peer_info_list
*Nov  6 19:31:25.250: ISAKMP:(83727): sending packet to 14.0.0.2 my_port 500 peer_port 500
(R) QM_IDLE
*Nov  6 19:31:25.250: ISAKMP:(83727):Node -134314170, Input = IKE_MESG_INTERNAL,
IKE_GOT_SPI
*Nov  6 19:31:25.250: ISAKMP:(83727):Old State = IKE_QM_SPI_STARVE  New State =
IKE_QM_R_QM2
*Nov  6 19:31:25.270: ISAKMP (83727): received packet from 14.0.0.2 dport 500 sport 500
fvrf (R) QM_IDLE
*Nov  6 19:31:25.274: ISAKMP:(83727):deleting node -134314170 error FALSE reason "QM done
(await)"
*Nov  6 19:31:25.274: ISAKMP:(83727):Node -134314170, Input = IKE_MESG_FROM_PEER,
IKE_QM_EXCH
*Nov  6 19:31:25.274: ISAKMP:(83727):Old State = IKE_QM_R_QM2  New State =
IKE_QM_PHASE2_COMPLETE
*Nov  6 19:32:15.282: ISAKMP:(83727):purging node -134314170
```

Command Output for show crypto isakmp sa [detail] for the Responder

```
Router# show crypto isakmp sa vrf vrf2
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
15.0.0.2        14.0.0.2        QM_IDLE          83727 ACTIVE prof2

IPv6 Crypto ISAKMP SA

Router# show crypto isakmp sa detail vrf vrf2
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id  Local           Remote          I-VRF    Status Encr Hash Auth DH Lifetime Cap.

83727 15.0.0.2        14.0.0.2        vrf2     ACTIVE 3des md5  rsig 1  23:59:15
       Engine-id:Conn-id =  :15727

IPv6 Crypto ISAKMP SA
```

Assigning the Group Name to the Peer

To associate a group name with an ISAKMP profile that will be assigned to a peer, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto isakmp profile`** *`profile-name`*	Defines an ISAKMP profile and enters ISAKMP profile configuration mode • *profile-name*—Name of the user profile.
Step 2	`Router (conf-isa-prof)#` **`client configuration group`** *`group-name`*	Accepts the name of a group that will be assigned to a peer when the peer is assigned this crypto ISAKMP profile. • *group-name*—Name of the group to be associated with the peer.

Verifying the Group Name to Peer Assignation Configuration

To verify that a group has been assigned to a peer, enter the **debug crypto isakmp** command.

The **debug crypto isakmp** command displays messages about IKE events.

The following **debug crypto isakmp** output shows that the peer has been matched to the ISAKMP profile named "certpro" and that it has been assigned a group named "new_group."

Initiator Configuration

```
crypto isakmp profile certpro
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
   client configuration group new_group
! The statement on the above line will assign the group "new_group" to any peer that
matches the ISAKMP profile "certpro."
   initiate mode aggressive
```

Command Output for debug crypto isakmp for the Responder

```
Router# debug crypto isakmp
6d23h: ISAKMP (0:268435461): received packet from 192.0.0.2 dport 500 sport 500 Global (R)
MM_KEY_EXCH
6d23h: ISAKMP: Main Mode packet contents (flags 1, len 892):
6d23h:           ID payload
6d23h:             FQDN <Router1.cisco.com> port 500 protocol 17
6d23h:           CERT payload
6d23h:           SIG payload
6d23h:           KEEPALIVE payload
6d23h:           NOTIFY payload
6d23h: ISAKMP:(0:5:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
6d23h: ISAKMP:(0:5:HW:2):Old State = IKE_R_MM4  New State = IKE_R_MM5
6d23h: ISAKMP:(0:5:HW:2): processing ID payload. message ID = 0
6d23h: ISAKMP (0:268435461): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : Router1.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
6d23h: ISAKMP:(0:5:HW:2):: peer matches *none* of the profiles
6d23h: ISAKMP:(0:5:HW:2): processing CERT payload. message ID = 0
6d23h: ISAKMP:(0:5:HW:2): processing a CT_X509_SIGNATURE cert
6d23h: ISAKMP:(0:5:HW:2): peer's pubkey isn't cached
6d23h: ISAKMP:(0:5:HW:2): OU = green
6d23h: ISAKMP:(0:5:HW:2): certificate map matches certpro profile
6d23h: ISAKMP:(0:5:HW:2): Trying to re-validate CERT using new profile
6d23h: ISAKMP:(0:5:HW:2): Creating CERT validation list: 2315, LaBcA,
6d23h: ISAKMP:(0:5:HW:2): CERT validity confirmed.
6d23h: ISAKMP:(0:5:HW:2):Profile has no keyring, aborting key search
6d23h: ISAKMP:(0:5:HW:2): Profile certpro assigned peer the group named new_group
```

For complete configuration information for certificate to ISAKMP profile mapping, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t8/feature/guide/gt_isakp.html

For certificate to ISAKMP profile mapping configuration examples, see the "Certificate to ISAKMP Profile Mapping Configuration Examples" section on page 24-23.

Configuring an Encrypted Preshared Key

The Encrypted Preshared Key feature allows you to securely store plain text passwords in type 6 (encrypted) format in NVRAM.

Encrypted Preshared Key Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring an encrypted preshared key:

- Old ROM monitors (ROMMONs) and boot images cannot recognize the new type 6 passwords. If you boot from an old ROMMON, you can expect errors.
- If the password (master key) is changed, or reencrypted, using the **key config-key password-encryption** command, the list registry passes the old key and the new key to the application modules that are using type 6 encryption.
- If the master key that was configured using the **key config-key password-encryption** command is deleted from the system, a warning is printed (and a confirm prompt is issued) that states that all type 6 passwords will become useless. As a security measure, after the passwords have been encrypted, they will never be decrypted in the Cisco IOS software. However, passwords can be reencrypted.

Caution If the password configured using the **key config-key password-encryption** command is lost, it cannot be recovered. The password should be stored in a safe location.

- If you later unconfigure password encryption using the **no password encryption aes** command, all existing type 6 passwords are left unchanged, and as long as the password (master key) that was configured using the **key config-key password-encryption** command exists, the type 6 passwords will be decrypted as and when required by the application.
- Because no one can "read" the password (configured using the **key config-key password-encryption** command), there is no way that the password can be retrieved from the switch. Existing management stations cannot "know" what it is unless the stations are enhanced to include this key somewhere, in which case the password needs to be stored securely within the management system. If configurations are stored using TFTP, the configurations are not standalone,

meaning that they cannot be loaded onto a switch. Before or after the configurations are loaded onto a switch, the password must be manually added (using the **key config-key password-encryption** command). The password can be manually added to the stored configuration but is not recommended because adding the password manually allows anyone to decrypt all passwords in that configuration.

- If you enter or cut and paste cipher text that does not match the master key, or if there is no master key, the cipher text is accepted or saved, but the following alert message is printed:

```
ciphertext>[for username bar>] is incompatible with the configured master key
```

- If a new master key is configured, all the plain keys are encrypted and made type 6 keys. The existing type 6 keys are not encrypted. The existing type 6 keys are left as is.
- If the old master key is lost or unknown, you have the option of deleting the master key using the **no key config-key password-encryption** command. Deleting the master key using the **no key config-key password-encryption** command causes the existing encrypted passwords to remain encrypted in the switch configuration. The passwords will not be decrypted.

Configuring an Encrypted Preshared Key

To configure an encrypted preshared key, perform the following task beginning global configuration mode:

	Command	Purpose

Step 1	`Router(config)# key config-key password-encryption`	Stores a type 6 encryption key in private NVRAM.
		Note the following:
		• If you are entering the key interactively (using the **Enter** key) and an encrypted key already exists, you will be prompted for the following:
		`Old key, New key, and Confirm key`
		• If you are entering the key interactively but an encryption key is not present, you will be prompted for the following:
		`New key and Confirm key`
		• If you are removing a password that is already encrypted, you will see the following prompt:
		`WARNING: All type 6 encrypted keys will become unusable. Continue with master key deletion? [yes/no]:`

Step 2	`Router(config)# password-encryption aes`	Enables the encrypted preshared key.

Verifying the Encrypted Preshared Key Configuration

To verify that a new master key has been configured and that the keys have been encrypted with the new master key, enter the **password logging** command. The following is an example of its output:

```
Router(config)# password logging
Router(config)# key config-key password-encrypt

New key:
Confirm key:
Router(config)#
01:40:57: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master keypas

Router(config)# key config-key password-encrypt
Old key:
New key:
Confirm key:
Router (config)#
01:42:11: TYPE6_PASS: Master key change heralded, re-encrypting the keys
with the new master key
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
```

For complete configuration information for the Encrypted Preshared Key feature, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t2/feature/guide/gt_epsk.html

For an encrypted preshared key configuration example, see the "Encrypted Preshared Key Configuration Example" section on page 24-23.

Configuring Call Admission Control for IKE

Call Admission Control (CAC) for IKE allows you to limit the number of simultaneous IKE security associations (SAs) that a switch can establish.

Note Call Admission Control is supported in Cisco IOS Release 12.2(33)SXH and later releases.

There are two ways to limit the number of IKE SAs that a switch can establish to or from another switch:

- Configure an absolute IKE SA limit by entering the **crypto call admission limit** command.

  When an IKE SA limit is defined, the switch no longer accepts or initiates new IKE SA requests when this value has been reached as follows: When there is a new SA request from a peer switch, IKE determines if the number of active IKE SAs plus the number of SAs being negotiated meets or exceeds the configured SA limit. If the number is greater than or equal to the limit, the new SA request is rejected and a syslog is generated. This log contains the source destination IP address of the SA request.

- Configure a system resource limit by entering the **call admission limit** command.

  When a system resource limit is defined, the switch no longer accepts or initiates new IKE SA requests when the specified level of system resources is being used as follows: Call Admission Control (CAC) polls a global resource monitor so that IKE knows when the switch is running short of CPU cycles or memory buffers. You can configure a resource limit, from 1 to 100000, that represents a level of system resources. When that level of the system resources is being used, IKE no longer accepts or initiates new IKE SA requests.

CAC is applied to new SAs (that is, when an SA does not already exist between the peers) and rekeying SAs. Every effort is made to preserve existing SAs. Only new SA requests will ever be denied due to a lack of system resources or because the configured IKE SA limit has been reached.

Configuring the IKE Security Association Limit

To configure an IKE Security Association limit, perform the following steps beginning in global configuration mode. When an IKE SA limit is defined, the switch no longer accepts or initiates new IKE SA requests when the limit has been reached:

	Command	Purpose
Step 1	`Router(config)# crypto call admission limit {ike {sa number | in-negotiation-sa number}}`	Specifies the maximum number of IKE SAs that the switch can establish before IKE no longer accepts or initiates new SA requests. • **sa** *number*—Number of active IKE SAs allowed on the switch. The range is 0 to 99999. • **in-negotiation-sa** *number*—Number of in-negotiation IKE SAs allowed on the switch. The range is 10 to 99999. **Note** An ISAKMP connection needs to be built in two directions. If you have 500 spokes in your network, you should set this value at a minimum of 1000 (500 x 2).
Step 2	`Router(config)# exit`	Returns to privileged EXEC mode.

Configuring a System Resource Limit

To configure a system resource limit, perform the following steps beginning in global configuration mode. When an IKE SA limit is defined, the switch no longer accepts or initiates new IKE SA requests when the specified level of system resources is being used.

	Command	Purpose
Step 1	`Router(config)# call admission limit charge`	Instructs IKE to stop initiating or accepting new SA requests (that is, calls for CAC) when the specified level of system resources is being used. • *charge*—Level of the system resources that, when used, causes IKE to stop accepting new SA requests. Valid values are 1 to 100000.
Step 2	`Router(config)# exit`	Returns to privileged EXEC mode.

Clearing Call Admission Statistics

To clear the Call Admission Control counters that track the number of accepted and rejected Internet Key Exchange (IKE) requests, use the **clear crypto call admission statistics** command in global configuration mode:

```
Router(config)# clear crypto call admission statistics
```

Verifying the Call Admission Control for IKE Configuration

To verify that Call Admission Control has been configured, enter the **show call admission statistics** and the **show crypto call admission statistics** commands.

The **show call admission statistics** command monitors the global CAC configuration parameters and the behavior of CAC.

```
Router# show call admission statistics
Total Call admission charges: 0, limit 25
Total calls rejected 12, accepted 51
Load metric: charge 0, unscaled 0
```

The **show crypto call admission statistics** command monitors crypto CAC statistics.

```
Router# show crypto call admission statistics
-----------------------------------------------------------
               Crypto Call Admission Control Statistics
-----------------------------------------------------------
System Resource Limit: 0    Max IKE SAs 0
Total IKE SA Count:    0    active:     0   negotiating: 0
Incoming IKE Requests: 0    accepted:   0   rejected:    0
Outgoing IKE Requests: 0    accepted:   0   rejected:    0
Rejected IKE Requests: 0    rsrc low:   0   SA limit:    0
```

For more complete configuration information for Call Admission Control for IKE, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t8/feature/guide/gtcallik.html

For Call Admission Control for IKE configuration examples, see the "Call Admission Control for IKE Configuration Examples" section on page 24-24.

Configuring Dead Peer Detection

Dead Peer Detection (DPD), defined in RFC 3706, is a mechanism used to detect dead IPsec peers. IPsec is a peer-to-peer type of technology. It is possible that IP connectivity may be lost between peers due to routing problems, peer reloading, or some other situation. This lost connectivity can result in black holes where traffic is lost. DPD, based on a traffic-detection method, is one possible mechanism to remedy this situation.

Note The **periodic** option of the **crypto isakmp keepalive** command is only supported as of Cisco IOS Release 12.2(33)SXH; the **on-demand** option is supported in all releases.

DPD supports two options: on-demand or periodic. The on-demand approach is the default. With on-demand DPD, messages are sent on the basis of traffic patterns. For example, if a switch must send outbound traffic and the liveliness of the peer is questionable, the switch sends a DPD message to query the status of the peer. If a switch has no traffic to send, it never sends a DPD message. If a peer is dead, and the switch never has any traffic to send to the peer, the switch will not find out until the IKE or IPsec security association (SA) has to be rekeyed (the liveliness of the peer is unimportant if the switch is not trying to communicate with the peer). On the other hand, if the switch has traffic to send to the peer, and the peer does not respond, the switch will initiate a DPD message to determine the state of the peer.

With the periodic option, you can configure your switch so that DPD messages are forced at regular intervals. This forced approach results in earlier detection of dead peers. For example, if a switch has no traffic to send, a DPD message is still sent at regular intervals, and if a peer is dead, the switch does not have to wait until the IKE SA times out to find out.

DPD is configured using the **crypto isakmp keepalive** command. DPD and Cisco IOS keepalives function on the basis of a timer. If the timer is set for 10 seconds, the switch will send a hello message every 10 seconds (unless, of course, the switch receives a hello message from the peer). The benefit of Cisco IOS keepalives and periodic DPD is earlier detection of dead peers. However, Cisco IOS keepalives and periodic DPD rely on periodic messages that have to be sent with considerable frequency. The result of sending frequent messages is that the communicating peers must encrypt and decrypt more packets.

DPD and Cisco IOS keepalive features can be used in conjunction with multiple peers in the crypto map to allow for stateless failover. DPD allows the switch to detect a dead IKE peer, and when the switch detects the dead state, the switch deletes the IPsec and IKE SAs to the peer. If you configure multiple peers, the switch will switch over to the next listed peer for a stateless failover.

DPD Configuration Guidelines and Restrictions

When configuring DPD, follow these guidelines and restrictions:

- When the **crypto isakmp keepalive** command is configured, the Cisco IOS software negotiates the use of Cisco IOS keepalives or DPD, depending on which protocol the peer supports.
- If you do not configure the **periodic** option using the **crypto isakmp keepalive** command, the switch defaults to the **on-demand** approach.
- Before configuring periodic DPD, you should ensure that your IKE peer supports DPD. Implementations that support DPD include the Cisco VPN 3000 concentrator, Cisco PIX Firewall, Cisco VPN Client, and Cisco IOS software in all modes of operation—site-to-site, Easy VPN remote, and Easy VPN server.
- Using periodic DPD potentially allows the switch to detect an unresponsive IKE peer with better response time when compared to on-demand DPD. However, use of periodic DPD incurs extra overhead. When communicating to large numbers of IKE peers, you should consider using on-demand DPD instead.
- When you configure DPD using the **crypto isakmp keepalive** *seconds* command, the *seconds* argument specifies the interval between DPD messages. In the case of on-demand DPD, the actual interval may be up to twice the configured value.

Configuring a Dead Peer Detection Message

To allow the switch to send DPD messages to the peer, perform the following task:

Command	Purpose
`Router# crypto isakmp keepalive seconds [retries] [periodic | on-demand]`	Converts Switch 1 to standalone mode. - *seconds*—Specifies the number of seconds between DPD messages; the range is from 10 to 3600 seconds. - *retries*—(Optional) Specifies the number of seconds between DPD retries if the DPD message fails; the range is from 2 to 60 seconds. If unspecified, the default is 2 seconds. - **periodic**—(Optional) Specifies that the DPD messages are sent at regular intervals. - **on-demand**—(Optional) Specifies that DPD retries are sent on demand. This is the default behavior.

Note Because the **on-demand** option is the default, the **on-demand** keyword does not appear in configuration output.

Verifying the DPD Configuration

To verify that DPD is enabled, use the **show crypto isakmp sa detail** command in global mode:

```
Router# show crypto isakmp sa detail
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption

C-id  Local           Remote          I-VRF    Encr Hash Auth DH Lifetime Cap.
273   11.0.0.2        11.0.0.1        ivrf21   3des sha  psk  2  01:59:35 D
       Connection-id:Engine-id =  273:2(hardware)
```

For more complete configuration information for Cisco IOS Dead Peer Detection (DPD) support, refer to the *Cisco IOS Security Command Reference, Release 12.3*.

For DPD configuration examples, see the "Dead Peer Detection Configuration Examples" section on page 24-24.

Understanding IPsec NAT Transparency

The IPsec NAT transparency feature introduces support for IP Security (IPsec) traffic to travel through Network Address Translation (NAT) or Port Address Translation (PAT) points in the network by addressing many known incompatibilities between NAT and IPsec.

Before the introduction of this feature, a standard IPsec virtual private network (VPN) tunnel would not work if there were one or more NAT or PAT points in the delivery path of the IPsec packet. This feature allows IPsec to operate through a NAT/PAT device.

For detailed information on NAT Transparency, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftipsnat.html

IPsec NAT Transparency Configuration Guidelines and Restrictions

When configuring IPsec NAT transparency, follow these guidelines and restrictions:

- For non-GRE over IPsec configurations, NAT transparency is supported in both tunnel and transport modes.
- For point-to-point GRE over IPsec configurations, NAT transparency is supported only in tunnel mode.
- For DMVPN configurations, NAT transparency is supported only in transport mode.

Configuring NAT Transparency

NAT transparency is a feature that is auto-detected by the IPsec VPN SPA. There are no configuration steps. If both VPN devices are NAT transparency-capable, NAT transparency is auto-detected and auto-negotiated.

Disabling NAT Transparency

You might want to disable NAT transparency if you already know that your network uses IPsec-awareness NAT (SPI-matching scheme). To disable NAT transparency, use the following command in global configuration mode:

```
Router(config)# no crypto ipsec nat-transparency udp-encapsulation
```

Configuring NAT Keepalives

By default, the NAT keepalive feature is disabled. To configure your switch to send NAT keepalive packets, enter the **crypto isakmp nat keepalive** command in global configuration mode:

```
Router(config)# crypto isakmp nat keepalive seconds
```

In this command, *seconds* specifies the number of seconds between keepalive packets; range is between 5 to 3,600 seconds.

For a NAT keepalive configuration example, see the "ISAKMP NAT Keepalive Configuration Example" section on page 24-24.

Verifying the NAT Configuration

To verify the NAT configuration, enter the **show crypto ipsec sa** command:

Note When you first enter the **show crypto ipsec sa** command, the packet counters may not show the correct values. Repeat the command to show the updated values.

```
Router# show crypto ipsec sa

interface:GigabitEthernet5/0/1
    Crypto map tag:testtag, local addr. 10.2.80.161

    local ident (addr/mask/prot/port):(10.2.80.161/255.255.255.255/0/0)
    remote ident (addr/mask/prot/port):(100.0.0.1/255.255.255.255/0/0)
    current_peer:100.0.0.1:4500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps:109, #pkts encrypt:109, #pkts digest 109
    #pkts decaps:109, #pkts decrypt:109, #pkts verify 109
    #pkts compressed:0, #pkts decompressed:0
    #pkts not compressed:0, #pkts compr. failed:0, #pkts decompress failed:0
    #send errors 90, #recv errors 0

    local crypto endpt.:10.2.80.161, remote crypto endpt.:100.0.0.1:4500
    path mtu 1500, media mtu 1500
    current outbound spi:23945537

    inbound esp sas:
    spi:0xF423E273(4095992435)
    transform:esp-des esp-sha-hmac ,
    in use settings ={Tunnel UDP-Encaps, }
    slot:0, conn id:200, flow_id:1, crypto map:testtag
    sa timing:remaining key lifetime (k/sec):(4607996/2546)
    IV size:8 bytes
    replay detection support:Y

    inbound ah sas:

    inbound pcp sas:

    outbound esp sas:
    spi:0x23945537(596923703)
    transform:esp-des esp-sha-hmac ,
    in use settings ={Tunnel UDP-Encaps, }
    slot:0, conn id:201, flow_id:2, crypto map:testtag
    sa timing:remaining key lifetime (k/sec):(4607998/2519)
    IV size:8 bytes
    replay detection support:Y

    outbound ah sas:

    outbound pcp sas:
```

For complete configuration information for Cisco IOS IPsec NAT transparency support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftipsnat.html

Configuration Examples

This section provides examples of the following configurations:

- Advanced Encryption Standard Configuration Example, page 24-22
- ISAKMP Keyrings Configuration Examples, page 24-22
- Certificate to ISAKMP Profile Mapping Configuration Examples, page 24-23
- Encrypted Preshared Key Configuration Example, page 24-23
- Call Admission Control for IKE Configuration Examples, page 24-24
- Dead Peer Detection Configuration Examples, page 24-24
- ISAKMP NAT Keepalive Configuration Example, page 24-24

Advanced Encryption Standard Configuration Example

The following example configures the Advanced Encryption Standard (AES) 256-bit key:

```
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
```

ISAKMP Keyrings Configuration Examples

The following examples show how to limit the scope of an Internet Security Association and Key Management Protocol (ISAKMP) profile or ISAKMP keyring configuration to a local termination address or interface:

- ISAKMP Profile Bound to a Local Interface Configuration Example, page 24-22
- ISAKMP Keyring Bound to a Local Interface Configuration Example, page 24-22
- ISAKMP Keyring Bound to a Local IP Address Configuration Example, page 24-23

ISAKMP Profile Bound to a Local Interface Configuration Example

The following example configures an ISAKMP profile bound to a local interface:

```
crypto isakmp profile prof1
   keyring key0
   match identity address 11.0.0.2 255.255.255.255
   local-address serial2/0
```

ISAKMP Keyring Bound to a Local Interface Configuration Example

The following example configures an ISAKMP keyring bound only to interface serial2/0:

```
crypto keyring key0
  local-address serial2/0
  pre-shared-key address 11.0.0.2 key 12345
```

ISAKMP Keyring Bound to a Local IP Address Configuration Example

The following example configures an ISAKMP keyring bound only to IP address 11.0.0.1:

```
crypto keyring key0
  local-address 11.0.0.1
  pre-shared-key address 11.0.0.2 key 12345
```

Certificate to ISAKMP Profile Mapping Configuration Examples

The following examples show how to configure Certificate to ISAKMP Profile Mapping:

- Certificates Mapped to the ISAKMP Profile on the Basis of Arbitrary Fields Configuration Example, page 24-23
- Group Name Assigned to a Peer Associated with an ISAKMP Profile Configuration Example, page 24-23

Certificates Mapped to the ISAKMP Profile on the Basis of Arbitrary Fields Configuration Example

The following example shows that whenever a certificate contains "ou = green," the ISAKMP profile "cert_pro" will be assigned to the peer:

```
crypto pki certificate map cert_map 10
 subject-name co ou = green
!
crypto isakmp identity dn
crypto isakmp profile cert_pro
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
```

Group Name Assigned to a Peer Associated with an ISAKMP Profile Configuration Example

The following example shows that the group "some_group" is to be associated with a peer that has been assigned an ISAKMP profile:

```
crypto isakmp profile id_profile
   ca trust-point 2315
   match identity host domain cisco.com

client configuration group some_group
```

Encrypted Preshared Key Configuration Example

The following example shows a configuration for which a type 6 preshared key has been encrypted:

```
Router(config)# password encryption aes
Router(config)# key config-key password-encrypt
New key:
Confirm key:
Router(config)#
0:46:40: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master key
Router(config)# exit
```

Call Admission Control for IKE Configuration Examples

The following examples show how to configure Call Admission Control (CAC) for IKE:

- IKE Security Association Limit Configuration Example, page 24-24
- System Resource Limit Configuration Example, page 24-24

IKE Security Association Limit Configuration Example

The following example shows how to specify that there can be a maximum of 25 SAs before IKE starts rejecting new SA requests:

```
Router(config)# crypto call admission limit ike sa 25
```

System Resource Limit Configuration Example

The following example shows how to specify that IKE should drop SA requests when a given level of system resources are being used:

```
Router(config)# call admission limit 50000
```

Dead Peer Detection Configuration Examples

The following examples show how to configure Dead Peer Detection (DPD):

- On-Demand DPD Configuration Example, page 24-24
- Periodic DPD Configuration Example, page 24-24

On-Demand DPD Configuration Example

The following example shows how to configure on-demand DPD messages. In this example, DPD messages will be sent every 60 seconds and every 5 seconds between retries if the peer does not respond:

```
Router(config)# crypto isakmp keepalive 60 5
```

Periodic DPD Configuration Example

The following example shows how to configure periodic DPD messages. In this example, DPD messages are to be sent at intervals of 10 seconds:

```
Router(config)# crypto isakmp keepalive 10 periodic
```

ISAKMP NAT Keepalive Configuration Example

The following example shows how to enable NAT keepalives to be sent every 20 seconds:

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key 1234 address 56.0.0.1
crypto isakmp nat keepalive 20
```

```
!
!
crypto ipsec transform-set t2 esp-des esp-sha-hmac
!
crypto map test2 10 ipsec-isakmp
 set peer 56.0.0.1
 set transform-set t2
 match address 101
```

CHAPTER 25

Configuring Enhanced IPsec Features Using the IPsec VPN SPA

This chapter provides information about configuring enhanced IPsec features using the IPsec VPN SPA on the Catalyst 6500 Series switch. It includes the following sections:

Note For detailed information on Cisco IOS IPsec cryptographic operations and policies, refer to the *Cisco IOS Security Configuration Guide, Release 12.2* and *Cisco IOS Security Command Reference, Release 12.2*.

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Tip To ensure a successful configuration of your VPN using the IPsec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Overview of Enhanced IPsec Features

IPsec is a framework of open standards developed by the Internet Engineering Task Force (IETF). It provides security for transmission of sensitive information over unprotected networks, such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers), such as Cisco routers.

This chapter describes the advanced IPsec features that can be used to improve scalability and performance of your IPsec VPN.

Configuring Advanced Encryption Standard in a Transform Set

The Advanced Encryption Standard (AES) is a privacy transform for IPsec and Internet Key Exchange (IKE) that has been developed to replace the Data Encryption Standard (DES). AES is designed to be more secure than DES. AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length. The algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.

To configure the AES encryption algorithm within a transform set, perform this task beginning in global configuration mode:

Command	Purpose

`Router(config)#` **`crypto ipsec transform-set`**
`transform-set-name`
`transform1``[`*`transform2`*`[`*`transform3`*`]]`
`...`
	Specifies a transform set and IPsec security profiles and algorithms.

transform-set-name specifies the name of the transform set.

transform1[*transform2*[*transform3*]] defines IPsec security protocols and algorithms. To configure AES, you must choose from the following AES Encapsulating Security Payload (ESP) encryption transforms:

- **esp-aes** specifies ESP with the 128-bit AES encryption algorithm.
- **esp-aes 192** specifies ESP with the 192-bit AES encryption algorithm.
- **esp-aes 256** specifies ESP with the 256-bit AES encryption algorithm.

For other accepted transform values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference.*

Verifying the AES Transform Set

To verify the configuration of the transform set, enter the **show crypto ipsec transform-set** command:

```
Router# show crypto ipsec transform-set

Transform set transform-1:{esp-256-aes esp-md5-hmac}
will negotiate = {Tunnel, }
```

For more complete configuration information about AES support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ft_aes.html

For an AES configuration example, see the "Advanced Encryption Standard Configuration Example" section on page 25-23.

Configuring Reverse Route Injection

Reverse Route Injection (RRI) provides the ability for static routes to be automatically inserted into the routing process for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created on the basis of the remote proxy network and mask, with the next hop to this network being the remote tunnel endpoint. By using the remote Virtual Private Network (VPN) router as the next hop, the traffic is forced through the crypto process to be encrypted.

After the static route is created on the VPN router, this information is propagated to upstream devices, allowing them to determine the appropriate VPN router to which to send returning traffic in order to maintain IPsec state flows. Being able to determine the appropriate VPN router is particularly useful if multiple VPN routers are used at a site to provide load balancing or failover or if the remote VPN devices are not accessible via a default route. Routes are created in either the global routing table or the appropriate virtual routing and forwarding (VRF) table.

RRI is applied on a per-crypto map basis, whether this is via a static crypto map or a dynamic crypto map template. For both dynamic and static maps, routes are created only at the time of IPsec SA creation. Routes are removed when the SAs are deleted. The **static** keyword can be added to the **reverse-route** command if routes are created on the basis of the content of the crypto ACLs that are permanently attached to the static crypto map.

RRI Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring RRI:

Note When RRI is enabled, do not make changes to the crypto configuration while VPN sessions are active. Enter the **clear crypto session** command before making changes.

- IP routing should be enabled and static routes should be redistributed if dynamic routing protocols are to be used to propagate RRI-generated static routes.
- You can specify an interface or address as the explicit next hop to the remote VPN device. This functionality allows the overriding of a default route to properly direct outgoing encrypted packets.
- You can add a route tag value to any routes that are created using RRI. This route tag allows redistribution of groups of routes using route maps, allowing you to be selective about which routes enter your global routing table.
- RRI can be configured on the same crypto map that is applied to multiple router interfaces.
- The **reverse-route remote-peer** [**static**] command creates two routes. One route is the standard remote proxy ID and the next hop is the remote VPN client tunnel address. The second route is the actual route to that remote tunnel endpoint and is used when a recursive lookup requires that the remote endpoint be reachable by the next hop. Creation of the second route for the actual next hop is important in the VRF case in which a default route must be overridden by a more explicit route.

  To reduce the number of routes created and support some platforms that do not readily facilitate route recursion, the **reverse-route** {*ip-address*} [**static**] keyword can be used to create one route only.

- For devices using an IPsec VPN SPA, reverse route specifies the next hop to be the interface, subinterface, or virtual LAN (VLAN) with the crypto map applied to it.

Configuring RRI Under a Static Crypto Map

To configure RRI under a static crypto map, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto map** *map-name seq-name* **ipsec-isakmp**	Creates or modifies a crypto map entry and enters crypto map configuration mode. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry.
Step 2	Router(config-crypto-map)# **reverse-route** [[**static**] | **tag** *tag-id* [**static**] | **remote-peer** [**static**] | **remote-peer** *ip-address* [**static**]]	Creates source proxy information for a crypto map entry. • **static**—(Optional) Creates permanent routes based on static ACLs. • **tag** *tag-id*—(Optional) Tag value that can be used as a match value for controlling redistribution via route maps. • **remote-peer** [**static**]—(Optional) Two routes are created, one for the remote endpoint and one for route recursion to the remote endpoint via the interface to which the crypto map is applied. The **static** keyword is optional. • **remote-peer** *ip-address* [**static**]—(Optional) One route is created to a remote proxy by way of a user-defined next hop. This next hop can be used to override a default route. The *ip-address* argument is required. The **static** keyword is optional.

Configuring RRI Under a Dynamic Crypto Map

To configure RRI under a dynamic crypto map, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`crypto dynamic-map`**` {`*`dynamic-map-name`*`} {`*`dynamic-seq-name`*`}`	Creates a dynamic crypto map entry and enters crypto map configuration mode. • *dynamic-map-name*—Name that identifies the map set. • *dynamic-seq-num*—Sequence number assigned to the crypto map entry.
Step 2	`Router(config-crypto-map)# `**`reverse-route`**` [`**`tag`**` `*`tag-id`*` | `**`remote-peer`**` | `**`remote-peer`**` `*`ip-address`*`]`	Creates source proxy information for a crypto map entry. • **tag** *tag-id*—(Optional) Tag value that can be used as a match value for controlling redistribution via route maps. • **remote-peer**—(Optional) Two routes are created, one for the remote endpoint and one for route recursion to the remote endpoint via the interface to which the crypto map is applied. • **remote-peer** *ip-address*—(Optional) One route is created to a remote proxy by way of a user-defined next hop. This next hop can be used to override a default route. The *ip-address* argument is required.

For more complete configuration information for RRI, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gt_rrie.html

For RRI configuration examples, see the "Reverse Route Injection Configuration Examples" section on page 25-23.

Configuring the IPsec Anti-Replay Window Size

Cisco IPsec authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. (Security association (SA) anti-replay is a security service in which the receiver can reject old or duplicate packets to protect itself against replay attacks.) The decryptor checks off the sequence numbers that it has seen before. The encryptor assigns sequence numbers in an increasing order. The decryptor remembers the value (X) of the highest sequence number that it has already seen. N is the window size of the decryptor. Any packet with a sequence number less than X minus N is discarded. Currently, N is set at 64.

Note The IPsec anti-replay window size feature is supported in Cisco IOS Release 12.2(18)SXF6 and later releases.

At times, the 64-packet window size is not sufficient. For example, Cisco quality of service (QoS) gives priority to high-priority packets, which could cause some low-priority packets to be discarded even though they are not replayed packets. The IPsec anti-replay window size feature allows you to expand the window size so that sequence number information can be kept for more than 64 packets.

Note A change in the anti-replay window size will not take effect until after the next rekeying.

Expanding the IPsec Anti-Replay Window Size Globally

To expand the IPsec anti-replay window globally so that it affects all SAs that are created (except for those that are specifically overridden on a per-crypto map basis), perform this task beginning in global configuration mode:

Command	Purpose
`Router(config)# crypto ipsec security-association replay window size [size]`	Expands the IPsec anti-replay window globally to the specified *size*. • *size*—(Optional) Size of the window. Values can be 64, 128, 256, 512, or 1024. This value becomes the default value.

Expanding the IPsec Anti-Replay Window at the Crypto Map Level

To expand the IPsec anti-replay window on a crypto map basis so that it affects those SAs that have been created using a specific crypto map or profile, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num ipsec-isakmp`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry.
Step 2	`Router(config-crypto-map)# crypto ipsec security-association replay window size [size]`	Controls the SAs that are created using the policy specified by a particular crypto map, dynamic crypto map, or crypto profile. • *size*—(Optional) Size of the window. Values can be 64, 128, 256, 512, or 1024. This value becomes the default value.

Verifying the IPsec Anti-Replay Window Size Configuration at the Crypto Map Level

To verify that IPsec anti-replay window size is enabled at a crypto map, enter the **show crypto map** command for that particular map. If anti-replay window size is enabled, the display will indicate that it is enabled and indicate the configured window size. If anti-replay window size is disabled, the results will indicate that also.

The following example indicates that IPsec anti-replay window size is enabled:

```
Router# show crypto map tag TESTMAP

Crypto Map "TESTMAP" 10 ipsec-isakmp
        WARNING: This crypto map is in an incomplete state!
                (missing peer or access-list definitions)
        No matching address list set.
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
        }
        Antireplay window size = 128
        Interfaces using crypto map TESTMAP:
```

For more complete configuration information for IPsec anti-replay window size, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gt_iarwe.html

For IPsec anti-replay window size configuration examples, see the "IPsec Anti-Replay Window Size Configuration Examples" section on page 25-24.

Note Anti-replay failures detected by the IPsec VPN SPA can be caused by reordering, requeueing, or fragmentation elsewhere in the network. As a defense against man-in-the-middle attacks, the IPsec VPN SPA will drop these packets. This is the expected behavior.

Disabling the IPsec Anti-Replay Checking

To disable the IPsec anti-replay checking, enter the **crypto ipsec security-association replay disable** command in global configuration mode as follows:

Command	Purpose
`Router(config)# crypto ipsec security-association replay disable`	Disables the IPsec anti-replay checking.

To disable the IPsec anti-replay checking on a particular crypto map, enter the **set security-association replay disable** command in crypto map configuration mode as follows:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num ipsec-isakmp`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry.
Step 2	`Router(config-crypto-map)# set security-association replay disable`	Disables IPsec anti-replay checking by a particular crypto map, dynamic crypto map, or crypto profile.

Configuring an IPsec Preferred Peer

The IP Security (IPsec) Preferred Peer feature allows you to control the circumstances by which multiple peers on a crypto map are tried in a failover scenario. If there is a default peer, the next time a connection is initiated, the connection is directed to the default peer instead of to the next peer in the peer list. If all connections to the current peer time out, the next time a connection is initiated, it is directed to the default peer.

Note The IPsec Preferred Peer feature is supported in Cisco IOS Release 12.2(33)SXH and later releases.

This feature includes the following capabilities:

- Default peer configuration

  If a connection timeout occurs, the connection to the current peer is closed. The **set peer** command allows you to configure the first peer as the default peer. If there is a default peer, the next time a connection is initiated, the connection is directed to the default peer instead of to the next peer in the peer list. If the default peer is unresponsive, the next peer in the peer list becomes the current peer and future connections through the crypto map try that peer.

  This capability is useful when traffic on a physical link stops due to the failure of a remote peer. DPD indicates that the remote peer is unavailable, but that peer remains the current peer.

  A default peer facilitates the failover to a preferred peer that was previously unavailable, but has returned to service. Users can give preference to certain peers in the event of a failover. This is useful if the original failure was due to a network connectivity problem rather than failure of the remote peer.

  To configure a default peer, see the "Configuring a Default Peer" section on page 25-10.

- IPsec idle timer with default peer configuration

  When a router running Cisco IOS software creates an IPsec security association (SA) for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the router could be prevented from creating new SAs with other peers.

IPsec SA idle timers increase the availability of resources by deleting SAs associated with idle peers. Because IPsec SA idle timers prevent the wasting of resources by idle peers, more resources are available to create new SAs when required. (If IPsec SA idle timers are not configured, only the global lifetimes for IPsec SAs are applied. SAs are maintained until the global timers expire, regardless of peer activity.)

When both an IPsec SA idle timer and a default peer are configured and all connections to the current peer time out, the next time a connection is initiated it is directed to the default peer configured in the **set peer** command. If a default peer is not configured and there is a connection timeout, the current peer remains the one that timed out.

This enhancement helps facilitate a failover to a preferred peer that was previously unavailable but is in service now.

To configure an IPsec idle timer, see the "Configuring the IPsec Idle Timer with a Default Peer" section on page 25-11.

IPsec Preferred Peer Configuration Guidelines and Restrictions

When configuring an IPsec preferred peer, follow these guidelines and restrictions:

- When configuring a default peer, follow these guidelines and restrictions:
  - Only one peer can be designated as the default peer in a crypto map.
  - The default peer must be the first peer in the peer list.

Note The default peer feature must be used in conjunction with Dead Peer Detection (DPD). It is most effective on a remote site running DPD in periodic mode. DPD detects the failure of a device quickly and resets the peer list so that the default peer is tried for the next attempted connection.

- When configuring IPsec idle timer usage with a default peer, follow these guidelines and restrictions:
  - The IPsec idle timer usage with a default peer feature works only on the crypto map for which it is configured. You cannot configure the capability globally for all crypto maps.
  - If there is a global idle timer, the crypto map idle timer value must be different from the global value; otherwise, the idle timer is not added to the crypto map.

Configuring a Default Peer

To configure a default peer, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*]	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—(Optional) Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry. • **dynamic** *dynamic-map-name*—(Optional) Specifies the name of the dynamic crypto map set that should be used as the policy template. • **discover**—(Optional) Enables peer discovery. By default, peer discovery is not enabled. • **profile** *profile-name*—(Optional) Name of the crypto profile being created.
Step 2	Router(config-crypto-map)# **set peer** {*host-name* [**dynamic**] [**default**] | *ip-address* [**default**]}	Specifies an IPsec peer in a crypto map entry. Ensures that the first peer specified is defined as the default peer. • *host-name*—Specifies the IPsec peer by its host name. This is the peer's host name concatenated with its domain name (for example, myhost.example.com). • **dynamic**—(Optional) The host name of the IPsec peer will be resolved via a domain name server (DNS) lookup right before the router establishes the IPsec tunnel. • **default**—(Optional) If there are multiple IPsec peers, designates that the first peer is the default peer. • *ip-address*—Specifies the IPsec peer by its IP address.
Step 3	Router(config-crypto-map)# **exit**	Exits crypto map configuration mode and returns to global configuration mode.

Configuring the IPsec Idle Timer with a Default Peer

To configure the IPsec idle timer with a default peer, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num [ipsec-isakmp] [dynamic dynamic-map-name] [discover] [profile profile-name]`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—(Optional) Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry. • **dynamic** *dynamic-map-name*—(Optional) Specifies the name of the dynamic crypto map set that should be used as the policy template. • **discover**—(Optional) Enables peer discovery. By default, peer discovery is not enabled. • **profile** *profile-name*—(Optional) Name of the crypto profile being created.
Step 2	`Router(config-crypto-map)# set security-association idle-time seconds [default]`	Specifies the maximum amount of time for which the current peer can be idle before the default peer is used. • *seconds*—Number of seconds for which the current peer can be idle before the default peer is used. Valid values are 60 to 86400. • **default**—(Optional) Specifies that the next connection is directed to the default peer.
Step 3	`Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.

For complete configuration information for IPsec preferred peer, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gt_ipspp.html

For IPsec preferred peer configuration examples, see the "IPsec Preferred Peer Configuration Examples" section on page 25-26.

Configuring IPsec Security Association Idle Timers

When a switch running Cisco IOS software creates an IPsec SA for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the switch could be prevented from creating new SAs with other peers. The IPsec security association idle timers feature introduces a configurable idle timer to monitor SAs for activity, allowing SAs for idle peers to be deleted. The idle timers can be configured either globally, on a per-crypto map basis, or through an ISAKMP profile. The benefits of this feature include the following:

- Increased availability of resources
- Improved scalability of Cisco IOS IPsec deployments

IPsec Security Association Idle Timer Configuration Guidelines

When configuring idle timers on a per-crypto map basis, follow these guidelines:

- The IPsec VPN SPA rounds up the CLI-configured interval to the nearest 10-minute interval. For example, if you configure 12 minutes for idle timeout, the IPsec VPN SPA uses a value of 20 minutes for idle timeout. If you configure 5 minutes, the IPsec VPN SPA uses a value of 10 minutes for idle timeout.
- Because of the way the IPsec VPN SPA does idle timeout detection, it can take anywhere between one to three (ten-minute) intervals for idle timeout detection. For example, if you configured 12 minutes for idle timeout, idle timeout could happen anywhere between 20 to 60 minutes.
- When the idle timer is configured globally, the idle timer configuration will be applied to all SAs.
- When the idle timer is configured for a crypto map, the idle timer configuration will be applied to all SAs under the specified crypto map.

Configuring the IPsec SA Idle Timer Globally

To configure the IPsec SA idle timer globally, enter the **crypto ipsec security-association idle-time** command in global configuration mode as follows:

Command	Purpose
`Router(config)#` **`crypto ipsec security-association idle-time`** *`seconds`*	Specifies the time, in *seconds*, that the idle timer will allow an inactive peer to maintain an SA. The range is from 60 to 86400 seconds.

Configuring the IPsec SA Idle Timer per Crypto Map

To configure the IPsec SA idle timer for a specified crypto map, use the **set security-association idle-time** command within a crypto map configuration:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-number ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations.
Step 2	`Router(config-crypto-map)# set security-association idle-time seconds`	Specifies the time, in *seconds*, that the idle timer will allow an inactive peer to maintain an SA. The range is from 60 to 86400 seconds.

For detailed information on configuring IPsec SA idle timers, refer to the following Cisco IOS documentation:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ftsaidle.html

For IPsec SA idle timer configuration examples, see the "IPsec Security Association Idle Timer Configuration Examples" section on page 25-27.

Configuring Distinguished Name-Based Crypto Maps

The distinguished name-based crypto maps feature allows you to configure the switch to restrict access to selected encrypted interfaces for those peers with specific certificates, especially certificates with particular distinguished names (DNs).

Previously, if the switch accepted a certificate or a shared secret from the encrypting peer, Cisco IOS did not have a method of preventing the peer from communicating with any encrypted interface other than the restrictions on the IP address of the encrypting peer. This feature allows you to configure which crypto maps are usable to a peer based on the DN that a peer used to authenticate itself, which enables you to control which encrypted interfaces a peer with a specified DN can access. You can configure a DN-based crypto map that can be used only by peers that have been authenticated by a DN or one that can be used only by peers that have been authenticated by a hostname.

Distinguished Name-Based Crypto Map Configuration Guidelines and Restrictions

When configuring a distinguished name-based crypto map, follow these guidelines and restrictions:

- If you restrict access to a large number of DNs, we recommend that you specify a few number of crypto maps referring to large identity sections instead of specifying a large number of crypto maps referring to small identity sections.

To configure a DN-based crypto map that can be used only by peers that have been authenticated by a DN, or one that can be used only by peers that have been authenticated by a hostname, perform this task beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)#` **`crypto isakmp policy`** *`priority`*
	`...`
	`Router(config-isakmp)#` **`exit`**
		Defines an ISAKMP policy and enters ISAKMP policy configuration mode.
		• *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
		Creates an ISAKMP policy at each peer.
		For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.

Step 2	`Router(config)#` **`crypto map`** *`map-name seq-number`* **`ipsec-isakmp`**	Creates or modifies a crypto map entry and enters the crypto map configuration mode. • *map-name*—Name that identifies the crypto map set. • *seq-number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations.

	Command	Purpose
Step 3	Router(config-crypto-map)# **set identity** *name* ... Router(config-crypto-map)# **exit**	Applies the identity to the crypto map. • *name*—Identity of the switch, which is associated with the given list of DNs. When this command is applied, only the hosts that match a configuration listed within the identity name can use the specified crypto map. **Note** If the **set identity** command does not appear within the crypto map, the encrypted connection does not have any restrictions other than the IP address of the encrypting peer. Specify any other policy values appropriate to your configuration. For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.
Step 4	Router(config)# **crypto identity** *name*	Configures the identity of a switch with the given list of DNs in the certificate of the switch and enters crypto identity configuration mode. • *name*—The name value specified in Step 3.
Step 5	Router(crypto-identity)# **dn** *name*=**string** [,*name*=**string**]| **fqdn** *name*	Associates the identity of the switch with either a DN or hostname (FQDN) to restrict access to peers with specific certificates. • *name*=**string**—The DN in the certificate of the switch. Optionally, you can associate more than one DN. • **fqdn** *name*—The hostname that the peer used to authenticate itself (FQDN) or the DN in the certificate of the switch. The identity of the peer must match the identity in the exchanged certificate.

For complete configuration information for distinguished name-based crypto maps, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t4/feature/guide/ftdnacl.html

For a distinguished name-based crypto map configuration example, see the "Distinguished Name-Based Crypto Maps Configuration Example" section on page 25-27.

Configuring QoS for VPN

Typical applications of quality of service (QoS) for VPN are the use of traffic policing to prevent a hub from overwhelming a lower-capacity spoke, and the prioritization over VPN of delay-sensitive traffic such as voice over IP (VoIP). QoS features for VPN traffic are provided both by the module (IPsec VPN SPA and SSC-400 carrier card) and by the platform (Catalyst 6500 Series switch). The module provides a dual-priority queue for module traffic. In Cisco IOS Release 12.2(33)SXI and later releases, the Catalyst 6500 Series switch provides data classification and either remarking or policing of traffic on the tunnel interface.

To activate the QoS capabilities of the module, carrier, and platform, you must enable QoS globally by entering the **mls qos** command.

When QoS is disabled globally, the system behavior is as follows:

- All QoS fields are left intact in packets.
- Packets flow through only one queue in the carrier card.

When QoS is enabled globally, the system behavior is as follows:

- The default state of all ports and VLANs is the untrusted state, causing ports to clear the QoS fields in all traffic to zero unless a QoS policy is configured on the the port.
- Packets flow through two queues in the carrier card. Packets with a CoS value of 5 will use the higher priority queue, while all other packets will use the lower priority queue.

Before configuring QoS for VPN, see the additional information provided in the following URLs:

Configuring QoS on the Catalyst 6500 Series switch:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/qos.html

Configuring QoS Features on a SIP:

http://www.cisco.com/en/US/docs/interfaces_modules/shared_port_adapters/configuration/6500series/76cfgsip.html#Configuring_QoS_Features_on_a_SIP

Configuring QoS on the FlexWAN Modules:

http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexqos.html

QoS Policing on the Catalyst 6500 Series switch:

http://www.cisco.com/en/US/products/hw/switches/ps700/products_tech_note09186a00801c8c4b.shtml

QoS Output Scheduling on the Catalyst 6500 Series switch:

http://www.cisco.com/en/US/products/hw/switches/ps700/products_tech_note09186a008015bf98.shtml

QoS Troubleshooting:

http://www.cisco.com/en/US/products/hw/switches/ps708/products_tech_note09186a008074d6b1.shtml

Using the Module QoS Features of the IPsec VPN SPA

The IPsec VPN SPA implements a two-level, strict-priority QoS. The Cisco 7600 SSC-400 and the IPsec VPN SPA together implement two queues for each direction, inbound and outbound. Packets are dequeued in a two-to-one ratio, meaning that two packets are dequeued from the high-priority low-latency queue (LLQ) before one packet is dequeued from the low-priority queue. Packets are enqueued based on your priority-queue configuration settings. To take advantage of the IPsec VPN

SPA’s QoS capability, you must use standard QoS commands to ensure that the class of service (CoS) of packets is marked on ingress. You must configure the CoS map for the inside and outside ports and you must also enable QoS globally for the IPsec VPN SPA to acknowledge the CoS mapping.

Module QoS Configuration Guidelines and Restrictions

When configuring QoS settings for an IPsec VPN SPA, follow these guidelines and note these restrictions:

- In releases before Cisco IOS Release 12.2(33)SXI, service policies should not be applied on GRE and VTI tunnel interfaces.
- Packets are enqueued based on the **mls qos** command and the priority-queue configuration settings as follows:
  - When the **mls qos** command is not configured, all data packets are enqueued into the high-priority queue.
  - When the **mls qos** command is configured and no explicit priority-queue configuration is present on the IPsec VPN SPA Ethernet interfaces, only packets with a CoS value of 5 are enqueued into the high-priority queue; all other packets are enqueued into the low-priority queue.
  - When the **mls qos** command is configured and priority-queue configuration is present on the IPsec VPN SPA Ethernet interfaces, traffic is enqueued based on the priority-queue configuration.
- A maximum of three CoS map values can be sent to the high-priority queue. Because the CoS value of 5 is preconfigured as high-priority, you can choose only two other values for high-priority queueing.

Note Do not configure more than three CoS map values because any additional values will overwrite previously configured values. If you overwrite the CoS value of 5, the system will restore it, overwriting one of your other configured values. To restore an overwritten CoS map value, you must first delete the new value and then reconfigure the earlier value.

- When the **mls qos** command is configured, you must also configure the **mls qos trust** command on the IPsec VPN SPA Ethernet interfaces, as in the following example:

```
Interface GigabitEthernet4/0/1
 mls qos trust cos
 priority-queue cos-map 1 0 1 5
!
Interface GigabitEthernet4/0/2
 mls qos trust cos
 priority-queue cos-map 1 0 1 5
```

  In this example, the CoS values of 0, 1, and 5 are sent to the high-priority queue.

- In a blade failover group, both IPsec VPN SPAs must have matching platform QoS configurations.
- If the **mls qos trust** command is not configured, the QoS fields in all traffic will be cleared to the default level. If the **mls qos trust** command is configured, the QoS fields will be preserved.

For a configuration example of module QoS, see the “Module QoS Configuration Example” section on page 25-28.

Using the Platform QoS Features of the Switch

With Cisco IOS Release 12.2(33)SXI and later releases, the Catalyst 6500 Series switch allows data classification and either remarking or policing of packets on the tunnel interface.

Platform QoS configuration uses the Cisco Modular QoS CLI (MQC) framework. You can define traffic classes, associate policies and actions to each traffic class, and attach these policies to interfaces by following these steps:

Step 1 Define traffic classes using **match** statements with the **class-map** command.

Step 2 Configure policies using the defined traffic classes with the **policy-map** command.

Step 3 Attach defined policies to an interface with the **service-policy** command.

For more information on configuring QoS in the Catalyst 6500 Series switch, see the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/qos.html

Remarking of Packets

Remarking is specified by using a **set** command within a policy map. Platform QoS for VPN is capable of remarking the priority settings of original IP, GRE, and IPsec headers of tunnel traffic, depending on the tunnel type. For remarking of packets, the port trust settings must be as follows:

- If matching is based on ToS bits of the incoming packets, the LAN interface must be configured as a trusted port, using the **mls qos trust** command.
- Depending on the VPN mode and the desired behavior, the inside interface of the IPsec VPN SPA must be configured as VLAN-based, using the **mls qos vlan-based** command, or as a trusted port, using the **mls qos trust** command.

The following sections describe remarking behavior in different modes:

Remarking of IPsec Packets in Crypto-Connect Mode

The outside interface of the IPsec VPN SPA must be configured as a trusted port, using the **mls qos trust** command.

If the inside interface of the IPsec VPN SPA is configured as VLAN-based, using the **mls qos vlan-based** command, the remarking behavior will be as follows:

- Apply the service policy to the interface VLAN.
- For outbound traffic, remarking is performed on the original IP header and copied to the IPsec header.
- For inbound traffic, remarking is performed on the original IP header.
- Traffic matching the service policy will be remarked. Traffic not matching the service policy will be set to a priority of 0.

If the inside interface of the IPsec VPN SPA is configured as as a trusted port, using the **mls qos trust** command, the remarking behavior will be as follows:

- Apply the service policy to the interface VLAN.
- Remarking is supported only on outbound traffic.

- Remarking is performed on the original IP header and copied to the IPsec header.
- Traffic matching the service policy will be remarked.

Remarking of GRE Packets in Crypto-Connect Mode

The outside interface of the VSPA must be configured as a trusted port, using the **mls qos trust** command.

The inside interface of the IPsec VPN SPA is configured as VLAN-based, using the **mls qos vlan-based** command. The remarking behavior will be as follows:

- Remarking is supported for both inbound and outbound traffic.
- If the GRE tunnel is taken over by the IPsec VPN SPA:
  - Apply the service policy to the tunnel interface. Any service policy on the interface VLAN will be ignored.
  - Remarking is performed on the original IP header, the GRE header, and the IPsec header.
- If the GRE tunnel is not taken over by the IPsec VPN SPA:
  - Apply the service policy to the tunnel interface or to the interface VLAN.
  - Remarking is performed on the GRE header and the IPsec header, and not on the original IP header (inner header).
- Traffic matching the service policy will be remarked.

Remarking of GRE Packets with Tunnel Protection in VRF Mode

The outside interface of the IPsec VPN SPA must be configured as a trusted port, using the **mls qos trust** command. After egress from the IPsec VPN SPA, an encrypted packet cannot be remarked.

The inside interface of the IPsec VPN SPA is configured as VLAN-based, using the **mls qos vlan-based** command. The remarking behavior will be as follows:

- Remarking is supported for both inbound and outbound traffic.
- If the GRE tunnel is taken over by the IPsec VPN SPA:
  - Apply the service policy to the tunnel interface.
  - Remarking is performed on the original IP header, the GRE header, and the IPsec header.
- If the GRE tunnel is not taken over by the IPsec VPN SPA:
  - Apply the service policy to the tunnel interface or to the interface VLAN.
  - Remarking is performed on the GRE header and the IPsec header, and not on the original IP header (inner header).
- Traffic matching the service policy will be remarked.

Remarking of VTI Packets in VRF Mode

The outside interface of the IPsec VPN SPA must be configured as a trusted port, using the **mls qos trust** command. After egress from the IPsec VPN SPA, an encrypted packet cannot be remarked.

The inside interface of the IPsec VPN SPA is configured as VLAN-based, using the **mls qos vlan-based** command. The remarking behavior will be as follows:

- Apply the service policy to the tunnel interface.
- Remarking is supported for both inbound and outbound traffic.

- Remarking is performed on the original IP header and the IPsec header.
- Traffic matching the service policy will be remarked.

Policing of Packets

Policing enforces a maximum packet rate by dropping excess packets. Policing can limit the rate that packets are passed to and from the IPsec VPN SPA.

For more information on configuring policing in the Catalyst 6500 Series switch, see the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/qos.html

Platform QoS Guidelines and Restrictions

When configuring platform QoS for VPN, follow these guidelines and note these restrictions:

- To enable QoS, you must apply the **mls qos** command globally.
- Platform QoS policies can be applied to the tunnel interface in any of these situations:
  - GRE in crypto-connect mode, whether or not GRE is taken over by the IPsec VPN SPA
  - GRE with tunnel protection, whether or not GRE is taken over by the IPsec VPN SPA
  - Static VTI
  - mGRE
- The **match all, match not,** and **match cos** classification criteria are not supported for platform QoS on the tunnel interface.
- In the **police** command, the **exceed-action set** options are not supported for platform QoS on the tunnel interface, and cannot be configured for remarking or policing.
- Platform QoS policies do not apply to packets generated by the route processor or destined for the route processor.
- Supports 1023 policers or remarkers.
- When applying a service policy during configuration, the policy does not take effect until after you exit the interface configuration mode.
- If you apply and then remove a service policy, some packets will be remarked to other priorities during the transition. (CSCso84671)
- In some cases, upon removal of a service policy from a tunnel, the tunnel continues to remark outbound traffic. (CSCsq99617)
- When configuring for blade-to-blade failover, you must enter identical QoS configurations on the inside interfaces of both IPsec VPN SPAs.

For a configuration example of platform QoS, see the "Platform QoS Configuration Example" section on page 25-28.

Configuring Platform ACLs for VPN

With Cisco IOS Release 12.2(33)SXI and later releases, you can apply access control lists (ACLs) to VPN tunnel interfaces.

Platform ACL Configuration Guidelines and Restrictions

When configuring platform ACLs for an IPsec VPN SPA, follow these guidelines and note these restrictions:

- ACLs can be applied to the tunnel interface in any of these situations:
  - GRE in crypto-connect mode, whether or not GRE is taken over by the IPsec VPN SPA
  - GRE with tunnel protection, whether or not GRE is taken over by the IPsec VPN SPA
  - Static VTI
  - DMVPN, in either crypto-connect or VRF mode
- Permit and deny ACLs can be applied to tunnel interfaces in either the inbound or outbound direction.
- In crypto-connect mode with GRE, when GRE is not taken over by the IPsec VPN SPA, apply the ACL to the interface VLAN to filter GRE-encapsulated packets, or to the tunnel interface to filter clear IP packets.
- In crypto-connect mode with GRE, when GRE is taken over by the IPsec VPN SPA, ACLs on the interface VLAN are not supported. Apply the ACL to the tunnel interface to filter clear IP packets.
- ACLs on tunnels are supported in blade-to-blade failover.
- ACLs will be applied to transit packets, but will not be applied to packets generated by the switch.

For an ACL configuration example, see the "Platform ACL Configuration Example" section on page 25-30.

Configuring Sequenced Crypto ACLs

Access control lists (ACLs) are made up of access control entries (ACEs). With sequenced ACLs, ACEs can be entered with a sequence number in front of the ACE and the ACEs are then processed by sequence number. Additionally, ACEs can be deleted one at a time by using the sequence number in the front of the ACE that you want to delete. The sequence numbers do not appear in the configuration but they can be displayed using the **show access-list** command.

Note If an ACE is removed or modified, the ACL is reconfigured on the IPsec VPN SPA, which might result in tearing down existing sessions.

Configuring Deny Policy Enhancements for Crypto ACLs

Specifying a deny address range in an ACL results in "jump" behavior. When a denied address range is hit, it forces the search to "jump" to the beginning of the ACL associated with the next sequence in a crypto map and continue the search. If you want to pass clear traffic on these addresses, you must insert a deny address range for each sequence in a crypto map. In turn, each permit list of addresses inherits all the deny address ranges specified in the ACL. A deny address range causes the software to do a subtraction of the deny address range from a permit list, and creates multiple permit address ranges that need to be programmed in hardware. This behavior can cause repeated address ranges to be programmed in the hardware for a single deny address range, resulting in multiple permit address ranges in a single ACL. To avoid this problem, use the **crypto ipsec ipv4-deny** {**jump** | **clear** | **drop**} command set as follows:

- The **jump** keyword results in the standard "jump" behavior.
- The **clear** keyword allows a deny address range to be programmed in hardware. The deny addresses are then filtered out for encryption and decryption. If the VPN mode is crypto-connect, when a deny address is hit, the search is stopped and traffic is allowed to pass in the clear (unencrypted) state. If the VPN mode is VRF, the deny address matching traffic is dropped.
- The **drop** keyword causes traffic to be dropped when a deny address is hit.

The **clear** and **drop** keywords can be used to prevent repeated address ranges from being programmed in the hardware, resulting in more efficient TCAM space utilization.

Deny Policy Enhancements for Crypto ACLs Configuration Guidelines and Restrictions

When configuring the deny policy enhancements, follow these guidelines and restrictions:

- The **crypto ipsec ipv4-deny** {**jump** | **clear** | **drop**} command is a global command that is applied to a single IPsec VPN SPA. The specified keyword (**jump**, **clear**, or **drop**) is propagated to the ACE software of the IPsec VPN SPA. The default behavior is **jump**.
- When the **clear** keyword is used with VRF mode, deny address traffic is dropped rather than passed in the clear state. VRF mode does not pass traffic in the clear state.
- If you apply the specified keyword (**jump**, **clear**, or **drop**) when crypto maps are already configured on the IPsec VPN SPA, all existing IPsec sessions are temporarily removed and restarted, which impacts traffic on your network.
- The number of deny entries that can be specified in an ACL are dependent on the keyword specified:
  - **jump**—Supports up to 8 deny entries in an ACL.

    Note The limit of 8 deny jump entries in an ACL should be considered a guideline rather than a fixed limit. Depending on your configuration, the practical limit could be fewer than 8.

  - **clear**—Supports up to 1000 deny entries in an ACL.
  - **drop**—Supports up to 1000 deny entries in an ACL.

For a deny policy enhancements configuration example, see the "Deny Policy Enhancements for ACLs Configuration Example" section on page 25-30.

Configuration Examples

This section provides examples of the following configurations:

- Advanced Encryption Standard Configuration Example, page 25-23
- Reverse Route Injection Configuration Examples, page 25-23
- IPsec Anti-Replay Window Size Configuration Examples, page 25-24
- IPsec Preferred Peer Configuration Examples, page 25-26
- IPsec Security Association Idle Timer Configuration Examples, page 25-27
- Distinguished Name-Based Crypto Maps Configuration Example, page 25-27

- QoS Configuration Examples, page 25-28
- Platform ACL Configuration Example, page 25-30
- Deny Policy Enhancements for ACLs Configuration Example, page 25-30

Note The following examples use commands at the level of Cisco IOS Release 12.2(33)SXH.

As of Cisco IOS Release 12.2(33)SXH, the **crypto engine subslot** command used in previous releases has been replaced with the **crypto engine slot** command (of the form **crypto engine slot** *slot* {**inside** | **outside**}). The **crypto engine subslot** command is no longer supported. When upgrading, ensure that this command has been modified in your start-up configuration to avoid extended maintenance time.

Advanced Encryption Standard Configuration Example

The following example configures the Advanced Encryption Standard (AES) 256-bit key:

```
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
 mode transport
crypto map aesmap 10 ipsec-isakmp
 set peer 10.0.110.1
 set transform-set aesset
```

Reverse Route Injection Configuration Examples

The following examples show how to configure RRI:

- RRI Under a Static Crypto Map Configuration Example, page 25-23
- RRI Under a Dynamic Crypto Map Configuration Example, page 25-23
- RRI with Existing ACLs Configuration Example, page 25-24
- RRI for Two Routes Configuration Example, page 25-24
- RRI Through a User-Defined Hop Configuration Example, page 25-24

RRI Under a Static Crypto Map Configuration Example

The following example shows how to configure RRI under a static crypto map. In this example, the RRI-created route has been tagged with a tag number. This tag number can then be used by a routing process to redistribute the tagged route via a route map:

```
Router(config)# crypto map mymap 1 ipsec-isakmp
Router(config-crypto-map)# reverse-route tag 5
```

RRI Under a Dynamic Crypto Map Configuration Example

The following example shows how to configure RRI under a dynamic crypto map:

```
Router(config)# crypto dynamic-map mymap 1
Router(config-crypto-map)# reverse-route remote peer 10.1.1.1
```

RRI with Existing ACLs Configuration Example

The following example shows how to configure RRI for a situation in which there are existing ACLs:

```
Router(config)# crypto map mymap 1 ipsec-isakmp
Router(config-crypto-map)# set peer 172.17.11.1
Router(config-crypto-map)# reverse-route static
Router(config-crypto-map)# set transform-set esp-3des-sha
Router(config-crypto-map)# match address 101

access-list 101 permit ip 192.168.1.0 0.0.0.255 172.17.11.0 0.0.0.255
```

RRI for Two Routes Configuration Example

The following example shows how to configure two routes, one for the remote endpoint and one for route recursion to the remote endpoint via the interface on which the crypto map is configured:

```
Router(config-crypto-map)# reverse-route remote-peer
```

RRI Through a User-Defined Hop Configuration Example

The following example shows that one route has been created to the remote proxy through a user-defined next hop. This next hop should not require a recursive route lookup unless it will recurse to a default route.

```
Router(config-crypto-map)# reverse-route remote-peer 10.4.4.4
```

IPsec Anti-Replay Window Size Configuration Examples

The following examples show how to configure the IPsec anti-replay window size:

- IPsec Anti-Replay Window Global Configuration Example, page 25-24
- IPsec Anti-Replay Window per Crypto Map Configuration Example, page 25-25

IPsec Anti-Replay Window Global Configuration Example

The following example shows that the anti-replay window size has been set globally to 1024:

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname VPN-Gateway1
!
boot-start-marker
boot-end-marker
!
clock timezone EST 0
no aaa new-model
ip subnet-zero
!
ip audit po max-events 100
no ftp-server write-enable
!
```

```
crypto isakmp policy 10
 authentication pre-share
 crypto isakmp key cisco123
 address 192.165.201.2
!
crypto ipsec security-association replay window-size 1024
!
crypto ipsec transform-set basic esp-des esp-md5-hmac
!
crypto map mymap 10 ipsec-isakmp
 set peer 192.165.201.2
 set transform-set basic
 match address 101
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Serial1/0
 ip address 192.165.200.2 255.255.255.252
 serial restart-delay 0
 crypto map mymap
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.165.200.1
no ip http server
no ip http secure-server
!
access-list 101 permit ip 192.168.1.0 0.0.0.255 172.16.2.0 0.0.0.255
!access-list 101 remark Crypto ACL
!
control-plane
!
line con 0
line aux 0
line vty 0 4
end
```

IPsec Anti-Replay Window per Crypto Map Configuration Example

The following example shows that anti-replay checking is disabled for IPsec connections to 172.150.150.2, but enabled (and the default window size is 64) for IPsec connections to 172.150.150.3 and 172.150.150.4:

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname dr_whoovie
!
enable secret 5 $1$KxKv$cbqKsZtQTLJLGPN.tErFZ1
enable password ww
!
ip subnet-zero
cns event-service server
crypto isakmp policy 1
 authentication pre-share
 crypto isakmp key cisco170
 address 172.150.150.2
 crypto isakmp key cisco180
 address 172.150.150.3
 crypto isakmp key cisco190
 address 172.150.150.4
crypto ipsec transform-set 170cisco esp-des esp-md5-hmac
crypto ipsec transform-set 180cisco esp-des esp-md5-hmac
crypto ipsec transform-set 190cisco esp-des esp-md5-hmac
crypto map ETH0 17 ipsec-isakmp
 set peer 172.150.150.2
 set security-association replay disable
 set transform-set 170cisco
 match address 170
crypto map ETH0 18 ipsec-isakmp
 set peer 150.150.150.3
 set transform-set 180cisco
 match address 180
crypto map ETH0 19 ipsec-isakmp
 set peer 150.150.150.4
 set transform-set 190cisco
 match address 190
!
interface Ethernet0
 ip address 172.150.150.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no mop enabled
 crypto map ETH0
!
interface Serial0
 ip address 172.160.160.1 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
ip classless
ip route 172.170.170.0 255.255.255.0 172.150.150.2
ip route 172.180.180.0 255.255.255.0 172.150.150.3
ip route 172.190.190.0 255.255.255.0 172.150.150.4
no ip http server
!
access-list 170 permit ip 172.160.160.0 0.0.0.255 172.170.170.0 0.0.0.255
access-list 180 permit ip 172.160.160.0 0.0.0.255 172.180.180.0 0.0.0.255
access-list 190 permit ip 172.160.160.0 0.0.0.255 172.190.190.0 0.0.0.255
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
line con 0
transport input none
line aux 0
line vty 0 4
password ww
login
end
```

IPsec Preferred Peer Configuration Examples

The following examples show how to configure an IPsec preferred peer:

- Default Peer Configuration Example, page 25-27
- IPsec Idle Timer with Default Peer Configuration Example, page 25-27

Default Peer Configuration Example

The following example shows how to configure a default peer. In this example, the first peer, at IP address 1.1.1.1, is the default peer:

```
Router(config)# crypto map tohub 1 ipsec-isakmp
Router(config-crypto-map)# set peer 1.1.1.1 default
Router(config-crypto-map)# set peer 2.2.2.2
Router(config-crypto-map)# exit
```

IPsec Idle Timer with Default Peer Configuration Example

The following example shows how to configure an IPsec idle timer with a default peer. In the following example, if the current peer is idle for 600 seconds, the default peer 1.1.1.1 (which was specified in the **set peer** command) is used for the next attempted connection:

```
Router (config)# crypto map tohub 1 ipsec-isakmp
Router(config-crypto-map)# set peer 1.1.1.1 default
Router(config-crypto-map)# set peer 2.2.2.2
Router(config-crypto-map)# set security-association idle-time 600 default
Router(config-crypto-map)# exit
```

IPsec Security Association Idle Timer Configuration Examples

The following examples show how to configure the IPsec SA idle timer:

- IPsec SA Idle Timer Global Configuration Example, page 25-27
- IPsec SA Idle Timer per Crypto Map Configuration Example, page 25-27

IPsec SA Idle Timer Global Configuration Example

The following example globally configures the IPsec SA idle timer to drop SAs for inactive peers after 600 seconds:

```
Router(config)# crypto ipsec security-association idle-time 600
```

IPsec SA Idle Timer per Crypto Map Configuration Example

The following example configures the IPsec SA idle timer for the crypto map named test to drop SAs for inactive peers after 600 seconds:

```
Router(config) # crypto map test 1 ipsec-isakmp
Router(config-crypto-map)# set security-association idle-time 600
```

Distinguished Name-Based Crypto Maps Configuration Example

The following example shows how to configure distinguished name based crypto maps that have been authenticated by DN and hostname. Comments are included inline to explain various commands.

```
! DN based crypto maps require you to configure an IKE policy at each peer.
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
 crypto isakmp key 1234567890 address 171.69.224.33
!
!The following is an IPsec crypto map (part of IPsec configuration). It can be used only
! by peers that have been authenticated by DN and if the certificate belongs to BigBiz.
crypto map map-to-bigbiz 10 ipsec-isakmp
 set peer 172.21.114.196
 set transform-set my-transformset
 match address 124
 identity to-bigbiz
!
crypto identity to-bigbiz
dn ou=BigBiz
!
!
! This crypto map can be used only by peers that have been authenticated by hostname
!and if the certificate belongs to little.com.
crypto map map-to-little-com 10 ipsec-isakmp
 set peer 172.21.115.119
 set transform-set my-transformset
 match address 125
 identity to-little-com
!
crypto identity to-little-com
fqdn little.com
!
```

QoS Configuration Examples

The following examples show how to configure QoS for VPN:

- Module QoS Configuration Example, page 25-28
- Platform QoS Configuration Example, page 25-28

Module QoS Configuration Example

The following example shows how to configure the dual-priority queue for module QoS:

```
mls qos
!
Interface GigabitEthernet4/0/1
 mls qos trust cos
 priority-queue cos-map 1 0 1 5
!
Interface GigabitEthernet4/0/2
 mls qos trust cos
 priority-queue cos-map 1 0 1 5
```

Platform QoS Configuration Example

This example shows how to configure platform QoS with inbound and outbound service policies:

```
mls qos
!
! Define class maps
```

```
!
class-map match-any IPP1
 match ip precedence 1
class-map match-any IPP0
 match ip precedence 0
class-map match-any IPP3
 match ip precedence 3
class-map match-any IPP2
 match ip precedence 2
class-map match-any IPP5
 match ip precedence 5
class-map match-any IPP4
 match ip precedence 4
class-map match-any IPP7
 match ip precedence 7
class-map match-any IPP6
 match ip precedence 6
!
! Define policy maps
!
policy-map SET_3TO5
 class IPP3
  set precedence 5
!
policy-map SET_1TO5
 class IPP1
  set precedence 5
!
!
! LAN interface configuration
!
interface GigabitEthernet2/3
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 46,51,3501-4000
 switchport mode trunk
 mls qos trust ip-precedence
!
! inside interface configuration
!
interface GigabitEthernet3/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan none
 switchport mode trunk
 mtu 9216
 mls qos vlan-based
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
! outside interface configuration
!
interface GigabitEthernet3/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan none
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
!
! tunnel interface configuration
!
interface Tunnel1
```

```
 ip vrf forwarding i1
 ip address 26.0.1.2 255.255.255.0
 ip access-group T1ACL_IN in
 ip access-group T1ACL_OUT out
 ip mtu 1400
 tunnel source 27.0.1.2
 tunnel destination 192.0.20.1
 tunnel mode ipsec ipv4
 tunnel vrf f1
 tunnel protection ipsec profile TUN_PROTECTION
 crypto engine slot 3/0 inside
 service-policy input SET_1TO5
 service-policy output SET_3TO5
```

Platform ACL Configuration Example

This example shows a tunnel configuration with inbound and outbound platform ACLs:

```
interface Tunnel1
 ip vrf forwarding i1
 ip address 26.0.1.2 255.255.255.0
 ip access-group T1ACL_IN in
 ip access-group T1ACL_OUT out
 ip mtu 1400
 tunnel source 27.0.1.2
 tunnel destination 67.0.1.6
 tunnel vrf f1
 tunnel protection ipsec profile TUN_PROTECTION
 crypto engine slot 3/0 inside
!
!
ip access-list extended T1ACL_IN
 permit tcp any any
 permit icmp any any
 permit ip any host 50.0.1.2 precedence critical
 permit ip any host 50.0.1.2 precedence internet
 permit ip any host 50.0.1.2 precedence priority
 permit ip any host 50.0.1.2 precedence flash
 deny   ip any any
ip access-list extended T1ACL_OUT
 permit tcp any any
 permit icmp any any
 permit ip any host 60.0.1.2 precedence critical
 permit ip any host 60.0.1.2 precedence internet
 permit ip any host 60.0.1.2 precedence priority
 permit ip any host 60.0.1.2 precedence flash
 deny   ip any any
```

Deny Policy Enhancements for ACLs Configuration Example

The following example shows a configuration using the deny policy **clear** option. In this example, when a deny address is hit, the search will stop and traffic will be allowed to pass in the clear (unencrypted) state:

```
Router(config)# crypto ipsec ipv4-deny clear
```

CHAPTER 26

Configuring PKI Using the IPsec VPN SPA

This chapter provides information about configuring PKI-related features using the IPsec VPN SPA on the Catalyst 6500 Series switch. It includes the following sections:

Note The procedures in this chapter assume you have some familiarity with PKI configuration concepts. For detailed information about PKI configuration concepts and IPsec cryptographic operations and policies, refer to the following Cisco IOS documentation:

Cisco IOS Security Configuration Guide, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

Cisco IOS Security Command Reference, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

Note For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For additional information about the commands used in this chapter, see the the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* and the related Cisco IOS Release 12.2 software configuration guide and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Tip To ensure a successful configuration of your VPN using the IPsec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Overview of PKI

Cisco IOS public key infrastructure (PKI) provides certificate management to support security protocols such as IP Security (IPsec), secure shell (SSH), and secure socket layer (SSL).

A PKI is composed of the following entities:

- Peers communicating on a secure network
- At least one certificate authority (CA) that grants and maintains certificates
- Digital certificates, which contain information such as the certificate validity period, peer identity information, encryption keys that are used for secure communications, and the signature of the issuing CA
- An optional registration authority (RA) to offload the CA by processing enrollment requests
- A distribution mechanism (such as Lightweight Directory Access Protocol (LDAP) or HTTP) for certificate revocation lists (CRLs)

PKI provides customers with a scalable, secure mechanism for distributing, managing, and revoking encryption and identity information in a secured data network. Every entity (a person or a device) participating in the secured communications is enrolled in the PKI , a process where the entity generates a Rivest, Shamir, and Adelman (RSA) key pair (one private key and one public key) and has their identity validated by a trusted entity (also known as a CA or trustpoint).

After each entity enrolls in a PKI, every peer (also known as an end host) in a PKI is granted a digital certificate that has been issued by a CA. When peers must negotiate a secured communication session, they exchange digital certificates. Based on the information in the certificate, a peer can validate the identity of another peer and establish an encrypted session with the public keys contained in the certificate.

Configuring PKI involves the following tasks:

- Deploying Rivest, Shamir, and Adelman (RSA) keys within a public key infrastructure (PKI). An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certificate authority (CA) to obtain a certificate and enroll in a PKI.
- Configuring authorization and revocation of certificates within a PKI. After a certificate is validated as a properly signed certificate, it is authorized using methods such as certificate maps, PKI-AAA, or a certificate-based access control list (ACL). The revocation status is checked by the issuing certificate authority (CA) to ensure that the certificate has not been revoked.

- Configuring certificate enrollment, which is the process of obtaining a certificate from a certificate authority (CA). Certificate enrollment occurs between the end host requesting the certificate and the CA. Each peer that participates in the public key infrastructure (PKI) must enroll with a CA. Various methods are available for certificate enrollment.
- Storing public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adelman (RSA) keys and certificates. These credentials can be stored in the default location on the router, which is NVRAM, or other locations.

Configuring Multiple RSA Key Pairs

The multiple RSA key pair support feature allows you to configure a Catalyst 6500 Series switch to have multiple Rivest, Shamir, and Adelman (RSA) key pairs. The Cisco IOS software can maintain a different key pair for each identity certificate.

Before this feature, Cisco IOS public key infrastructure (PKI) configurations allowed either one general-purpose key pair or a set of special-purpose key pairs (an encryption and a signing key pair). The scenarios in which the key pairs were deployed often required configurations that required the switch to enroll with multiple certificate servers because each server has an independent policy and may also have different requirements regarding general-purpose versus special-purpose certificates or key length. With this feature, a user can configure different key pairs for each certification authority (CA) with which the switch enrolls and can match policy requirements for each CA without compromising the requirements specified by the other CAs, such as key length, key lifetime, and general-purpose versus special-usage keys.

Multiple RSA Key Pairs Configuration Guidelines and Restrictions

When configuring multiple RSA key pair support, follow these guidelines and restrictions:

- We recommend that Secure Socket Layer (SSL) or other PKI clients do not attempt to enroll with the same CA multiple times.
- Internet Key Exchange (IKE) will not work for any identity that is configured to use a named key pair. If an IKE peer requests a certificate from a PKI trustpoint that is using multiple key support, the initial portion of the exchange will work, that is, the correct certificate will be sent in the certificate response; however, the named keypair will not be used and the IKE negotiation will fail.
- Whenever you regenerate a key pair, you must always reenroll the certificate identities with that key pair.

To configure an RSA key pair, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto key generate rsa [usage-keys | general-keys] [modulus modulus-size] [key-pair-label]`	Generates RSA key pairs. • **usage-keys**—(Optional) Specifies that two special-usage key pairs should be generated, instead of one general-purpose key pair. • **general-keys**—(Optional) Specifies that the general-purpose key pair should be generated. • *key-pair-label*—(Optional) Specifies the name of the key pair that the switch will use. (If this argument is enabled, you must specify either **usage-keys** or **general-keys**.) • **modulus** *modulus-size*—(Optional) Specifies the modulus for generating the RSA keys. The range is 384 to 2048 bits, and the modulus must be a multiple of 64. The default is 1024.
Step 2	`Router(config)# crypto pki trustpoint name`	Declares the CA that the switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 3	`Router(ca-trustpoint)# rsakeypair key-label [key-size [encryption-key-size]]`	Specifies which key pair to associate with the certificate. • *key-label*—The name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) The size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) The size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.)

Removing RSA Key Pair Settings

To delete a specified RSA key pair or all RSA key pairs that have been generated by your switch, enter the **crypto key zeroize rsa** command in global configuration mode as follows:

```
Router(config)# crypto key zeroize rsa [key-pair-label]
```

key-pair-label specifies the name of the key pair to be deleted. If the *key-pair-label* argument is used, you will delete only the specified RSA key pair. If no argument is used, you will delete all the RSA key pairs from your switch.

Verifying RSA Key Information

To verify RSA key information, use at least one of the privileged EXEC commands used in the examples.

To display your switch's RSA public keys, use the **show crypto key mypubkey rsa** command:

```
Router# show crypto key mypubkey rsa

% Key pair was generated at: 06:07:50 UTC Jan 13 1996

Key name: myswitch.example.com

 Usage: Encryption Key

 Key Data:

  00302017 4A7D385B 1234EF29 335FC973 2DD50A37 C4F4B0FD 9DADE748 429618D5

  18242BA3 2EDFBDD3 4296142A DDF7D3D8 08407685 2F2190A0 0B43F1BD 9A8A26DB

  07953829 791FCDE9 A98420F0 6A82045B 90288A26 DBC64468 7789F76E EE21
```

To display a list of all the RSA public keys stored on your switch (including the public keys of peers that have sent your switch their certificates during peer authentication for IPsec), or to display details of a particular RSA public key stored on your switch, use the **show crypto key pubkey-chain rsa** command:

```
Router# show crypto key pubkey-chain rsa

Codes: M - Manually Configured, C - Extracted from certificate

Code  Usage         IP-address     Name

M     Signature     10.0.0.1       myrouter.example.com

M     Encryption    10.0.0.1       myrouter.example.com

C     Signature     172.16.0.1     routerA.example.com

C     Encryption    172.16.0.1     routerA.example.com

C     General       192.168.10.3   routerB.domain1.com
```

For complete configuration information for Multiple RSA Key Pair Support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/ftmltkey.html

For an RSA key pair configuration example, see the "Multiple RSA Key Pairs Configuration Example" section on page 26-54.

Configuring Protected Private Key Storage

The protected private key storage feature allows a user to encrypt and lock the RSA private keys that are used on a Catalyst 6500 Series switch, which prevents unauthorized use of the private keys.

Protected Private Key Storage Configuration Guidelines and Restrictions

When configuring protected private key storage, follow these guidelines and restrictions:

- An encrypted key is not effective after the switch boots up until you manually unlock the key (using the **crypto key unlock rsa** command). Depending on which key pairs are encrypted, this functionality may adversely affect applications such as IP Security (IPsec), Secure Shell (SSH) and Secure Socket Layer (SSL); that is, management of the switch over a secure channel may not be possible until the necessary key pair is unlocked.
- If a passphrase is lost, you must regenerate the key, enroll with the CA server again, and obtain a new certificate. A lost passphrase cannot be recovered.
- If you want to change a passphrase, you must decrypt the key with the current passphrase using the **crypto key decrypt rsa** command and encrypt the key once more to specify the new passphrase.

Configuring Private Keys

To encrypt, decrypt, lock, and unlock private keys, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto key encrypt [write] rsa [name key-name] passphrase passphrase`	Encrypts the RSA keys. After this command is entered, the switch can continue to use the key; the key remains unlocked. • **write**—(Optional) Switch configuration is immediately written to NVRAM. If the **write** keyword is not specified, the configuration must be manually written to NVRAM; otherwise, the encrypted key will be lost next time the switch is reloaded. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be encrypted. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to encrypt the RSA key. To access the RSA key pair, the passphrase must be specified.
Step 2	`Router(config)# exit`	Exits global configuration mode.
Step 3	`Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is encrypted (protected) and unlocked.

	Command	Purpose
Step 4	`Router# crypto key lock rsa [name key-name] passphrase passphrase`	(Optional) Locks the encrypted private key on a running switch. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be locked. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to lock the RSA key. To access the RSA key pair, the passphrase must be specified. **Note** After the key is locked, it cannot be used to authenticate the switch to a peer device. This behavior disables any IPsec or SSL connections that use the locked key. Any existing IPsec tunnels created on the basis of the locked key will be closed. If all RSA keys are locked, SSH will automatically be disabled.
Step 5	`Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is protected and locked. The output will also show failed connection attempts by applications such as IKE, SSH, and SSL.
Step 6	`Router# crypto key unlock rsa [name key-name] passphrase passphrase`	(Optional) Unlocks the private key. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be unlocked. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to unlock the RSA key. To access the RSA key pair, the passphrase must be specified. **Note** After this command is entered, you can continue to establish IKE tunnels.

	Command	Purpose
Step 7	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 8	`Router(config)#` **`crypto key decrypt`** `[`**`write`**`]` **`rsa`** `[`**`name`** *`key-name`*`]` **`passphrase`** *`passphrase`*	(Optional) Deletes the encrypted key and leaves only the unencrypted key. • **write**—(Optional) Unencrypted key is immediately written to NVRAM. If the **write** keyword is not specified, the configuration must be manually written to NVRAM; otherwise, the key will remain encrypted the next time the switch is reloaded. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be deleted. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to delete the RSA key. To access the RSA key pair, the passphrase must be specified.

Verifying the Protected and Locked Private Keys

To verify that the key is protected (encrypted) and locked, enter the **show crypto key mypubkey rsa** command:

```
Router# show crypto key mypubkey rsa

% Key pair was generated at:20:29:41 GMT Jun 20 2003
Key name:pki1-72a.cisco.com
Usage:General Purpose Key
*** The key is protected and LOCKED. ***
Key is exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
0D2B55AC 5D199F2F 7CB4B355 C555E07B 6D0DECBE 4519B1F0 75B12D6F 902D6E9F
B6FDAD8D 654EF851 5701D5D7 EDA047ED 9A2A619D 5639DF18 EB020301 0001
```

For complete configuration information for protected private key storage, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gt_ppkey.html

For protected private key configuration examples, see the "Protected Private Key Storage Configuration Examples" section on page 26-54.

Configuring a Trustpoint CA

The **crypto pki trustpoint** command allows you to declare the certificate authority (CA) that your switch should use and to specify characteristics for the CA.

The **crypto pki trustpoint** command combines and replaces the functionality of the existing **crypto ca identity** command and the **crypto ca trusted-root** command. Although both of these existing commands allow you to declare the certification authority (CA) that your switch should use, only the **crypto ca identity** command supports enrollment (the requesting of a switch certificate from a CA).

Trustpoint CA Configuration Guidelines and Restrictions

When configuring a trustpoint CA, follow these guidelines and restrictions:

- After the trustpoint CA has been configured, you can obtain the certificate of the CA by using the **crypto pki authenticate** command or you can specify that certificates should not be stored locally but retrieved from a CA trustpoint by using the **crypto pki certificate query** command.
- Normally, certain certificates are stored locally in the switch's NVRAM, and each certificate uses a moderate amount of memory. To save NVRAM space, you can use the **crypto pki certificate query** command to put the switch into query mode, preventing certificates from being stored locally; instead, they are retrieved from a specified CA trustpoint when needed. This will save NVRAM space but could result in a slight performance impact.

To declare the CA that your switch should use and specify characteristics for the trustpoint CA, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoint** *name*	Declares the CA that your switch should use. Enabling this command puts you in ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	Router(ca-trustpoint)# **enrollment** [[**mode ra**] | [**retry period** *minutes*] | [**retry count** *number*] | [**url** *url*]]	Specifies enrollment parameters for your CA. • **mode ra**—(Optional) Specifies registration authority (RA) mode if your CA system provides a RA. RA mode is turned off until you enable the **mode ra** keyword. • *minutes*—(Optional) Specifies the wait period between certificate request retries. The default is 1 minute between retries. (Specify from 1 to 60 minutes.) • *number*—(Optional) Specifies the number of times a switch will resend a certificate request when it does not receive a response from the previous request. The default is 10 retries. (Specify from 1 to 100 retries.) • *url*—Specifies the URL of the CA where your switch should send certificate requests; for example, http://ca_server. *url* must be in the form http://CA_*name*, where CA_*name* is the CA's host Domain Name System (DNS) name or IP address.
	Router(ca-trustpoint)# **root tftp** *server-hostname filename*	Obtains the CA via TFTP. • *server-hostname*—Name for the server that will store the trustpoint CA • *filename*—Name for the file that will store the trustpoint CA.

	Command	Purpose
Step 3	`Router(ca-trustpoint)# enrollment http-proxy host-name port-num`	Obtains the CA via HTTP through the proxy server. • *host-name*—Name of the proxy server used to get the CA. • *port-num*—Port number used to access the CA. **Note** This command can be used in conjunction only with the **enrollment** command.
Step 4	`Router(ca-trustpoint)# primary name`	(Optional) Assigns a specified trustpoint as the primary trustpoint of the switch. • *name*—Name of the primary trustpoint of the switch.
Step 5	`Router(ca-trustpoint)# crl {query url | optional}`	(Optional) Queries the certificate revocation list (CRL) to ensure that the certificate of the peer has not been revoked. • *url* —Lightweight Directory Access Protocol (LDAP) URL published by the certificate authority (CA) server is specified to query the CRL; for example, ldap://another_server. • **optional**—CRL verification is optional. **Note** If the **query** *url* option is not enabled, the switch will check the certificate distribution point (CDP) that is embedded in the certificate.
Step 6	`Router(ca-trustpoint)# default command-name`	(Optional) Sets the value of ca-trustpoint configuration mode to its default. • *command-name*—pki-trustpoint configuration subcommand. Default is off.
Step 7	`Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 8	`Router(config)# crypto pki authenticate name`	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 9	`Router(config)# crypto pki trustpoint name`	Reenters ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 10	`Router(ca-trustpoint)# crypto pki certificate query`	(Optional) Turns on query mode per specified trustpoint, causing certificates not to be stored locally.

Verifying a Trustpoint CA

To verify information about your certificate, the certificate of the CA, and registration authority (RA) certificates, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates
```

```
CA Certificate

  Status: Available

  Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F

  Key Usage: Not Set

RA Signature Certificate

  Status: Available

  Certificate Serial Number: 34BCF8A0

  Key Usage: Signature


RA KeyEncipher Certificate

  Status: Available

  Certificate Serial Number: 34BCF89F

  Key Usage: Encryption
```

To display the trustpoints that are configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:
Subject Name:
CN = bomborra Certificate Manager
O = cisco.com
C = US
Serial Number:01
Certificate configured.
CEP URL:http://bomborra
CRL query url:ldap://bomborra
```

For complete configuration information for the trustpoint CA, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/fttrust.html

For a trustpoint CA configuration example, see the "Trustpoint CA Configuration Example" section on page 26-54.

Configuring Query Mode Definition Per Trustpoint

Certificates contain public key information and are signed by certificate authority (CA) as proof of identity. Normally, all certificates are stored locally in the switch's NVRAM, and each certificate uses a moderate amount of memory. The query mode definition per trustpoint feature allows you to define a query for a specific trustpoint so that the certificates associated with that specific trustpoint can be stored on a remote server.

This feature is especially useful for environments where multiple trustpoints are configured on a switch because it allows you more control over use of the trustpoint. Query mode can be activated on specific trustpoints rather than on all of the trustpoints on a switch.

Query Mode Definition Per Trustpoint Configuration Guidelines and Restrictions

When configuring query mode definition per trustpoint, follow these guidelines and restrictions:

- Normally, certain certificates are stored locally in the switch's NVRAM, and each certificate uses a moderate amount of memory. To save NVRAM space, you can use the **query certificate** command to prevent certificates from being stored locally; instead, they are retrieved from a remote server, such as a CA or LDAP server, during startup. This will save NVRAM space but could result in a slight performance impact.
- Certificates associated with a specified trustpoint will not be written into NVRAM and the certificate query will be attempted during the next reload of the switch.
- When the global **crypto pki certificate query** command is used, the query certificate will be added to all trustpoints on the switch. When the **no crypto pki certificate query** command is used, any previous query certificate configuration will be removed from all trustpoints and any query in progress will be halted and the feature disabled.

To configure a trustpoint CA and initiate query mode for the trustpoint, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use. Enabling this command puts you in ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	`Router(ca-trustpoint)# enrollment [[mode ra] | [retry period minutes] | [retry count number] | [url url]]`	Specifies enrollment parameters for your CA. • **mode ra**—(Optional) Specifies registration authority (RA) mode if your CA system provides a RA. RA mode is turned off until you enable the **mode ra** keyword. • *minutes*—(Optional) Specifies the wait period between certificate request retries. The default is 1 minute between retries. (Specify from 1 to 60 minutes.) • *number*—(Optional) Specifies the number of times a switch will resend a certificate request when it does not receive a response from the previous request. The default is 10 retries. (Specify from 1 to 100 retries.) • *url*—Specifies the URL of the CA where your switch should send certificate requests; for example, http://ca_server. *url* must be in the form http://CA_*name*, where CA_*name* is the CA's host Domain Name System (DNS) name or IP address.

	Command	Purpose
Step 3	Router(ca-trustpoint)# **enrollment http-proxy** *host-name port-num*	(Optional) Obtains the CA via HTTP through the proxy server. • *host-name*—Name of the proxy server used to get the CA. • *port-num*—Port number used to access the CA. **Note** This command can be used in conjunction only with the **enrollment** command.
Step 4	Router(ca-trustpoint)# **crl query** *url*	(Optional) Specifies the URL for the CA server if the CA server supports query mode through LDAP. • *url* —Lightweight Directory Access Protocol (LDAP) URL published by the certificate authority (CA) server.
Step 5	Router(ca-trustpoint)# **query certificate**	Turns on query mode per specified trustpoint, causing certificates not to be stored locally and to be retrieved from a remote server.
Step 6	Router(ca-trustpoint)# **exit**	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 7	Router(config)# **crypto pki authenticate** *name*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 8	Router(config)# **crypto key generate rsa**	(Optional) Generates RSA key pairs.
Step 9	Router(config)# **crypto pki enroll** t*rustpoint-nam*e	(Optional) Obtains switch certificate. • *trustpoint-name*—Name of the CA. Enter the *name* value entered in Step 1.

Verifying Query Mode Definition Per Trustpoint CA

For query mode to operate correctly during the next reload, the certificates must be associated with the trustpoint. Use the **show crypto pki certificates** command to verify that each of the trustpoints has the needed certificates before storing the configuration and reloading the switch:

```
Router# show crypto pki certificates

Trustpoint yni:

  Issuing CA certificate pending:

    Subject Name:

     cn=nsca-r1 Cert Manager,ou=pki,o=cisco.com,c=US

    Fingerprint: C21514AC 12815946 09F635ED FBB6CF31

  Router certificate pending:

    Subject Name:

     hostname=trance.cisco.com,o=cisco.com
```

```
Next query attempt:

  52 seconds
```

For complete configuration information for Query Mode Definition Per Trustpoint, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gt_qerym.html

For a query mode definition per trustpoint configuration example, see the "Query Mode Definition Per Trustpoint Configuration Example" section on page 26-55.

Configuring a Local Certificate Storage Location

The Local Certificate Storage Location feature enables you to store public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adelman (RSA) keys and certificates in a specific location. An example of a certificate storage location includes NVRAM, which is the default location, and other local storage locations as supported by your platform, such as flash.

Note The Local Certificate Storage Location feature is supported in Cisco IOS Release 12.2(33)SXH and later releases.

Local Certificate Storage Location Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring a local certificate storage location:

- Before you can specify the local certificate storage location, your system should meet the following requirements:
  - A Cisco IOS Release 12.4(2)T PKI-enabled image or a later image
  - A platform that supports storing PKI credentials as separate files
  - A configuration that contains at least one certificate
  - An accessible local file system
- When storing certificates to a local storage location, the following restrictions are applicable:
  - Only local file systems may be used. An error message will be displayed if a remote file system is selected, and the command will not take effect.
  - A subdirectory may be specified if supported by the local file system. NVRAM does not support subdirectories.
  - Certificates are stored to NVRAM by default, however, some routers do not have the required amount of NVRAM to successfully store certificates. Introduced in Cisco IOS Release 12.4(2)T is the ability to specify where certificates are stored on a local file system.
  - During run time, you can specify what active local storage device you would like to use to store certificates.

Specifying a Local Storage Location for Certificates

To specify the local storage location for certificates, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki certificate storage location-name`	Specifies the local storage location for certificates. • *location-name*—Name of the storage location.
Step 2	`Router (config)# exit`	Exits global configuration mode.
Step 3	`Router# copy source-url destination-url`	(Optional) Saves the running configuration to the startup configuration. • *source-url*—The location URL (or alias) of the source file or directory to be copied. The source can be either local or remote, depending upon whether the file is being downloaded or uploaded. • *destination-url*—The destination URL (or alias) of the copied file or directory. The destination can be either local or remote, depending upon whether the file is being downloaded or uploaded. **Note** Settings will only take effect when the running configuration is saved to the startup configuration.

Verifying the Local Certificate Storage Location Configuration

To verify a local certificate storage location configuration, enter the **show crypto pki certificates storage** command.

The **show crypto pki certificates storage** command displays the current setting for the PKI certificate storage location.

The following example shows that certificates are stored in the certs subdirectory of disk0:

```
Router# show crypto pki certificates storage

Certificates will be stored in disk0:/certs/
```

For complete configuration information for local certificate storage location, refer to the *Cisco IOS Security Configuration Guide* or the following URL:

http://www.cisco.com/en/US/docs/ios/12_4t/12_4t2/st_pkist.html

For local certificate storage configuration examples, see the "Local Certificate Storage Location Configuration Example" section on page 26-55.

Configuring Direct HTTP Enroll with CA Servers (Reenroll Using Existing Certificates)

The direct HTTP enroll with CA servers feature allows users to bypass the registration authority (RA) when enrolling with a certification authority (CA) by configuring an enrollment profile. HTTP enrollment requests can be sent directly to the CA server.

The reenroll using existing certificates functionality allows a switch that is enrolled with a third-party vendor CA to use its existing certificate to enroll with the Cisco IOS certificate server so the enrollment request is automatically granted.

Direct HTTP Enroll with CA Servers Configuration Guidelines and Restrictions

When configuring direct HTTP enroll with CA servers, follow these guidelines and restrictions:

- The CA certificate and switch certificates must be returned in the privacy enhanced mail (PEM) format.
- If an enrollment profile is specified, an enrollment URL can not be specified in the trustpoint configuration.
- Because there is no standard for the HTTP commands used by various CAs, the user is required to enter the command that is appropriate to the CA that is being used.
- The newly created trustpoint can only be used one time (which occurs when the switch is enrolled with the Cisco IOS CA). After the initial enrollment is successfully completed, the credential information will be deleted from the enrollment profile.
- The Cisco IOS certificate server will automatically grant only the requests from clients who were already enrolled with the non-Cisco IOS CA. All other requests must be manually granted unless the server is set to be in auto grant mode (using the **grant automatic** command).
- To configure direct HTTP enroll with CA servers, you must perform the following steps:
  - Either configure a certificate enrollment profile for the client switch (see the "Configuring an Enrollment Profile for a Client Switch" section on page 26-17) or configure an enrollment profile for a client switch that is already enrolled with a third-party vendor (see the "Configuring an Enrollment Profile for a Client Switch Enrolled with a Third-Party Vendor CA" section on page 26-19).
  - Configure the CA certificate server to accept enrollment requests only from clients who are already enrolled with the third-party vendor CA trustpoint (see the "Configuring the CA to Accept Enrollment Requests from Clients of a Third-Party Vendor CA" section on page 26-20).

Configuring an Enrollment Profile for a Client Switch

To configure a certificate enrollment profile, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the trustpoint a given name and enters ca-trustpoint configuration mode. - *name*—Name of the CA trustpoint.
Step 2	`Router(ca-trustpoint)# enrollment profile label`	Specifies that an enrollment profile can be used for certificate authentication and enrollment. - *label*—Name for the enrollment profile.
Step 3	`Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 4	`Router(config)# crypto pki profile enrollment label`	Defines an enrollment profile and enters ca-profile-enroll configuration mode. - *label*—Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command.
Step 5	`Router(ca-profile-enroll)# authentication url url`	(Optional) Specifies the URL of the CA server to which to send certificate authentication requests. - *url*—URL of the CA server to which your switch should send authentication requests. If using HTTP, the URL should read "http://CA_name," where CA_name is the host Domain Name System (DNS) name or IP address of the CA. If using TFTP, the URL should read "tftp://certserver/file_specification." (If the URL does not include a file specification, the fully qualified domain name (FQDN) of the switch will be used.
	`Router(ca-profile-enroll)# authentication terminal`	(Optional) Specifies manual cut-and-paste certificate authentication.
Step 6	`Router(ca-profile-enroll)# authentication command http-command`	(Optional) Sends the HTTP request to the CA for authentication. - *http-command*—HTTP request to be sent to the CA server. This command should be used after the **authentication url** command has been entered.
Step 7	`Router(ca-profile-enroll)# enrollment url url` or	Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP or TFTP. - *url*—URL of the CA server.
	`Router(ca-profile-enroll)# enrollment terminal`	Specifies manual cut-and-paste certificate enrollment.

	Command	Purpose
Step 8	`Router(ca-profile-enroll)# enrollment command http-command`	(Optional) Specifies the HTTP command to be sent to the CA for enrollment. - *http-command*—HTTP command to be sent to the CA server.
Step 9	`Router(ca-profile-enroll)# parameter number {value value | prompt string}`	(Optional) Specifies parameters for an enrollment profile. - *number*—User parameters. Valid values range from 1 to 8. - *value*—To be used if the parameter has a constant value. - *string*—To be used if the parameter is supplied after the **crypto pki authenticate** command or the **crypto pki enroll** command has been entered. **Note** The value of the *string* argument does not have an effect on the value that is used by the switch. This command can be used multiple times to specify multiple values.
Step 10	`Router(ca-profile-enroll config)# exit`	Exits ca-profile-enroll configuration mode and enters global configuration mode.
Step 11	`Router(config)# exit`	Exits global configuration mode and enters Privileged EXEC mode.
Step 12	`Router# show crypto pki certificates`	(Optional) Verifies information about your certificate, the certificate of the CA, and RA certificates.
Step 13	`Router# show crypto pki trustpoints`	(Optional) Displays the trustpoints that are configured in the switch.

In configuring the direct HTTP enrollment profile, you can use the **parameter** command within an enrollment profile to provide predefined or console-input parameters to the **authentication command** command or the **enrollment command** command.

When you enter the **parameter** *number* command, a macroinstruction (macro) is created and named $P*number* (for example, $P1). If the **value** keyword is specified, the *value* argument is assigned to the macro. If the prompt keyword is specified, when the switch executes the **authentication command** command or the **enrollment command** command, the console will display the *string* argument as a prompt for user input. You can then enter a value to be assigned to the macro.

In addition to user-defined macros, three predefined macros are available:

- $REQ—The Certification Request Standard (PKCS #10) message to request certification of a key
- $FQDN—The FQDN of the switch
- $HOST—The hostname of the switch

This example shows how to use one predefined and three user-defined macros:

```
Router(config)# crypto ca profile enrollment E
Router(ca-profile-enroll)# authentication url http://ca-server.example.com
Router(ca-profile-enroll)# authentication command GET $P1
Router(ca-profile-enroll)# enrollment url
Router(ca-profile-enroll)# enrollment command ^C
POST action=getServerCert
&pkcs10Request=$REQ
&reference_number=$P2
&authcode=P3
&retrievedAs=rawDER
^C
Router(ca-profile-enroll)# parameter 1 value cacert.crt
Router(ca-profile-enroll)# parameter 2 prompt Enter the Reference Number:
Router(ca-profile-enroll)# parameter 3 prompt Enter the Auth Code:
```

Before the HTTP authentication and enrollment commands are posted by the switch to the CA, the console will prompt for any required user input, and macro values will be substituted for the macro names in the posted commands.

For an example of how to configure an enrollment profile for direct HTTP enrollment with a CA server, see the "Enrollment Profile for a Client Switch Configuration Example" section on page 26-55.

Configuring an Enrollment Profile for a Client Switch Enrolled with a Third-Party Vendor CA

When a client switch is already enrolled with a third-party vendor CA, but you want to reenroll that switch with a Cisco IOS certificate server, perform the following procedures. Note that some prerequisite steps are required before beginning the configuration.

Prerequisites

Before configuring a certificate enrollment profile for the client switch enrolled with a third-party vendor, you should have already performed the following tasks at the client switch:

- Defined a trustpoint that points to a third-party vendor CA.
- Authenticated and enrolled the client switch with the third-party vendor CA.

To configure a certificate enrollment profile for a client switch that is already enrolled with a third-party vendor CA so that the switch can reenroll with a Cisco IOS certificate server, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the Cisco IOS CA that is to be used.
Step 2	`Router(ca-trustpoint)# enrollment profile label`	Specifies that an enrollment profile is to be used for certificate reenrollment. • *label*—Name for the enrollment profile.
Step 3	`Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and enters global configuration mode.

	Command	Purpose
Step 4	`Router(config)# `**`crypto pki profile enrollment`** *`label`*	Defines an enrollment profile and enters ca-profile-enroll configuration mode. • *label*—Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command in Step 2.
Step 5	`Router(ca-profile-enroll)# `**`enrollment url`** *`url`*	Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP. • *url*—The enrollment URL should point to the Cisco IOS CA.
Step 6	`Router(ca-profile-enroll)# `**`enrollment credential`** *`label`*	Specifies the non-Cisco IOS CA trustpoint that is to be enrolled with the Cisco IOS CA. • *label*—Name of the CA trustpoint of another vendor.
Step 7	`Router(ca-profile-enroll)# `**`exit`**	Exits ca-profile-enroll configuration mode and enters global configuration mode.
Step 8	`Router(config)# `**`exit`**	Exits global configuration mode and enters privileged EXEC mode.
Step 9	`Router# `**`show crypto pki certificates`**	(Optional) Verifies information about your certificate, the certificate of the CA, and RA certificates
Step 10	`Router# `**`show crypto pki trustpoints`**	(Optional) Displays the trustpoints that are configured in the switch.

For an example of how to configure a certificate enrollment profile for a client switch that is already enrolled with a third-party vendor CA, see the "Enrollment Profile for a Client Switch Already Enrolled with a Third-Party Vendor CA Example" section on page 26-56.

Configuring the CA to Accept Enrollment Requests from Clients of a Third-Party Vendor CA

To configure the CA certificate server to accept enrollment requests only from clients who are already enrolled with the third-party vendor CA trustpoint, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`ip http server`**	Enables the HTTP server on your system.
Step 2	`Router(config)# `**`crypto pki server`** *`cs-label`*	Enables the certificate server and enters certificate server configuration mode. • *cs-label*—The *cs-label* argument must match the name that was specified by the **crypto pki trustpoint** command for the client switch.

	Command	Purpose
Step 3	`Router(cs-server)#` **`database url`** *`root-url`*	Specifies the location where all database entries for the certificate server will be stored. • *root-url*—Root URL. **Note** If this command is not specified, all database entries will be written to NVRAM.
Step 4	`Router(cs-server)#` **`database level`** `{`**`minimal`** `|` **`names`** `|` **`complete`**`}`	Controls what type of data is stored in the certificate enrollment database. • **minimal**—Enough information is stored only to continue issuing new certificates without conflict; the default value. • **names**—In addition to the information given in the minimal level, the serial number and subject name of each certificate. • **complete**—In addition to the information given in the minimal and names levels, each issued certificate is written to the database. **Note** The **complete** keyword produces a large amount of information; if it is specified, you should also specify an external TFTP server in which to store the data using the **database url** command.
Step 5	`Router(cs-server)#` **`issuer-name`** *`DN-string`*	Sets the CA issuer name to the specified DN-string. • *DN-string*—The default value is as follows: **issuer-name CN=*cs***-*label*.
Step 6	`Router(cs-server)#` **`grant auto trustpoint`** *`label`*	Enables the certificate server to automatically grant only the requests from clients that are already enrolled with the specified non-Cisco IOS CA trustpoint. • *label*—Name of the CA trustpoint of another vendor. **Note** The *label* argument should match the trustpoint that was specified for the client switch's enrollment profile (using the **enrollment credential** command).
Step 7	`Router(cs-server)#` **`lifetime`** `{`**`ca-certificate`** `|` **`certificate`**`}` *`time`*	(Optional) Specifies the lifetime, in days, of a CA certificate or a certificate. • *time*—Valid values range from 1 day to 1825 days. The default CA certificate lifetime is 3 years; the default certificate lifetime is 1 year. The maximum certificate lifetime is 1 month less than the lifetime of the CA certificate.

	Command	Purpose
Step 8	`Router(cs-server)#` **`lifetime crl`** *`time`*	(Optional) Defines the lifetime, in hours, of the Certificate Revocation List (CRL) that is used by the certificate server. • *time*—Maximum lifetime value is 336 hours (2 weeks). The default value is 168 hours (1 week).
Step 9	`Router(cs-server)#` **`cdp-url`** *`url`*	(Optional) Defines a Certificate Distribution Point (CDP) to be used in the certificates that are issued by the certificate server. • *url*—URL must be an HTTP URL.
Step 10	`Router(cs-server)#` **`shutdown`**	Disables a certificate server without removing the configuration. You should enter this command only after you have completely configured your certificate server.
Step 11	`Router(cs-server)#` **`exit`**	Exits certificate server configuration mode.
Step 12	`Router(config)#` **`exit`**	Exits global configuration mode.
Step 13	`Router#` **`show crypto pki server`**	(Optional) Displays the current state and configuration of the certificate server.

For complete configuration information for direct HTTP enroll with CA servers, including the "reenroll using existing certificates" functionality, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t4/feature/guide/gthttpca.html

For direct HTTP enroll with CA servers configuration examples, see the "Direct HTTP Enrollment with CA Servers Configuration Examples" section on page 26-55.

Configuring Manual Certificate Enrollment (TFTP and Cut-and-Paste)

The manual certificate enrollment (TFTP and cut-and-paste) feature allows users to generate a certificate request and accept certification authority (CA) certificates as well as the switch's certificates; these tasks are accomplished by a TFTP server or manual cut-and-paste operations. You might want to utilize TFTP or manual cut-and-paste enrollment in the following situations:

- The CA does not support Simple Certificate Enrollment Protocol (SCEP) (which is the most commonly used method for sending and receiving requests and certificates).
- A network connection between the switch and CA is not possible (which is how a switch running Cisco IOS software obtains its certificate).

Manual Certificate Enrollment (TFTP and Cut-and-Paste) Configuration Guidelines and Restrictions

When configuring nanualcertificate enrollment (TFTP and cut-and-paste), follow these guidelines and restrictions:

- You can switch between TFTP and cut-and-paste; for example, you can paste the CA certificate using the **enrollment terminal** command, then enter **no enrollment terminal** and **enrollment url tftp://certserver/file_specification** to switch to TFTP to send or receive requests and switch certificates. However, Cisco does not recommend switching URLs if SCEP is used; that is, if the enrollment URL is http://, do not change the enrollment URL between fetching the CA certificate and enrolling the certificate.

Configuring Manual Enrollment Using TFTP

Before configuring manual enrollment using TFTP, you must meet the following prerequisites:

- You must know the correct URL to use if you are configuring certificate enrollment using TFTP.
- The switch must be able to write a file to the TFTP server for the **crypto pki enroll** command.
- Some TFTP servers require that the file exist on the server before it may be written.
- Most TFTP servers require that the file be writeable by anyone. This requirement may pose a risk because any switch or other device may write or overwrite the certificate request; thus, the switch will not be able to use the certificate once it is granted by the CA because the request was modified.

To declare the trustpoint CA that your switch should use and configure that trustpoint CA for manual enrollment using TFTP, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)# enrollment [mode] [retry period minutes] [retry count number] url url`	Specifies the enrollment parameters of your CA. • **mode**—Specifies registration authority (RA) mode if your CA system provides a RA. • *minutes*—Specifies the wait period between certificate request retries. The default is 1 minute between retries. • *number*—Specifies the number of times a switch will resend a certificate request when it does not receive a response from the previous request. (Specify from 1 to 100 retries.) • *url*—Specifies the URL of the CA where your switch should send certificate requests. If you are using SCEP for enrollment, the URL must be in the form http://CA_name, where CA_name is the CA's host Domain Name System (DNS) name or IP address. If you are using TFTP for enrollment, the URL must be in the form tftp://certserver/file_specification.

	Command	Purpose
Step 3	`Router(ca-trustpoint)#` **`crypto pki authenticate`** *`name`*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 4	`Router(ca-trustpoint)#` **`exit`**	Exits ca-trustpoint configuration mode and returns to global configuration.
Step 5	`Router(config)#` **`crypto pki enroll`** *`name`*	Obtains your switch's certificates from the CA. • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 6	`Router(config)#` **`crypto pki import`** *`name`* **`certificate`**	Imports a certificate using TFTP. • *name*—Name of the CA. Enter the *name* value entered in Step 1.

Configuring Certificate Enrollment Using Cut-and-Paste

To declare the trustpoint CA that your switch should use and configure that trustpoint CA for manual enrollment using cut-and-paste, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`crypto pki trustpoint`** *`name`*	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)#` **`enrollment terminal`**	Specifies manual cut-and-paste certificate enrollment.
Step 3	`Router(ca-trustpoint)#` **`crypto pki authenticate`** *`name`*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Specifies the name of the CA. Enter the *name* value entered in Step 1.
Step 4	`Router(ca-trustpoint)#` **`exit`**	Exits ca-trustpoint configuration mode and returns to global configuration.

	Command	Purpose
Step 5	`Router(config)# crypto pki enroll name`	Obtains your switch's certificates from the CA. • *name*—Specifies the name of the CA. Enter the *name* value entered in Step 1.
Step 6	`Router(config)# crypto pki import name certificate`	Imports a certificate manually at the terminal. • *name*—Specifies the name of the CA. Enter the *name* value entered in Step 1. **Note** You must enter the **crypto pki import** command twice if usage keys (signature and encryption keys) are used. The first time the command is entered, one of the certificates is pasted into the switch; the second time the command is entered, the other certificate is pasted into the switch. (It does not matter which certificate is pasted first.)

Verifying the Manual Certificate Enrollment Configuration

To verify information about your certificate, the certificate of the CA, and RA certificates, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates

Certificate
  Status:Available
  Certificate Serial Number:14DECE05000000000C48
  Certificate Usage:Encryption

  Issuer:
    CN = msca-root
    O = Cisco Systems
    C = U

  Subject:
    Name:Switch.cisco.com
    OID.1.2.840.113549.1.9.2 = Switch.cisco.com

     CRL Distribution Point:
    http://msca-root/CertEnroll/msca-root.crl

     Validity Date:
    start date:18:16:45 PDT Jun 7 2008
    end   date:18:26:45 PDT Jun 7 2009
    renew date:16:00:00 PST Dec 31 1969

     Associated Trustpoints:MS

   Certificate
    Status:Available
    Certificate Serial Number:14DEC2E9000000000C47
    Certificate Usage:Signature

     Issuer:
    CN = msca-root
    O = Cisco Systems
    C = US
     Subject:
    Name:Switch.cisco.com
    OID.1.2.840.113549.1.9.2 = Switch.cisco.com

     CRL Distribution Point:
       http://msca-root/CertEnroll/msca-root.crl

     Validity Date:
    start date:18:16:42 PDT Jun 7 2008
    end   date:18:26:42 PDT Jun 7 2009
    renew date:16:00:00 PST Dec 31 1969

     Associated Trustpoints:MS

   CA Certificate
    Status:Available
    Certificate Serial Number:3AC0A65E9547C2874AAF2468A942D5EE
    Certificate Usage:Signature

     Issuer:
    CN = msca-root
    O = Cisco Systems
    C = US

     Subject:
    CN = msca-root
    O = Cisco Systems
    C = US

     CRL Distribution Point:
       http://msca-root/CertEnroll/msca-root.crl

  Validity Date:
    start date:16:46:01 PST Feb 13 2008
    end   date:16:54:48 PST Feb 13 2013

  Associated Trustpoints:MS
```

To display the trustpoints that are configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:

    Subject Name:

    CN = ACSWireless Certificate Manager

     O = cisco.com

     C = US

          Serial Number:01

    Certificate configured.

    CEP URL:http://ACSWireless

    CRL query url:ldap://ACSWireless
```

For complete configuration information for manual certificate enrollment (TFTP and cut-and-paste), refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftmancrt.html

For manual certificate enrollment configuration examples, see the "Manual Certificate Enrollment Configuration Examples" section on page 26-57.

Configuring Certificate Autoenrollment

The certificate autoenrollment feature allows you to configure your switch to automatically request a certificate from the certification authority (CA) that is using the parameters in the configuration. Thus, operator intervention is no longer required at the time the enrollment request is sent to the CA server.

Automatic enrollment will be performed on startup for any trustpoint CA that is configured and does not have a valid certificate. When the certificate expires that is issued by a trustpoint CA that has been configured for autoenrollment, a new certificate is requested. Although this feature does not provide seamless certificate renewal, it does provide unattended recovery from expiration.

Before the certificate autoenrollment feature, certificate enrollment required complicated, interactive commands that had to be executed on every switch. This feature allows you to preload all of the necessary information into the configuration and cause each switch to obtain certificates automatically when it is booted. Autoenrollment also checks for expired switch certificates.

Note Before submitting an automatic enrollment request, all necessary enrollment information must be configured.

To configure autoenrollment with a CA on startup, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the name of the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)# enrollment url url`	Specifies the URL of the CA on which your switch should send certificate requests; for example, http://ca_server. • *url*—Must be in the form of http://CA_name, where CA_name is the name of the CA's host Domain Name System or the IP address.
Step 3	`Router(ca-trustpoint)# subject-name [x.500-name]`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—(Optional) If the *x-500-name* argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.
Step 4	`Router(ca-trustpoint)# ip-address {interface | none}`	Includes the IP address of the specified interface in the certificate request. • *interface*—IP address of the interface. • **none**—Specify this keyword if no IP address should be included. If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 5	`Router(ca-trustpoint)# serial-number [none]`	Specifies the switch serial number in the certificate request, unless the **none** keyword is specified. • **none**—(Optional) Specify this keyword if no serial number should be included.
Step 6	`Router(ca-trustpoint)# auto-enroll [regenerate]`	Enables autoenrollment. This command allows you to automatically request a switch certificate from the CA. By default, only the DNS name of the switch is included in the certificate. • **regenerate**—(Optional) Specify this keyword to generate a new key for the certificate even if a named key already exists.

	Command	Purpose
Step 7	`Router(ca-trustpoint)# **password** *string*`	(Optional) Specifies the revocation password for the certificate. • *string*—Text of the password. **Note** If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.
Step 8	`Router(ca-trustpoint)# **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]`	Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) If this command is not enabled, the FQDN key pair is used.

Preloading Root CAs

After enabling automatic enrollment, you must authenticate the CA to establish a chain of trust. This can be done by implementing one of the following methods:

- Obtaining the CA Certificate, page 26-29
- Adding the Certificate of the CA, page 26-30

Obtaining the CA Certificate

To obtain the certificate of the CA, enter the **crypto pki authenticate** command in global configuration mode.

```
Router(config)# crypto pki authenticate name
```

name specifies the name of the CA.

Adding the Certificate of the CA

To add the certificate of the CA, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki certificate chain name`	Enters certificate chain configuration mode, which allows you to add or delete specified certificates. • *name*—Name of the CA.
Step 2	`Router(config-cert-chain)# certificate certificate-serial-number`	Manually adds or deletes certificates. • *certificate-serial-number*—Serial number of the CA to add.

Verifying CA Information

To display information about your certificates, the certificates of the CA, and registration authority (RA) certificates, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates

Certificate

  Subject Name

    Name: myrouter.example.com
    IP Address: 10.0.0.1

  Status: Available

  Certificate Serial Number: 428125BDA34196003F6C78316CD8FA95

  Key Usage: Signature


Certificate

  Subject Name

    Name: myswitch.example.com
    IP Address: 10.0.0.1

  Status: Available

  Certificate Serial Number: AB352356AFCD0395E333CCFD7CD33897

  Key Usage: Encryption


CA Certificate

  Status: Available

  Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F

  Key Usage: Not Set
```

To display the trustpoints configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:

    Subject Name:

    CN = bomborra Certificate Manager

     O = cisco.com

     C = US

    Serial Number:01

    Certificate configured.

    CEP URL:http://bomborra

    CRL query url:ldap://bomborra
```

For complete configuration information for Certificate Autoenrollment, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/ftautoen.html

For a certificate autoenrollment configuration example, see the "Certificate Autoenrollment Configuration Example" section on page 26-60.

Configuring Key Rollover for Certificate Renewal

Automatic certificate enrollment was introduced to allow the switch to automatically request a certificate from the certification authority (CA) server. By default, the automatic enrollment feature requests a new certificate when the old certificate expires. Connectivity can be lost while the request is being serviced because the existing certificate and key pairs are deleted immediately after the new key is generated. The new key does not have a certificate to match it until the process is complete, and incoming Internet Key Exchange (IKE) connections cannot be established until the new certificate is issued. The key rollover for certificate renewal feature allows the certificate renewal request to be made before the certificate expires and retains the old key and certificate until the new certificate is available.

Key rollover can also be used with a manual certificate enrollment request. Using the same method as key rollover with certificate autoenrollment, a new key pair is created with a temporary name, and the old certificate and key pair are retained until a new certificate is received from the CA. When the new certificate is received, the old certificate and key pair are discarded and the new key pair is renamed with the name of the original key pair. Do not regenerate the keys manually; key rollover will occur when you enter the **crypto pki enroll** command.

Key Rollover for Certificate Renewal Configuration Guidelines and Restrictions

When configuring key rollover for certificate renewal, follow these guidelines and restrictions:

- Trustpoints configured to generate a new key pair using the **regenerate** command or the **regenerate** keyword of the **auto-enroll** command must not share key pairs with other trustpoints. To give each trustpoint its own key pair, use the **rsakeypair** command in ca-trustpoint configuration mode. Sharing key pairs among regenerating trustpoints is not supported and will cause loss of service on some of the trustpoints because of key and certificate mismatch.

Configuring Automatic Certificate Enrollment with Key Rollover

To configure key rollover with automatic certificate enrollment, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ` **`crypto pki trustpoint`** *`name`*	Declares the name of the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)# ` **`enrollment url`** *`url`*	Specifies the URL of the CA on which your switch should send certificate requests; for example, http://ca_server. • *url*—Must be in the form of http://CA_name, where CA_name is the name of the CA's host Domain Name System or the IP address.
Step 3	`Router(ca-trustpoint)# ` **`subject-name`** `[`*`x.500-name`*`]`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—(Optional) If the *x-500-name* argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.
Step 4	`Router(ca-trustpoint)# ` **`ip-address`** `{`*`interface`* `|` **`none`**`}`	Includes the IP address of the specified interface in the certificate request. • *interface*—IP address of the interface. • **none**—Specify this keyword if no IP address should be included. If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 5	`Router(ca-trustpoint)# ` **`serial-number`** `[`**`none`**`]`	Specifies the switch serial number in the certificate request, unless the **none** keyword is specified. • **none**—(Optional) Specify this keyword if no serial number should be included.

	Command	Purpose
Step 6	`Router(ca-trustpoint)# `**`auto-enroll`** [*`percent`*][**`regenerate`**]	Enables autoenrollment. This command allows you to automatically request a switch certificate from the CA. By default, only the DNS name of the switch is included in the certificate. • *percent*—Use the *percent* argument to specify that a new certificate will be requested after the percent lifetime of the current certificate is reached. • **regenerate**—Specify this keyword to generate a new key for the certificate even if a named key already exists. **Note** If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: ! RSA key pair associated with trustpoint is exportable.
Step 7	`Router(ca-trustpoint)# `**`password`**` `*`string`*	(Optional) Specifies the revocation password for the certificate. • *string*—Text of the password. **Note** If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.
Step 8	`Router(ca-trustpoint)# `**`rsakeypair`**` `*`key-label`* [*`key-size`*` [`*`encryption-key-size`*`]]`	Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the e*ncryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) **Note** If this command is not enabled, the FQDN key pair is used.
Step 9	`Router(ca-trustpoint)# `**`exit`**	Exits ca-trustpoint configuration mode and returns to global configuration mode.

	Command	Purpose
Step 10	Router(config)# **crypto pki authenticate** *name*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1. Check the certificate fingerprint if prompted. **Note** This command is optional if the CA certificate is already loaded into the configuration.
Step 11	Router(config)# **exit**	Exits global configuration mode and returns to privileged EXEC mode.
Step 12	Router# **copy system:running-config nvram:startup-config**	(Optional) Copies the running configuration to the NVRAM startup configuration.

Configuring Manual Certificate Enrollment with Key Rollover

Note Do not regenerate the keys manually using the **crypto key generate** command; key rollover will occur when the **crypto pki enroll** command is entered.

To configure key rollover with manual certificate enrollment, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoint** *name*	Declares the name of the CA that your switch should use and enters ca-trustpoint configuration mode. • name—Name of the CA.
Step 2	Router(ca-trustpoint)# **enrollment url** *url*	Specifies the URL of the CA on which your switch should send certificate requests; for example, http://ca_server. • *url*—Must be in the form of http://CA_name,_ where CA_name is the name of the CA's host Domain Name System or the IP address.
Step 3	Router(ca-trustpoint)# **subject-name** [*x.500-name*]	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—If the *x-500-name* argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.

	Command	Purpose
Step 4	`Router(ca-trustpoint)# ip-address {interface | none}`	Includes the IP address of the specified interface in the certificate request. • *interface*—IP address of the interface. • **none**—Specify this keyword if no IP address should be included. If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 5	`Router(ca-trustpoint)# serial-number [none]`	Specifies the switch serial number in the certificate request, unless the **none** keyword is specified. • **none**—Specify this keyword if no serial number should be included.
Step 6	`Router(ca-trustpoint)# regenerate`	Enables key rollover with certificate enrollment when the **crypto pki enroll** command is entered. **Note** This command generates a new key for the certificate even if a named key already exists. Do not use the **crypto key generate** command with the key rollover feature. If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: ! RSA key pair associated with trustpoint is exportable.
Step 7	`Router(ca-trustpoint)# password string`	(Optional) Specifies the revocation password for the certificate. • *string*—Text of the password. **Note** If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.

	Command	Purpose
Step 8	`Router(ca-trustpoint)# rsakeypair key-label [key-size [encryption-key-size]]`	Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) **Note** If this command is not enabled, the FQDN key pair is used.
Step 9	`Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 10	`Router(config)# crypto pki authenticate name`	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1. Check the certificate fingerprint if prompted. **Note** This command is optional if the CA certificate is already loaded into the configuration.
Step 11	`Router(config)# crypto pki enroll name`	Requests certificates for all of your RSA key pairs. • *name*—Name of the CA. This command causes your switch to request as many certificates as there are RSA key pairs, so you need perform this command only once, even if you have special-usage RSA key pairs. When the **regenerate** configuration command is configured, this command will perform key rollover. **Note** This command requires you to create a challenge password that is not saved with the configuration. This password is required if your certificate needs to be revoked, so you must remember this password.
Step 12	`Router(config)# exit`	Exits global configuration mode.

For complete configuration information for key rollover for certificate renewal, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gtkyroll.html

For key rollover configuration examples, see the "Key Rollover for Certificate Renewal Configuration Examples" section on page 26-60.

Configuring PKI: Query Multiple Servers During Certificate Revocation Check

Before an X.509 certificate presented by a peer is validated, the certificate revocation list (CRL) is checked to make sure that the certificate has not been revoked by the issuing certification authority (CA). The certificate usually contains a certificate distribution point (CDP) in the form of a URL. Cisco IOS software uses the CDP to locate and retrieve the CRL.

Previous versions of Cisco IOS software make only one attempt to retrieve the CRL, even when the certificate contains more than one CDP. If the CDP server does not respond, the Cisco IOS software reports an error, which may result in the peer's certificate being rejected.

The PKI:query multiple servers during certificate revocation check feature provides the ability for Cisco IOS software to make multiple attempts to retrieve the CRL by trying all of the available CDPs in a certificate. This allows operations to continue when a particular server is not available. In addition, the ability to override the CDPs in a certificate with a manually configured CDP is also provided. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

To manually override the existing CDPs for a certificate with a URL or directory specification, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto pki trustpoin`**`t` *`name`*	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	`Router(ca-trustpoint)#` **`match certificate`** *`certificate-map-label`* **`override cdp`** `{`**`url`** `|` **`directory`**`}` *`string`*	Manually overrides the existing CDP entries for a certificate with a URL or directory specification. • *certificate-map-label*—A user-specified label that must match the label argument specified in a previously defined **crypto pki certificate map** command. • **url**—Specifies that the certificate's CDPs will be overridden with an HTTP or LDAP URL. • **directory**—Specifies that the certificate's CDPs will be overridden with an LDAP directory specification. • *string*—The URL or directory specification. Some applications may time out before all CDPs have been tried and will report an error message. This will not affect the switch, and the Cisco IOS software will continue attempting to retrieve a CRL until all CDPs have been tried.

For complete configuration information for the PKI: Query Multiple Servers During Certificate Revocation Check feature, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gtcertrc.html

For a query multiple servers configuration example, see the "PKI: Query Multiple Servers During Certificate Revocation Check (CDP Override) Configuration Example" section on page 26-61.

Configuring the Online Certificate Status Protocol

The Online Certificate Status Protocol (OCSP) feature allows users to enable OCSP instead of certificate revocation lists (CRLs) to check certificate status. Unlike CRLs, which provide only periodic certificate status, OCSP can provide timely information regarding the status of a certificate.

OCSP Configuration Guidelines and Restrictions

When configuring OCSP, follow these guidelines and restrictions:

- OCSP transports messages over HTTP, so there may be a time delay when you access the OCSP server. If the OCSP server is unavailable, certificate verification will fail.

- The increased certificate size may cause a problem for low-end switches when certificates are stored on NVRAM. Before you add the Authority Info Access (AIA) extension to a certificate, make sure that the increased size will not cause deployment problems.
- An OCSP server usually operates in either push or poll mode. You can configure a CA server to push revocation information to an OCSP server or configure an OCSP server to periodically download (poll) a CRL from the CA server. To ensure that timely certificate revocation status is obtained, you should carefully consider the push and poll interval.
- When configuring an OCSP server to return the revocation status for a CA server, the OCSP server must be configured with an OCSP response signing certificate that is issued by that CA server. Ensure that the signing certificate is in the correct format, or the switch will not accept the OCSP response. Refer to your OCSP manual for additional information.

To configure your switch for OCSP to check certificate status, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto pki trustpoint`** *`name`*	Declares the CA that your switch should use and puts you in ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	`Router(ca-trustpoint)#` **`ocsp url`** *`url`*	(Optional) Specifies the URL of an OCSP server so that the trustpoint can check the certificate status. This URL will override the URL of the OCSP server (if one exists) in the Authority Info Access (AIA) extension of the certificate. • *url* —Specifies the HTTP URL to be used.
Step 3	`Router(ca-trustpoint)#` **`revocation-check`** *`method1`* `[`*`method2`*`[`*`method3`*`]]`	Checks the revocation status of a certificate. • *method1* [*method2*[*method3*]]—Specifies the method used by the switch to check the revocation status of the certificate. Available methods are as follows: – **crl**—Certificate checking is performed by a CRL. This is the default option. – **none**—Certificate checking is ignored. – **ocsp**—Certificate checking is performed by an OCSP server. If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down.

Verifying the OCSP Configuration

To display information about your certificate and the CA certificate, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates

Certificate
```

```
Status: Available
Version: 3
Certificate Serial Number: 18C1EE03000000004CBD
Certificate Usage: General Purpose

Issuer:
  cn=msca-root
  ou=pki msca-root
  o=cisco
  l=santa cruz2
  st=CA
  c=US
  ea=user@example.com

Subject:
  Name: myrouter.example.com
  hostname=myrouter.example.com

CRL Distribution Points:
  http://msca-root/CertEnroll/msca-root.crl

Validity Date:
  start date: 19:50:40 GMT Oct 5 2004
  end   date: 20:00:40 GMT Oct 12 2004

  Subject Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (360 bit)

  Signature Algorithm: SHA1 with RSA Encryption
  Fingerprint MD5: 2B5F53E6 E3E892E6 3A9D3706 01261F10
  Fingerprint SHA1: 315D127C 3AD34010 40CE7F3A 988BBDA5 CD528824

  X509v3 extensions:
  X509v3 Key Usage: A0000000
  Digital Signature
  Key Encipherment
  X509v3 Subject Key ID: D156E92F 46739CBA DFE66D2D 3559483E B41ECCF4
  X509v3 Authority Key ID: 37F3CC61 AF5E7C0B 434AB364 CF9FA0C1 B17C50D9
  Authority Info Access:

  Associated Trustpoints: msca-root
  Key Label: myrouter.example.com

 CA Certificate

  Status: Available
  Version: 3
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature

  Issuer:
  cn=msca-root
  ou=pki msca-root
  o=cisco
  l=santa cruz2
  st=CA
  c=US
  ea=user@example.com

  Subject:
  cn=msca-root
  ou=pki msca-root
  o=cisco
```

```
    l=santa cruz2
    st=CA
    c=US
    ea=user@example.com

    CRL Distribution Points:
    http://msca-root.example.com/CertEnroll/msca-root.crl

    Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017

    Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (512 bit)

    Signature Algorithm: SHA1 with RSA Encryption
    Fingerprint MD5: 84E470A2 38176CB1 AA0476B9 C0B4F478
    Fingerprint SHA1: 0F57170C 654A5D7D 10973553 EFB0F94F 2FAF9837

    X509v3 extensions:
    X509v3 Key Usage: C6000000
    Digital Signature
    Non Repudiation
    Key Cert Sign
    CRL Signature

  X509v3 Subject Key ID: 37F3CC61 AF5E7C0B 434AB364 CF9FA0C1 B17C50D9
  X509v3 Basic Constraints:
    CA: TRUE

  Authority Info Access:
  Associated Trustpoints: msca-root
```

To display the trustpoints and configured trustpoint subcommands that are configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:
    Subject Name:
    CN = bomborra Certificate Manager
    O = cisco.com
    C = US
    Serial Number:01
    Certificate configured.
    CEP URL:http://bomborra
    CRL query url:ldap://bomborra
```

For complete configuration information for OCSP, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t2/feature/guide/gt_ocsp.html

For OCSP configuration examples, see the "Online Certificate Status Protocol Configuration Examples" section on page 26-61.

Configuring Certificate Security Attribute-Based Access Control

Under the IPsec protocol, certificate authority (CA) interoperability permits Cisco IOS devices and a CA to communicate so that the Cisco IOS device can obtain and use digital certificates from the CA. Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action. The certificate security attribute-based access control feature adds fields to the certificate to create a certificate-based ACL.

Certificate Security Attribute-Based Access Control Configuration Guidelines and Restrictions

When configuring certificate security attribute-based access control, follow these guidelines and restrictions:

- The certificate-based ACL specifies one or more fields within the certificate and an acceptable value for each specified field. You can specify which fields within a certificate should be checked and which values those fields may or may not have. There are six logical tests for comparing the field with the value: equal, not equal, contains, does not contain, less than, and greater than or equal.
- If more than one field is specified within a single certificate-based ACL, the tests of all of the fields within the ACL must succeed to match the ACL.
- The same field can be specified multiple times within the same ACL.
- More than one ACL can be specified. Each ACL will be processed in turn until a match is found or all of the ACLs have been processed.
- Memory is required to hold the ACLs as they are created and as they are loaded from the configuration file. The amount of memory depends on which fields within the certificate are being checked and how many ACLs have been defined. Certificate-based ACL support requires one or more compare operations when the fields in a certificate are being checked. Only the fields specified by the ACL are checked. The compare operations are a small part of certificate validation and will not have a noticeable effect on switch performance when validating certificates.

To configure Certificate Security Attribute-Based Access Control, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki certificate map** *label sequence-number*	Starts ca-certificate-map mode and defines certificate-based ACLs by assigning a label for the ACL that will also be referenced within the **crypto pki trustpoint** command. • *label*—An arbitrary string that identifies the ACL. • *sequence-number*—A sequence number that orders ACLs with the same label.
Step 2	Router(ca-certificate-map)# *field-name match-criteria match-value*	In ca-certificate-map mode, you specify one or more certificate fields together with their matching criteria and the value to match. • *field-name*—Specifies one of the following case-insensitive name strings or a date: – **subject-name** – **issuer-name** – **unstructured-subject-name** – **alt-subject-name** – **name** – **valid-start** – **expires-on** **Note** Date field format is *dd mm yyyy hh*:*mm*:*ss* or *mmm dd yyyy hh*:*mm*:*ss*. • *match-criteria*—Specifies one of the following logical operators: – **eq**—Equal (valid for name and date fields) – **ne**—Not equal (valid for name and date fields) – **co**—Contains (valid only for name fields) – **nc**—Does not contain (valid only for name fields) – **lt** —Less than (valid only for date fields) – **ge** —Greater than or equal (valid only for date fields) • *match-value*—Specifies the name or date to test with the logical operator assigned by *match-criteria*. For example: Router(ca-certificate-map)# **subject-name co Cisco**
Step 3	Router(ca-certificate-map)# **exit**	Exits ca-certificate-map mode.

	Command	Purpose
Step 4	`Router(config)# crypto pki trustpoint name`	Starts ca-trustpoint configuration mode and creates a name for the CA. • *name*—Specifies a name for the CA.
Step 5	`Router(ca-trustpoint)# match certificate certificate-map-label`	Associates the certificate-based ACL defined with the **crypto pki certificate map** command to the trustpoint. • *certificate-map-label*—Specifies the label argument specified in the previously defined **crypto pki certificate map** command in Step 1.
Step 6	`Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.

Verifying Certificate-Based ACLs

To verify the certificate-based ACL configuration, enter the **show crypto pki certificates** command. The following example shows the components of the certificates (CA and switch certificate) installed on the switch when the switch has both authenticated and enrolled with a trustpoint:

```
Router# show crypto pki certificates

CA Certificate
   Status: Available
   Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
   Certificate Usage: Signature

   Issuer:
    CN = new-user
    OU = pki new-user
    O = cisco
    L = santa cruz2
    ST = CA
    C = US
    EA = user@cysco.net

    Subject:
    CN = new-user
    OU = pki new-user
    O = cisco
    L = santa cruz2
    ST = CA
    C = US
    EA = user@cysco.net

    CRL Distribution Point:
    http://new-user.cysco.net/CertEnroll/new-user.crl

    Validity Date:
    start date: 14:19:29 PST Oct 31 2002
    end date: 14:27:27 PST Oct 31 2017

    Associated Trustpoints: MS


Certificate
  Status: Available
  Certificate Serial Number: 193E28D20000000009F7
  Certificate Usage: Signature

  Issuer:
     CN = new-user
     OU = pki new-user
     O = cisco
     L = santa cruz2
     ST = CA
    C = US
     EA = user@cysco.net

  Subject:
    Name: User1.Cysco.Net
    OID.1.2.840.113549.1.9.2 = User1.Cysco.Net

  CRL Distribution Point:
    http://new-user.cysco.net/CertEnroll/new-user.crl

  Validity Date:
    start date: 12:40:14 PST Feb 26 2003
    end   date: 12:50:14 PST Mar 5 2003
    renew date: 16:00:00 PST Dec 31 1969

  Associated Trustpoints: MS
```

For complete configuration information for Certificate Security Attribute-Based Access Control, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ftcrtacl.html

For a certificate-based ACL example, see the "Certificate Security Attribute-Based Access Control Configuration Example" section on page 26-62.

Configuring PKI AAA Authorization Using the Entire Subject Name

When using public key infrastructure (PKI) and authentication, authorization, and accounting (AAA) functionality, users sometimes have attribute-value (AV) pairs that are different from those of every other user. As a result, a unique username is required for each user. The PKI AAA authorization using the entire subject name feature provides users with the ability to query the AAA server using the entire subject name from the certificate as a unique AAA username.

PKI AAA Authorization Using the Entire Subject Name Configuration Guidelines and Restrictions

When configuring PKI AAA authorization using the entire subject name, follow these guidelines and restrictions:

- Some AAA servers limit the length of the username (for example, to 64 characters). As a result, the entire certificate subject name cannot be longer than the limitation of the server.
- Some AAA servers limit the available character set that may be used for the username (for example, a space [ ] and an equal sign [=] may not be acceptable). This feature will not work for the AAA server having such a character-set limitation.

- The **subject-name** command in the trustpoint configuration might not always be the final AAA subject name. If the fully qualified domain name (FQDN), serial number, or IP address of the switch are included in a certificate request, the subject name field of the issued certificate will also have these components. To turn off the components, use the **fqdn**, **serial-number**, and **ip-address** commands with the **none** keyword.
- Certificate authority (CA) servers sometimes change the requested subject name field when they issue a certificate. For example, CA servers of some vendors switch the relative distinguished names (RDNs) in the requested subject names to the following order: CN, OU, O, L, ST, and C. However, another CA server might append the configured Lightweight Directory Access Protocol (LDAP) directory root (for example, O=cisco.com) to the end of the requested subject name.
- Depending on the tools you choose for displaying a certificate, the printed order of the RDNs in the subject name could be different. Cisco IOS software always displays the least significant RDN first, but other software, such as Open Source Secure Socket Layer (OpenSSL), does the opposite. Therefore, if you are configuring the AAA server with a full DN (subject name) as the corresponding username, ensure that the Cisco IOS software style (that is, with the least-significant RDN first) is used.

To configure the entire certificate subject name for PKI authentication, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the AAA access control model.
Step 2	`Router config)# aaa authorization network listname [method]`	Sets the parameters that restrict user access to a network. • *listname*—Character string used to name the list of authorization methods. • *method*—(Optional) Specifies an authorization method to be used for authorization. The *method* argument can be **group radius**, **group tacacs+**, or **group** *group-name*.
Step 3	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 4	`Router(ca-trustpoint)# enrollment url url`	Specifies the enrollment parameters of your CA. • *url*—The *url* argument is the URL of the CA to which your switch should send certificate requests.
Step 5	`Router(ca-trustpoint)# revocation-check method`	(Optional) Checks the revocation status of a certificate. • *method*—Method used by the switch to check the revocation status. Available methods are **ocsp**, **none**, and **crl**.
Step 6	`Router(ca-trustpoint)# exit`	Exits ca-truspoint configuration mode and enters global configuration mode.
Step 7	`Router(config)# authorization list {listname}`	Specifies the AAA authorization list. • *listname*—Name of the list.

	Command	Purpose
Step 8	`Router(config)# authorization username subjectname all`	Sets parameters for the different certificate fields that are used to build the AAA username. • **all**—Specifies that the entire subject name of the certificate will be used as the authorization username.
Step 9	`Router(config)# tacacs-server host hostname [key string]` or	Specifies a TACACS+ host. • *hostname*—Name of the host. • **key** *string*—(Optional) Character string specifying authentication and encryption key.
	`Router(config)# radius-server host hostname [key string]`	Specifies a RADIUS host.

For complete configuration information for the PKI AAA authorization using the entire subject name feature, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t11/feature/guide/gt_dnall.html

For a PKI AAA Authorization Using the Entire Subject Name configuration example, see the "PKI AAA Authorization Using the Entire Subject Name Configuration Example" section on page 26-62.

Configuring Source Interface Selection for Outgoing Traffic with Certificate Authority

The source interface selection for outgoing traffic with certificate authority feature allows you to specify that the address of an interface be used as the source address for all outgoing TCP connections associated with that trustpoint when a designated trustpoint has been configured.

To configure the interface that you want to use as the source address for all outgoing TCP connections associated with a trustpoint, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	`Router(ca-trustpoint)# enrollment url url`	Specifies the enrollment parameters of your CA. • *url*—Specifies the URL of the CA where your switch should send certificate requests; for example, http://ca_server. *url* must be in the form http://CA_*name*, where CA_*name* is the CA's host Domain Name System (DNS) name or IP address.
Step 3	`Router(ca-trustpoint)# source interface interface-address`	Specifies the interface to be used as the source address for all outgoing TCP connections associated with that trustpoint. • *interface-address*—Interface address.

	Command	Purpose
Step 4	Router(config)# **interface** *type slot*/[subslot]/*port*	Configures an interface type and enters interface configuration mode. • *type***—**Type of interface being configured. • *slot*/[subslot]/ *port*—Number of the slot, subslot (optional), and port to be configured.
Step 5	Router(config-if)# **description** *string*	Adds a description to an interface configuration. • *string*—Descriptive string.
Step 6	Router(config-if)# **ip address** *ip-address mask*	Sets a primary or secondary IP address for an interface. • *ip-address*—IP address. • *mask*—Subnet mask.
Step 7	Router(config-if)# **interface** *type slot*/[*subslot*]/*port*	Configures an interface type. • *type***—**Type of interface being configured. • *slot*/[*subslot*]/ *port*—Number of the slot, subslot (optional), and port to be configured.
Step 8	Router(config-if)# **description** *string*	Adds a description to an interface configuration. • *string*—Descriptive string.
Step 9	Router(config-if)# **ip address** *ip-address mask* [*secondary*]	Sets a primary or secondary IP address for an interface. • *ip-address*—IP address. • *mask*—Subnet mask. • [*secondary*]—(Optional) Secondary address.
Step 10	Router(config-if)# **crypto map** *map-name*	Applies a previously defined crypto map set to the interface. • *map-name*—Name that identifies the crypto map set.

For complete configuration information for source interface selection for outgoing traffic with certificate authority, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ft_asish.html

For a source interface selection configuration example, see the "Source Interface Selection for Outgoing Traffic with Certificate Authority Configuration Example" section on page 26-63.

Configuring Persistent Self-Signed Certificates

The persistent self-signed certificates feature saves a certificate generated by a Secure HTTP (HTTPS) server for the Secure Sockets Layer (SSL) handshake in a router's startup configuration.

Note The persistent self-signed certificates feature is supported in Cisco IOS Release 12.2(33)SXH and later releases.

Cisco IOS software has an HTTPS server that allows access to web-based management pages using a secure SSL connection. SSL requires the server to have an X.509 certificate that is sent to the client (web browser) during the SSL handshake to establish a secure connection between the server and the client.

The client expects the SSL server's certificate to be verifiable using a certificate the client already possesses.

If Cisco IOS software does not have a certificate that the HTTPS server can use, the server generates a self-signed certificate by calling a public key infrastructure (PKI) application programming interface (API). When the client receives this self-signed certificate and is unable to verify it, intervention is needed. The client asks you if the certificate should be accepted and saved for future use. If you accept the certificate, the SSL handshake continues.

Future SSL handshakes between the same client and the server use the same certificate. However, if the router is reloaded, the self-signed certificate is lost. The HTTPS server must then create a new self-signed certificate. This new self-signed certificate does not match the previous certificate, so you are once again asked to accept it.

Requesting acceptance of the router's certificate each time that the router reloads can be annoying and may present an opportunity for an attacker to substitute an unauthorized certificate during the time that you are being asked to accept the certificate.

The persistent self-signed certificates feature overcomes all these limitations by saving a certificate in the router's startup configuration, resulting in the following benefits:

- Having a persistent self-signed certificate stored in the router's startup configuration (NVRAM) lessens the opportunity for an attacker to substitute an unauthorized certificate because the browser is able to compare the certificate offered by the router with the previously saved certificate and warn you if the certificate has changed.
- Having a persistent self-signed certificate stored in the router's startup configuration eliminates the user intervention that is necessary to accept the certificate every time that the router reloads.
- Because user intervention is no longer necessary to accept the certificate, the secure connection process is faster.

Persistent Self-Signed Certificates Configuration Guidelines and Restrictions

When configuring persistent self-signed certificates, follow these guidelines and restrictions:

- You must load an image that supports SSL.
- You can configure only one trustpoint for a persistent self-signed certificate.

Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters

Note This section is optional because if you enable the Secure HTTP (HTTPS) server, it generates a self-signed certificate automatically using default values. To specify parameters, you must create a trustpoint and configure it. To use default values, delete any existing self-signed trustpoints. Deleting all self-signed trustpoints causes the HTTPS server to generate a persistent self-signed certificate using default values as soon as it is enabled.

To configure a trustpoint and specify self-signed certificate parameters, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the certificate authority (CA) that your router should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)# enrollment selfsigned`	Specifies self-signed enrollment.
Step 3	`Router(ca-trustpoint)# subject-name [x.500-name]`	(Optional) Specifies the requested subject name to be used in the certificate request. • *x.500-name*—(Optional) If the x.500-name argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.
Step 4	`Router(ca-trustpoint)# rsakeypair key-label [key-size [encryption-key-size]]`	(Optional) Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) **Note** If this command is not enabled, the FQDN key pair is used.
Step 5	`Router(ca-trustpoint)# crypto pki enroll trustpoint-name`	Tells the router to generate the persistent self-signed certificate. • *trustpoint-name*—Name of the CA.
Step 6	`Router(ca-trustpoint)# end`	(Optional) Exits ca-trustpoint configuration mode.

Enabling the HTTPS Server

To enable the HTTPS server, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip http secure-server`	Enables the secure HTTP web server. **Note** A key pair (modulus 1024) and a certificate are generated.
Step 2	`Router(config)# end`	Exits global configuration mode.

Note You must enter a **write memory** command to save the configuration. This command also saves the self-signed certificate and the HTTPS server in enabled mode.

Verifying the Persistent Self-Signed Certificate Configuration

To verify that a self-signed certificate and a trustpoint have been created, use the **show crypto pki certificates**, **show crypto mypubkey rsa,** and the **show crypto pki trustpoints** commands.

The **show crypto pki certificates** command displays information about your certificate, the CA certificate, and any registration authority certificates:

```
Router# show crypto pki certificates

Router Self-Signed Certificate
   Status: Available
   Certificate Serial Number: 01
   Certificate Usage: General Purpose
   Issuer:
    cn=IOS-Self-Signed-Certificate-3326000105
   Subject:
    Name: IOS-Self-Signed-Certificate-3326000105
    cn=IOS-Self-Signed-Certificate-3326000105
   Validity Date:
    start date: 19:14:14 GMT Dec 21 2004
    end   date: 00:00:00 GMT Jan 1 2020
   Associated Trustpoints: TP-self-signed-3326000105
```

Note The number 3326000105 above is the router’s serial number and varies depending on the router’s actual serial number.

The **show crypto mypubkey rsa** command displays information about the key pair corresponding to the self-signed certificate:

```
Router# show crypto mypubkey rsa

% Key pair was generated at: 19:14:10 GMT Dec 21 2004
Key name: TP-self-signed-3326000105
  Usage: General Purpose Key
  Key is not exportable.
  Key Data:
    30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B88F70
    6BC78B6D 67D6CFF3 135C1D91 8F360292 CA44A032 5AC1A8FD 095E4865 F8C95A2B
    BFD1C2B7 E64A3804 9BBD7326 207BD456 19BAB78B D075E78E 00D2560C B09289AE
    6DECB8B0 6672FB3A 5CDAEE92 9D4C4F71 F3BCB269 214F6293 4BA8FABF 9486BCFC
    2B941BCA 550999A7 2EFE12A5 6B7B669A 2D88AB77 39B38E0E AA23CB8C B7020301 0001
% Key pair was generated at: 19:14:13 GMT Dec 21 2004
Key name: TP-self-signed-3326000105.server
  Usage: Encryption Key
  Key is not exportable.
  Key Data:
    307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00C5680E 89777B42
    463E5783 FE96EA9E F446DC7B 70499AF3 EA266651 56EE29F4 5B003D93 2FC9F81D
    8A46E12F 3FBAC2F3 046ED9DD C5F27C20 1BBA6B9B 08F16E45 C34D6337 F863D605
    34E30F0E B4921BC5 DAC9EBBA 50C54AA0 BF551BDD 88453F50 61020301 0001
```

Note The second key pair with the name TP-self-signed-3326000105.server is the SSH key pair and is generated once any key pair is created on the router and SSH starts up.

The **show crypto pki trustpoints** command displays the trustpoints that are configured in the router:

```
Router# show crypto pki trustpoints

Trustpoint local:
    Subject Name:
    serialNumber=C63EBBE9+ipaddress=10.3.0.18+hostname=test.cisco.com
        Serial Number: 01
    Persistent self-signed certificate trust point
```

For complete configuration information for persistent self-signed certificates, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gtpsscer.html

For persistent self-signed certificates configuration examples, see the "Persistent Self-Signed Certificates Configuration Examples" section on page 26-63.

Configuration Examples

This section provides examples of the following configurations:

- Multiple RSA Key Pairs Configuration Example, page 26-54
- Protected Private Key Storage Configuration Examples, page 26-54
- Trustpoint CA Configuration Example, page 26-54
- Query Mode Definition Per Trustpoint Configuration Example, page 26-55
- Local Certificate Storage Location Configuration Example, page 26-55
- Direct HTTP Enrollment with CA Servers Configuration Examples, page 26-55
- Manual Certificate Enrollment Configuration Examples, page 26-57
- Certificate Autoenrollment Configuration Example, page 26-60
- Key Rollover for Certificate Renewal Configuration Examples, page 26-60
- PKI: Query Multiple Servers During Certificate Revocation Check (CDP Override) Configuration Example, page 26-61
- Online Certificate Status Protocol Configuration Examples, page 26-61
- Certificate Security Attribute-Based Access Control Configuration Example, page 26-62

- PKI AAA Authorization Using the Entire Subject Name Configuration Example, page 26-62
- Source Interface Selection for Outgoing Traffic with Certificate Authority Configuration Example, page 26-63
- Persistent Self-Signed Certificates Configuration Examples, page 26-63

Note The following examples use commands at the level of Cisco IOS Release 12.2(33)SXH.

As of Cisco IOS Release 12.2(33)SXH, the **crypto engine subslot** command used in previous releases has been replaced with the **crypto engine slot** command (of the form **crypto engine slot** *slot* {**inside** | **outside**}). The **crypto engine subslot** command is no longer supported. When upgrading, ensure that this command has been modified in your start-up configuration to avoid extended maintenance time.

Multiple RSA Key Pairs Configuration Example

The following example is a sample trustpoint configuration that specifies the RSA key pair exampleCAkeys:

```
Router(config)# crypto key generate rsa general-keys label exampleCAkeys
Router(config)# crypto pki trustpoint exampleCAkeys
Router(ca-trustpoint)# enrollment url http://exampleCAkeys/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# rsakeypair exampleCAkeys 1024 1024
```

Protected Private Key Storage Configuration Examples

This section contains the following configuration examples:

- Encrypted Key Configuration Example, page 26-54
- Locked Key Configuration Example, page 26-54

Encrypted Key Configuration Example

The following example shows how to encrypt the pki1-72a.cisco.com RSA key:

```
Router(config)# crypto key encrypt rsa name pki1-72a.cisco.com passphrase cisco1234
Router(config)# exit
```

Locked Key Configuration Example

The following example shows how to lock the pki1-72a.cisco.com key:

```
Router# crypto key lock rsa name pki1-72a.cisco.com passphrase cisco1234
```

Trustpoint CA Configuration Example

The following example shows how to declare the CA named kahului and specify characteristics for the trustpoint CA:

```
Router(config)# crypto pki trustpoint kahului
Router(ca-trustpoint)# enrollment url http://kahului
Router(ca-trustpoint)# crl query ldap://kahului
```

Query Mode Definition Per Trustpoint Configuration Example

The following configuration example shows a trustpoint CA that uses query mode:

```
Router(config)# crypto pki trustpoint trustpoint1
Router(ca-trustpoint)# enrollment url http://ca-server1
Router(ca-trustpoint)# crl query http://ca-server1
Router(ca-trustpoint)# query certificate
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate trustpoint1
Router(config)# crypto key generate rsa
Router(config)# crypto pki enroll trustpoint1
```

Local Certificate Storage Location Configuration Example

The following example shows how to store certificates to the certs subdirectory. Note that the certs subdirectory does not exist and is automatically created.

```
Router(config)# crypto pki certificate storage disk0:/certs
Requested directory does not exist -- created
Certificates will be stored in disk0:/certs/
Router(config)# end
Router# write
*May 27 02:09:00:%SYS-5-CONFIG_I:Configured from console by consolemem
Building configuration...
[OK]
Router# directory disk0:/certs
Directory of disk0:/certs/
14  -rw-         707  May 27 2005 02:09:02 +00:00  ioscaroot#7401CA.cer
15  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#826E.cer
16  -rw-         759  May 27 2005 02:09:02 +00:00  msca-root#1BA8CA.cer
17  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#75B8.cer
18  -rw-        1149  May 27 2005 02:09:02 +00:00  storagename#6500CA.cer
19  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#83EE.cer
47894528 bytes total (20934656 bytes free)

! The certificate files are now on disk0/certs:
```

Direct HTTP Enrollment with CA Servers Configuration Examples

This section provides the following configuration examples:

- Enrollment Profile for a Client Switch Configuration Example, page 26-55
- Enrollment Profile for a Client Switch Already Enrolled with a Third-Party Vendor CA Example, page 26-56
- Certificate Server Automatically Accepting Enrollment Requests Only from the Client Switch Configuration Example, page 26-56

Enrollment Profile for a Client Switch Configuration Example

The following example show how to configure an enrollment profile for direct HTTP enrollment with a CA server:

```
Router(config)# crypto pki trustpoint Entrust
Router(ca-trustpoint)# enrollment profile E
Router(ca-trustpoint)# exit
Router(config)# crypto pki profile enrollment E
Router(ca-profile-enroll)# authentication url  http://entrust:81
Router(ca-profile-enroll)# authentication command  GET /certs/cacert.der
Router(ca-profile-enroll)# enrollment url  http://entrust:81/cda-cgi/clientcgi.exe
Router(ca-profile-enroll)# enrollment command  POST reference_number=$P2&authcode=$P1
&retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
Router(ca-profile-enroll)# parameter 1 value aaaa-bbbb-cccc
Router(ca-profile-enroll)# parameter 2 value 5001

Router(config)# crypto ca profile enrollment E
Router(ca-profile-enroll)# authentication url http://ca-server.example.com
Router(ca-profile-enroll)# authentication command GET $P1
Router(ca-profile-enroll)# enrollment url
Router(ca-profile-enroll)# enrollment command ^C
POST action=getServerCert
&pkcs10Request=$REQ
&reference_number=$P2
&authcode=P3
&retrievedAs=rawDER
^C
Router(ca-profile-enroll)# parameter 1 value cacert.crt
Router(ca-profile-enroll)# parameter 2 prompt Enter the Reference Number:
Router(ca-profile-enroll)# parameter 3 prompt Enter the Auth Code:
```

Enrollment Profile for a Client Switch Already Enrolled with a Third-Party Vendor CA Example

The following example shows how to configure the following tasks on the client switch:

- Define the msca-root trustpoint that points to the third-party vendor CA and enroll and authenticate the client with the third-party vendor CA.
- Define cs trustpoint for the Cisco IOS CA.
- Define enrollment profile “cs1,” which points to Cisco IOS CA, and mention (via the enrollment credential command) that msca-root is being initially enrolled with the Cisco IOS CA.

```
! Define trustpoint "msca-root" for non-Cisco IOS CA.
Router(config)# crypto pki trustpoint msca-root
Router(ca-trustpoint)# enrollment mode ra
Router(ca-trustpoint)# enrollment url http://msca-root:80/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# ip-address FastEthernet2/0
Router(ca-trustpoint)# revocation-check crl

! Configure trustpoint "cs" for Cisco IOS CA.
Router(config)# crypto pki trustpoint cs
Router(ca-trustpoint)# enrollment profile cs1
Router(ca-trustpoint)# revocation-check crl

! Define enrollment profile "cs1."
Router(config)# crypto pki profile enrollment cs1
Router(ca-profile-enroll)# enrollment url  http://cs:80
Router(ca-profile-enroll)# enrollment credential  msca-root
```

Certificate Server Automatically Accepting Enrollment Requests Only from the Client Switch Configuration Example

The following example shows how to configure the certificate server, and enter the **grant auto trustpoint** command to instruct the certificate server to accept enrollment requests only from clients who are already enrolled with msca-root trustpoint:

```
Router(config)# crypto pki server cs
Router(cs-server)# database level minimum
Router(cs-server)# database url nvram:
Router(cs-server)# issuer-name CN=cs
Router(cs-server)# grant auto trustpoint msca-root

Router(config)# crypto pki trustpoint cs
Router(ca-trustpoint)# revocation-check crl
Router(ca-trustpoint)# rsakeypair cs

Router(ca-trustpoint)# crypto pki trustpoint msca-root
Router(ca-trustpoint)# enrollment mode ra
Router(ca-trustpoint)# enrollment url http://msca-root:80/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# revocation-check crl
```

Manual Certificate Enrollment Configuration Examples

This section provides the following manual certificate enrollment configuration examples:

- Manual Certificate Enrollment Using TFTP Configuration Example, page 26-57
- Manual Certificate Enrollment Using Cut-and-Paste Configuration Example, page 26-57

Manual Certificate Enrollment Using TFTP Configuration Example

The following example shows the configuration of manual certificate enrollment using TFTP:

```
Router(config)# crypto pki trustpoint MS
Router(ca-trustpoint)# enrollment url tftp://CA-Server/TFTPfiles/switch1
Router(ca-trustpoint)# crypto pki authenticate MS
Router(ca-trustpoint)# exit
Router(config)# crypto pki enroll MS
Router(config)# crypto pki import MS certificate
```

Manual Certificate Enrollment Using Cut-and-Paste Configuration Example

The following example shows how to configure manual cut-and-paste certificate enrollment. In this example, the name of the trustpoint CA is MS, and the **crypto pki import** command is entered twice because usage keys (signature and encryption keys) are used.

```
Router(config)# crypto pki trustpoint MS
Router(ca-trustpoint)# enrollment terminal
Router(ca-trustpoint)# crypto pki authenticate MS


Enter the base 64 encoded CA certificate.

End with a blank line or the word "quit" on a line by itself


-----BEGIN CERTIFICATE-----

MIICNDCCAd6gAwIBAgIQOsCmXpVHwodKryRoqULV7jANBgkqhkiG9w0BAQUFADA5
MQswCQYDVQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJ
bXNjYS1yb290MB4XDTAyMDIxNDAwNDYwMVoXDTA3MDIxNDAwNTQ0OFowOTELMAkG
A1UEBhMCVVMxFjAUBgNVBAoTDUNpc2NvIFN5c3RlbXMxEjAQBgNVBAMTCW1zY2Et
cm9vdDBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQCix8nIGFg+wvy3BjFbVi25wYoG
K2N0HWWHpqxFuFhqyBnIC0OshIn9CtrdN3JvUNHr0NIKocEwNKUGYmPwWGTfAgMB
AAGjgcEwgb4wCwYDVR0PBAQDAgHGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYE
FKIacsl6dKAfuNDVQymlSp7esf8jMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9t
```

```
c2NhLXJvb3QvQ2VydEVucm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8v
XFxtc2NhLXJvb3RcQ2VydEVucm9sbFxtc2NhLXJvb3QuY3JsMBAGCSsGAQQBgjcV
AQQDAgEAMA0GCSqGSIb3DQEBBQUAA0EAeuZkZMX9qkoLHfETYTpVWjZPQbBmwNRA
oJDSdYdtL3BcI/uLL5q7EmODyGfLyMGxuhQYx5r/40aSQgLCqBq+yg==

-----END CERTIFICATE-----


Certificate has the following attributes:

Fingerprint:D6C12961 CD78808A 4E02193C 0790082A

% Do you accept this certificate? [yes/no]:y

Trustpoint CA certificate accepted.

% Certificate successfully imported


Router(config)#

Router(config)# crypto pki enroll MS

% Start certificate enrollment..



% The subject name in the certificate will be:Router.cisco.com

% Include the router serial number in the subject name? [yes/no]:n

% Include an IP address in the subject name? [no]:n

Display Certificate Request to terminal? [yes/no]:y

Signature key certificate request -

Certificate Request follows:

MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAxdhXFDiWAn/hIZs9zfOtssKA
daoWYu0ms9Fe/Pew01dh14vXdxgacstOs2Pr5wk6jLOPxpvxOJPWyQM6ipLmyVxv
ojhyLTrVohrh6Dnqcvk+G/5ohss9o9RxvONwx042pQchFnx9EkMuZC7evwRxJEqR
mBHXBZ8GmP3jYQsjS8MCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgeAMA0GCSqGSIb3DQEBBAUAA4GBAMT6WtyFw95POY7UtF+YIYHiVRUf4SCq
hRIAGrljUePLo9iTqyPU1Pnt8JnIZ5P5BHU3MfgP8sqodaWub6mubkzaohJ1qD06
O87fnLCNid5Tov5jKogFHIki2EGGZxBosUw9lJlenQdNdDPbJc5LIWdfDvciA6jO
Nl8rOtKnt8Q+

---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]:

Encryption key certificate request -

Certificate Request follows:

MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwG60QojpDbzbKnyj8FyTiOcv
THkDP7XD4vLT1XaJ409z0gSIoGnIcdFtXhVlBWtpq3/O9zYFXr1tH+BMCRQi3Lts
0IpxYa3D9iFPqev7SPXpsAIsY8a6FMq7TiwLObqiQjLKL4cbuV0Frjl0Yuv5A/Z+
kqMOm7c+pWNWFdLe9lsCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgUgMA0GCSqGSIb3DQEBBAUAA4GBACF7feURj/fJMojPBlR6fa9BrlMJx+2F
H91YM/CIiz2n4mHTeWTWKhLoT8wUfa9NGOk7yi+nF/F7035twLfq6n2bSCTW4aem
```

```
8jLMMaeFxwkrV/ceQKrucmNC1uVx+fBy9rhnKx8j60XE25tnp1U08r6om/pBQABU
eNPFhozcaQ/2

---End - This line not part of the certificate request---


Redisplay enrollment request? [yes/no]:

n

Router(config)#crypto pki import MS certificate


Enter the base 64 encoded certificate.

End with a blank line or the word "quit" on a line by itself


MIIDajCCAxSgAwIBAgIKFN7C6QAAAAAMRzANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0MloXDTAzMDYwODAxMjY0MlowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMXYVxQ4lgJ/4SGbPc3zrbLCgHWqFmLtJrPRXvz3sNNXYdeL13cYGnLL
TrNj6+cJOoyzj8ab8TiT1skDOoqS5slcb6I4ci061aIa4eg56nL5Phv+aIbLPaPU
cbzjcMdONqUHIRZ8fRJDLmQu3r8EcSRKkZgR1wWfBpj942ELI0vDAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCB4AwHQYDVR0OBBYEFL8Quz8dyz4EGIeKx9A8UMNHLE4s
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAJo2
r6sHPGBdTQX2EDoJpR/A2UHXxRYqVSHkFKZw0z31r5JzUM0oPNUETV7mnZlYNVRZ
CSEX/G8boi3WOjz9wZo=

% Router Certificate successfully imported


Router(config)#

Router(config)# crypto pki import MS certificate


Enter the base 64 encoded certificate.

End with a blank line or the word "quit" on a line by itself


MIIDajCCAxSgAwIBAgIKFN7OBQAAAAAMSDANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0NVoXDTAzMDYwODAxMjY0NVowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMButEKI6Q282yp8o/Bck4jnL0x5Az+1w+Ly09V2ieNPc9IEiKBpyHHR
bV4VZQVraat/zvc2BV69bR/gTAkUIty7bNCKcWGtw/YhT6nr+0j16bACLGPGuhTK
u04sCzm6okIyyi+HG7ldBa45dGLr+QP2fpKjDpu3PqVjVhXS3vZbAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCBSAwHQYDVR0OBBYEFPDO29oRdlEUSgBMg6jZR+YFRWlj
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
```

```
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAHaU
hyCwLirUghNxCmLzXRG7C3W1j0kSX7a4fX9OxKR/Z2SoMjdMNPPyApuh8SoT2zBP
ZKjZU2WjcZG/nZF4W5k=

% Router Certificate successfully imported
```

Certificate Autoenrollment Configuration Example

The following example shows how to configure the switch to autoenroll with a CA on start-up:

```
Router(config)# crypto pki trustpoint frog
Router(ca-trustpoint)# enrollment url http://frog.phoobin.com/
Router(ca-trustpoint)# subject-name OU=Spiral Dept., O=tiedye.com
Router(ca-trustpoint)# ip-address ethernet-0
Router(ca-trustpoint)# auto-enroll regenerate
Router(ca-trustpoint)# password revokeme
Router(ca-trustpoint)# rsa-key frog 2048
!
Router(config)# crypto pki certificate chain frog
Router(config-cert-chain)# certificate ca 0B
30820293 3082023D A0030201 0202010B 300D0609 2A864886 F70D0101 04050030
79310B30 09060355 04061302 5553310B 30090603 55040813 02434131 15301306
0355040A 130C4369 73636F20 53797374 656D3120 301E0603 55040B13 17737562
6F726420 746F206B 6168756C 75692049 50495355 31243022 06035504 03131B79
6E692D75 31302043 65727469 66696361 7465204D 616E6167 6572301E 170D3030
30373134 32303536 32355A17 0D303130 37313430 31323834 335A3032 310E300C
06035504 0A130543 6973636F 3120301E 06092A86 4886F70D 01090216 11706B69
2D343562 2E636973 636F2E63 6F6D305C 300D0609 2A864886 F70D0101 01050003
4B003048 024100B3 0512A201 3B4243E1 378A9703 8AC5E3CE F77AF987 B5A422C4
15E947F6 70997393 70CF34D6 63A86B9C 4347A81A 0551FC02 ABA62360 01EF7DD2
6C136AEB 3C6C3902 03010001 A381F630 81F3300B 0603551D 0F040403 02052030
1C060355 1D110415 30138211 706B692D 3435622E 63697363 6F2E636F 6D301D06
03551D0E 04160414 247D9558 169B9A21 23D289CC 2DDA2A9A 4F77C616 301F0603
551D2304 18301680 14BD742C E892E819 1D551D91 683F6DB2 D8847A6C 73308185
0603551D 1F047E30 7C307AA0 3CA03AA4 38303631 0E300C06 0355040A 13054369
73636F31 24302206 03550403 131B796E 692D7531 30204365 72746966 69636174
65204D61 6E616765 72A23AA4 38303631 0E300C06 0355040A 13054369 73636F31
24302206 03550403 131B796E 692D7531 30204365 72746966 69636174 65204D61
6E616765 72300D06 092A8648 86F70D01 01040500 03410015 BC7CECF9 696697DF
E887007F 7A8DA24F 1ED5A785 C5C60452 47860061 0C18093D 08958A77 5737246B
0A25550A 25910E27 8B8B428E 32F8D948 3DD1784F 954C70
quit
```

Key Rollover for Certificate Renewal Configuration Examples

This section contains the following examples:

- Certificate Autoenrollment with Key Rollover Configuration Example, page 26-61
- Manual Certificate Enrollment with Key Rollover Configuration Example, page 26-61

Certificate Autoenrollment with Key Rollover Configuration Example

The following example shows how to configure the switch to autoenroll with the CA named trustme1 on startup. In this example, the **regenerate** keyword is specified, so a new key will be generated for the certificate. The renewal percentage is configured as 90 so if the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. The changes made to the running configuration are saved to the NVRAM startup configuration because autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

```
Router(config)# crypto pki trustpoint trustme1
Router(ca-trustpoint)# enrollment url http://trustme1.company.com/
Router(ca-trustpoint)# subject-name OU=Spiral Dept., O=tiedye.com
Router(ca-trustpoint)# ip-address ethernet0
Router(ca-trustpoint)# serial-number none
Router(ca-trustpoint)# auto-enroll 90 regenerate
Router(ca-trustpoint)# password revokeme
Router(ca-trustpoint)# rsakeypair trustme1 2048
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate trustme1
```

Manual Certificate Enrollment with Key Rollover Configuration Example

The following example shows how to configure key rollover to regenerate new keys with a manual certificate enrollment from the CA named trustme2:

```
Router(config)# crypto pki trustpoint trustme2
Router(ca-trustpoint)# enrollment url http://trustme2.company.com/
Router(ca-trustpoint)# subject-name OU=Spiral Dept., O=tiedye.com
Router(ca-trustpoint)# ip-address ethernet0
Router(ca-trustpoint)# serial-number none
Router(ca-trustpoint)# regenerate
Router(ca-trustpoint)# password revokeme
Router(ca-trustpoint)# rsakeypair trustme2 2048
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate trustme2
Router(config)# crypto pki enroll trustme2
Router(config)# exit
```

PKI: Query Multiple Servers During Certificate Revocation Check (CDP Override) Configuration Example

The following example uses the **match certificate override cdp** command to override the CDPs for the certificate map named Group1 defined in a **crypto pki certificate map** command:

```
Router(config)# crypto pki certificate map Group1 10
Router(ca-certificate-map)# subject-name co ou=WAN
Router(ca-certificate-map)# subject-name co o=Cisco
Router(config)# crypto pki trustpoint pki
Router(ca-trustpoint)# match certificate Group1 override cdp url http://server.cisco.com
```

Online Certificate Status Protocol Configuration Examples

This section provides the following configuration examples:

- OCSP Server Configuration Example, page 26-62
- CRL Then OCSP Server Configuration Example, page 26-62

- Specific OCSP Server Configuration Example, page 26-62

OCSP Server Configuration Example

The following example shows how to configure the switch to use the OCSP server that is specified in the AIA extension of the certificate:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check ocsp
```

CRL Then OCSP Server Configuration Example

The following example shows how to configure the switch to download the CRL from the certificate distribution point (CDP); if the CRL is unavailable, the OCSP server that is specified in the AIA extension of the certificate will be used. If both options fail, certificate verification will also fail.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check crl ocsp
```

Specific OCSP Server Configuration Example

The following example shows how to configure your switch to use the OCSP server at the HTTP URL http://myocspserver:81. If the server is down, revocation check will be ignored.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
```

Certificate Security Attribute-Based Access Control Configuration Example

The following example shows how to configure a certificate-based ACL:

```
Router(config)# crypto pki certificate map Group 10
Router(ca-certificate-map)# subject-name co Cisco
Router(config-cert-map)# exit
Router(config)# crypto pki trustpoint Access
Router(ca-trustpoint)# match certificate Group
Router(ca-trustpoint)# exit
```

PKI AAA Authorization Using the Entire Subject Name Configuration Example

The following example shows that the entire subject name of the certificate is to be used for PKI AAA authorization:

```
Router(config)# aaa new-model
Router(config)# aaa authorization network tac-o group tacacs+

Router(config)# crypto pki trustpoint test
Router(ca-trustpoint)# enrollment url http://caserver:80
Router(ca-trustpoint)# revocation-check crl
Router(ca-trustpoint)# exit
Router(ca-trustpoint)# authorization list tac-o
Router(ca-trustpoint)# authorization username subjectname all

Router(config)# tacacs-server host 20.2.2.2 key a_secret_key
```

Source Interface Selection for Outgoing Traffic with Certificate Authority Configuration Example

In the following example, the switch is located in a branch office. The switch uses IP Security (IPsec) to communicate with the main office. Ethernet 1 is the outside interface that connects to the Internet Service Provider (ISP). Ethernet 0 is the interface connected to the LAN of the branch office. To access the CA server located in the main office, the switch must send its IP datagrams out interface Ethernet 1 (address 10.2.2.205) using the IPsec tunnel. Address 10.2.2.205 is assigned by the ISP. Address 10.2.2.205 is not a part of the branch office or main office.

The CA cannot access any address outside the company because of a firewall. The CA sees a message coming from 10.2.2.205 and cannot respond (that is, the CA does not know that the switch is located in a branch office at address 10.1.1.1, which it is able to reach).

Adding the **source interface** command tells the switch to use address 10.1.1.1 as the source address of the IP datagram that it sends to the CA. The CA is able to respond to 10.1.1.1.

This example is configured using the **source interface** command and the interface addresses as described above.

```
Router(config)# crypto pki trustpoint ms-ca
Router(ca-trustpoint)# enrollment url http://ms-ca:80/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# source interface ethernet0

Router(config)# interface ethernet 0
Router(config-if)# description inside interface
Router(config-if)# ip address 10.1.1.1 255.255.255.0

Router(config)# interface ethernet 1
Router(config-if)# description outside interface
Router(config-if)# ip address 10.2.2.205 255.255.255.0
Router(config-if)# crypto map main-office
```

Persistent Self-Signed Certificates Configuration Examples

The following examples show how to configure a persistent self-signed certificate:

- Trustpoint and Self-Signed Certificate Configuration Example, page 26-63
- Enabling the HTTPS Server Configuration Example, page 26-64

Trustpoint and Self-Signed Certificate Configuration Example

The following example shows how to configure a trustpoint and a self-signed certificate. In this example, a trustpoint named local is declared, its enrollment is requested, and a self-signed certificate with an IP address is generated.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto pki trustpoint local
Router(ca-trustpoint)# enrollment selfsigned
Router(ca-trustpoint)# end
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto pki enroll local
Nov 29 20:51:13.067: %SSH-5-ENABLED: SSH 1.99 has been enabled
Nov 29 20:51:13.267: %CRYPTO-6-AUTOGEN: Generated new 512 bit key pair
```

```
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: yes
Enter Interface name or IP Address[]: ethernet 0
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created
```

Note A switch can have only one self-signed certificate. If you attempt to enroll a trustpoint configured for a self-signed certificate and one already exists, you receive a notification and are asked if you want to replace it. If so, a new self-signed certificate is generated to replace the existing one.

Enabling the HTTPS Server Configuration Example

In the following example, the HTTPS server is enabled and a default trustpoint is generated because one was not previously configured:

```
Router(config)# ip http secure-server

% Generating 1024 bit RSA keys ...[OK]

*Dec 21 19:14:15.421:%PKI-4-NOAUTOSAVE:Configuration was modified. Issue "write memory"
to save new certificate

Router(config)#
```

Note You must save the configuration to NVRAM if you want to keep the self-signed certificate and have the HTTPS server enabled following switch reloads.

The following message also appears:

```
*Dec 21 19:14:10.441:%SSH-5-ENABLED:SSH 1.99 has been enabled

Router(config)#
```

Note Creation of the key pair used with the self-signed certificate causes the Secure Shell (SSH) server to start. This behavior cannot be suppressed. You may want to modify your access control lists (ACLs) to permit or deny SSH access to the router.

CHAPTER 27

Configuring Advanced VPNs Using the IPsec VPN SPA

This chapter provides information about configuring advanced IPsec VPNs on the IPsec VPN SPA on the Catalyst 6500 Series switch. It includes the following sections:

Note The procedures in this chapter assume you have familiarity with security configuration concepts, such as VLANs, ISAKMP policies, preshared keys, transform sets, access control lists, and crypto maps. For more information about these and other security configuration concepts, refer to the following Cisco IOS documentation:

Cisco IOS Security Configuration Guide, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

Cisco IOS Security Command Reference, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For additional information about the commands used in this chapter, see the the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* and the related Cisco IOS Release 12.2 software configuration guide and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Tip To ensure a successful configuration of your VPN using the IPsec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Overview of Advanced VPNs

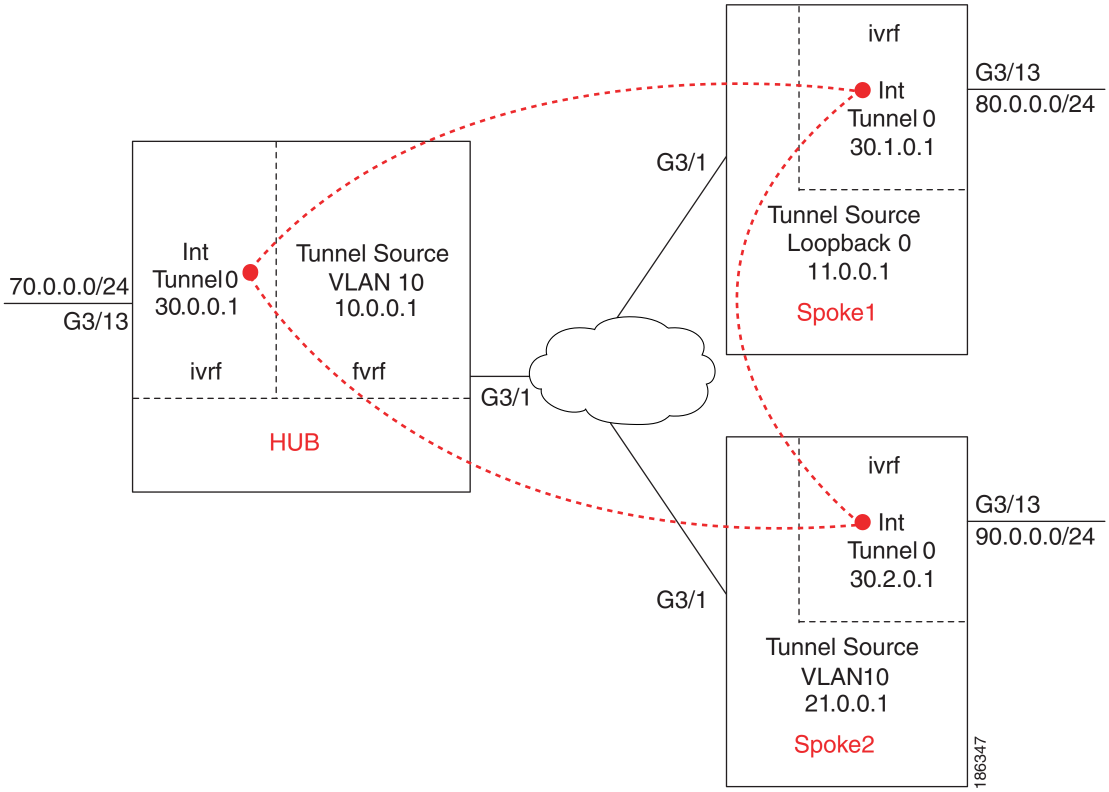
Configuring IP Security (IPsec) Virtual Private Networks (VPNs) in large, complicated networks can be quite complex. This chapter introduces Dynamic Multipoint VPN (DMVPN) and Easy VPN, two features that ease IPsec configuration in advanced environments.

Configuring DMVPN

The DMVPN feature allows users to better scale large and small IPsec VPNs by combining generic routing encapsulation (GRE) tunnels, IPsec encryption, and Next Hop Resolution Protocol (NHRP).

Figure 27-1 shows an example of a DMVPN configuration with a hub and two spokes.

Figure 27-1 DMVPN Configuration Example



DMVPN Configuration Guidelines and Restrictions

When configuring DMVPN, follow these guidelines and restrictions:

- A tunnel key should not be configured. If a tunnel key is configured, neither the PFC3 nor the IPsec VPN SPA will take over the tunnel and the tunnel will be switched in software.
- GRE tunnels in different Virtual Routing and Forwarding (VRF) instances cannot share the same tunnel source.

- In non-VRF mode, multipoint GRE tunnels should not share the same tunnel source.
- Multicast streaming is not supported across DMVPN on a Catalyst 6500 Series switch. Only multicast packets from a control plane such as routing protocols are supported.
- In a VRF-Aware DMVPN configuration, the **mls mpls tunnel-recir** command must be configured globally on the PE/hub if the CE/DMVPN spokes need to talk to other CEs across the MPLS cloud.
- For the NAT-transparency aware enhancement to work with DMVPN, you must use IPsec transport mode on the transform set. Also, even though NAT-transparency (IKE and IPsec) can support two peers (IKE and IPsec) being translated to the same IP address (using the User Datagram Protocol [UDP] ports to differentiate them [this would be Peer Address Translation]), this functionality is not supported for DMVPN. All DMVPN spokes must have a unique IP address after they have been NAT translated. They can have the same IP address before they are NAT translated.
- If you use the dynamic creation for spoke-to-spoke tunnels benefit of this feature, you must use IKE certificates or wildcard preshared keys for Internet Security Association and Key Management Protocol (ISAKMP) authentication.

Note We recommend that you do not use wildcard preshared keys because access to the entire VPN is compromised if one spoke switch is compromised.

- GRE tunnel keepalive (that is, the **keepalive** command under the GRE interface) is not supported on multipoint GRE tunnels
- FVRF is not supported on a multipoint GRE (mGRE) tunnel configured on a DMVPN spoke. FVRF is supported on an mGRE tunnel configured on a DMVPN hub.

To enable mGRE and IPsec tunneling for hub and spoke switches, configure your mGRE tunnel for IPsec encryption using the following procedures:

- DMVPN Prerequisites, page 27-3
- Configuring an IPsec Profile, page 27-4
- Configuring the Hub for DMVPN in VRF Mode, page 27-5
- Configuring the Hub for DMVPN in Crypto-Connect Mode, page 27-7
- Configuring the Spoke for DMVPN in VRF Mode, page 27-8
- Configuring the Spoke for DMVPN in Crypto-Connect Mode, page 27-10
- Verifying the DMVPN Configuration, page 27-12
- DMVPN Configuration Examples, page 27-18

For complete configuration information for DMVPN support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftgreips.html

DMVPN Prerequisites

Before configuring an IPsec profile, you must define a transform set by using the **crypto ipsec transform-set** command.

Configuring an IPsec Profile

The IPsec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands are valid in an IPsec profile. Only commands that pertain to an IPsec policy can be issued under an IPsec profile; you cannot specify the IPsec peer address or the access control list (ACL) to match the packets that are to be encrypted.

To configure an IPsec profile, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto ipsec profile`** *`name`*	Defines the IPsec parameters that are to be used for IPsec encryption between "spoke and hub" and "spoke and spoke" switches. This command enters crypto map configuration mode. • *name*—Name of the IPsec profile.
Step 2	`Router(config-crypto-map)#` **`set transform-set`** *`transform-set-name`*	Specifies which transform sets can be used with the IPsec profile. • *transform-set-name*—Name of the transform set.
Step 3	`Router(config-crypto-map)#` **`set identity`**	(Optional) Specifies identity restrictions to be used with the IPsec profile.
Step 4	`Router(config-crypto-map)#` **`set security association lifetime`** `{`**`seconds`** *`seconds`* `|` **`kilobytes`** *`kilobytes`*`}`	(Optional) Overrides the global lifetime value for the IPsec profile. • *seconds*— Number of seconds a security association will live before expiring. • *kilobytes*— Volume of traffic (in kilobytes) that can pass between IPsec peers using a given security association before that security association expires.
Step 5	`Router(config-crypto-map)#` **`set pfs`** `[`**`group1`** `|` **`group2`**`]`	(Optional) Specifies that IP Security should ask for perfect forward secrecy (PFS) when requesting new security associations for this IPsec profile. If this command is not specified, the default (group1) will be enabled. • **group1**—(Optional) Specifies that IPsec should use the 768-bit Diffie-Hellman (DH) prime modulus group when performing the new DH exchange. • **group2**—(Optional) Specifies the 1024-bit DH prime modulus group.

Configuring the Hub for DMVPN in VRF Mode

In VPN routing and forwarding instance (VRF) mode, to configure the hub switch for mGRE and IPsec integration (that is, to associate the tunnel with the IPsec profile configured in the previous procedure), perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface tunnel** *tunnel-number*	Configures a tunnel interface and enters interface configuration mode. • *tunnel-number*—Number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 2	Router(config-if)# **ip vrf forwarding** *inside-vrf-name*	(Optional) Associates a VRF with an interface or subinterface. This step is required only when configuring an inside VRF. • *inside-vrf-name*—Name assigned to the VRF.
Step 3	Router(config-if)# **ip address** *ip-address mask* [*secondary*]	Sets a primary or secondary IP address for the tunnel interface. • *address*—IP address. • *mask*—Subnet mask. • *secondary*—(Optional) Secondary IP address.
Step 4	Router(config-if)# **ip mtu** *bytes*	(Optional) Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface. • *bytes*—MTU size in bytes.
Step 5	Router(config-if)# **ip nhrp authentication** *string*	(Optional) Configures the authentication string for an interface using the Next Hop Resolution Protocol (NHRP). • *string*—Text of the authentication string. This string must be identical for all tunnels belonging to the same DMVPN.
Step 6	Router(config-if)# **ip nhrp map multicast dynamic**	Allows NHRP to automatically add spoke switches to the multicast NHRP mappings.
Step 7	Router(config-if)# **ip nhrp network-id** *number*	Enables NHRP on an interface. • *number*—A 32-bit network identifier, unique within this chassis, from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 8	Router(config-if)# **tunnel source** {*ip-address* | *type number*}	Sets source address for a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel. • *type number*—Interface type and number (for example, VLAN 2).

	Command	Purpose
Step 9	Router(config-if)# **tunnel mode gre multipoint**	Sets the encapsulation mode to mGRE for the tunnel interface.
Step 10	Router(config-if)# **tunnel vrf** *front-door-vrf-name*	(Optional) Associates a VRF instance with a specific tunnel destination, interface, or subinterface. This step is required only when configuring a front door VRF (FVRF). • *front-door-vrf-name*—Name assigned to the VRF. This may or may not be the same as the *inside-vrf-name.*
Step 11	Router(config-if)# **tunnel protection ipsec profile** *name*	Associates a tunnel interface with an IPsec profile. • *name*—Name of the IPsec profile; this value must match the name specified in the **crypto ipsec profile** command.
Step 12	Router(config-if)# **crypto engine slot** *slot/subslot* **inside**	Assigns the specified crypto engine to the inside interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located.
Step 13	Router(config-if)# **interface** *type slot/subslot/port*	Configures the DMVPN physical egress interface.
Step 14	Router(config-if)# **ip vrf forwarding** *front-door-vrf-name*	(Optional) Associates a VRF with an interface or subinterface. This step is required only when configuring a front door VRF (FVRF). • *front-door-vrf-name*—Name assigned to the VRF. This is the same name used in Step 10.
Step 15	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for an interface. • *address*—IP address. • *mask*—Subnet mask.
Step 16	Router(config-if)# **crypto engine slot** *slot/subslot* **outside** or in Cisco IOS Release 12.2(33)SXI and later releases: Router(config-if)# **crypto engine outside**	Enables the crypto engine on the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located. In Cisco IOS Release 12.2(33)SXI and later releases, it is not necessary to specify the slot for the outside interface in VRF mode.

Configuring the Hub for DMVPN in Crypto-Connect Mode

In crypto-connect mode, to configure the hub switch for mGRE and IPsec integration (that is, to associate the tunnel with the IPsec profile configured in the previous procedure), perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface tunnel`** *`tunnel-number`*	Configures a tunnel interface and enters interface configuration mode. • *tunnel-number*—Number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 2	`Router(config-if)#` **`ip address`** *`ip-address mask`* `[`*`secondary`*`]`	Sets a primary or secondary IP address for the tunnel interface. • *address*—IP address. • *mask*—Subnet mask. • *secondary*—(Optional) Secondary IP address.
Step 3	`Router(config-if)#` **`ip mtu`** *`bytes`*	(Optional) Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface. • *bytes*—MTU size in bytes.
Step 4	`Router(config-if)#` **`ip nhrp authentication`** *`string`*	(Optional) Configures the authentication string for an interface using the Next Hop Resolution Protocol (NHRP). • *string*—Text of the authentication string. This string must be identical for all tunnels belonging to the same DMVPN.
Step 5	`Router(config-if)#` **`ip nhrp map multicast dynamic`**	Allows NHRP to automatically add spoke switches to the multicast NHRP mappings.
Step 6	`Router(config-if)#` **`ip nhrp network-id`** *`number`*	Enables NHRP on an interface. • *number*—A 32-bit network identifier, unique within this chassis, from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 7	`Router(config-if)#` **`tunnel source`** `{`*`ip-address`* `|` *`type number`*`}`	Sets source address for a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel. • *type number*—Interface type and number (for example, VLAN 2).
Step 8	`Router(config-if)#` **`tunnel mode gre multipoint`**	Sets the encapsulation mode to mGRE for the tunnel interface.

	Command	Purpose
Step 9	Router(config-if)# **tunnel protection ipsec profile** *name*	Associates a tunnel interface with an IPsec profile. • *name*—Name of the IPsec profile; this value must match the name specified in the **crypto ipsec profile** command.
Step 10	Router(config-if)# **crypto engine slot** *slot/subslot*	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located.
Step 11	Router(config)# **interface vlan** *ifvlan*	Configures the DMVPN inside VLAN.
Step 12	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for an interface. • *address*—IP address. Enter the value specified in Step 7. • *mask*—Subnet mask.
Step 13	Router(config-if)# **crypto engine slot** *slot/subslot*	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located.
Step 14	Router(config-if)# **interface** *type slot/subslot/port*	Configures the DMVPN physical egress interface.
Step 15	Router(config-if)# **no ip address**	Assigns no IP address to the interface.
Step 16	Router(config-if)# **crypto connect vlan** *ifvlan*	Connects the outside access port VLAN to the inside (crypto) interface VLAN and enters crypto-connect mode. • *ifvlan*—DMVPN inside VLAN identifier.

Configuring the Spoke for DMVPN in VRF Mode

In VRF mode, to configure spoke switches for mGRE and IPsec integration, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface tunnel** *tunnel-number*	Configures a tunnel interface and enters interface configuration mode • *tunnel-number*—Number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 2	Router(config-if)# **ip vrf forwarding** *inside-vrf-name*	(Optional) Associates a VRF with an interface or subinterface. This step is required only when configuring an inside VRF. • *inside-vrf-name*—Name assigned to the VRF.

	Command	Purpose
Step 3	Router(config-if)# **ip address** *ip-address mask* [*secondary*]	Sets a primary or secondary IP address for the tunnel interface. • *address*—IP address. • *mask*—Subnet mask. • *secondary*—(Optional) Secondary IP address.
Step 4	Router(config-if)# **ip mtu** *bytes*	(Optional) Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface. • *bytes*—MTU size in bytes.
Step 5	Router(config-if)# **ip nhrp authentication** *string*	Configures the authentication string for an interface using NHRP. • *string*—Text of the authentication string. This string must be identical for all tunnels belonging to the same DMVPN.
Step 6	Router(config-if)# **ip nhrp map** *hub-tunnel-ip-address hub-physical-ip-address*	Statically configures the IP-to-NonBroadcast MultiAccess (NBMA) address mapping of IP destinations connected to an NBMA network. • *hub-tunnel-ip-address*—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub. • *hub-physical-ip-address*—Defines the static public IP address of the hub.
Step 7	Router(config-if)# **ip nhrp map multicast** *hub-physical-ip-address*	Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub switch. • *hub-physical-ip-address*—Defines the static public IP address of the hub.
Step 8	Router(config-if)# **ip nhrp nhs** *hub-tunnel-ip-address*	Configures the hub switch as the NHRP next-hop server. • *hub-tunnel-ip-address*—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub.
Step 9	Router(config-if)# **ip nhrp network-id** *number*	Enables NHRP on an interface. • *number*—A 32-bit network identifier, unique within this chassis, from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 10	Router(config-if)# **tunnel source** {*ip-address* | *type number*}	Sets source address for a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel. • *type number*—Interface type and number; for example, VLAN 2.

	Command	Purpose
Step 11	Router(config-if)# **tunnel mode gre multipoint**	Sets the encapsulation mode to mGRE for the tunnel interface. Use this command if data traffic can use dynamic spoke-to-spoke traffic.
Step 12	Router(config-if)# **tunnel protection ipsec profile** *name*	Associates a tunnel interface with an IPsec profile. • *name*—Name of the IPsec profile; this value must match the name specified in the **crypto ipsec profile** command.
Step 13	Router(config-if)# **crypto engine slot** *slot/subslot* **inside**	Assigns the specified crypto engine to the inside interface. • *slot/subslot*—The slot where the VSPA is located.
Step 14	Router(config-if)# **interface** *type slot/subslot/port*	Configures the DMVPN physical egress interface.
Step 15	Router(config-if)# **ip address** *address mask*	Sets a primary or secondary IP address for an interface. • *address*—IP address. • *mask*—Subnet mask.
Step 16	Router(config-if)# **crypto engine slot** *slot/subslot* **outside** or in Cisco IOS Release 12.2(33)SXI and later releases: Router(config-if)# **crypto engine outside**	Enables the crypto engine on the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located. In Cisco IOS Release 12.2(33)SXI and later releases, it is not necessary to specify the slot for the outside interface in VRF mode.

Configuring the Spoke for DMVPN in Crypto-Connect Mode

In crypto-connect mode, to configure spoke switches for mGRE and IPsec integration, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface tunnel** *tunnel-number*	Configures a tunnel interface and enters interface configuration mode • *tunnel-number*—Number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 2	Router(config-if)# **ip address** *ip-address mask* [*secondary*]	Sets a primary or secondary IP address for the tunnel interface. • *address*—IP address. • *mask*—Subnet mask. • *secondary*—(Optional) Secondary IP address.

	Command	Purpose
Step 3	`Router(config-if)#` **`ip mtu`** *`bytes`*	(Optional) Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface. • *bytes*—MTU size in bytes.
Step 4	`Router(config-if)#` **`ip nhrp authentication`** *`string`*	Configures the authentication string for an interface using NHRP. • *string*—Text of the authentication string. This string must be identical for all tunnels belonging to the same DMVPN.
Step 5	`Router(config-if)#` **`ip nhrp map`** *`hub-tunnel-ip-address hub-physical-ip-address`*	Statically configures the IP-to-NonBroadcast MultiAccess (NBMA) address mapping of IP destinations connected to an NBMA network. • *hub-tunnel-ip-address*—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub. • *hub-physical-ip-address*—Defines the static public IP address of the hub.
Step 6	`Router(config-if)#` **`ip nhrp map multicast`** *`hub-physical-ip-address`*	Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub switch. • *hub-physical-ip-address*—Defines the static public IP address of the hub.
Step 7	`Router(config-if)#` **`ip nhrp nhs`** *`hub-tunnel-ip-address`*	Configures the hub switch as the NHRP next-hop server. • *hub-tunnel-ip-address*—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub.
Step 8	`Router(config-if)#` **`ip nhrp network-id`** *`number`*	Enables NHRP on an interface. • *number*—A 32-bit network identifier, unique within this chassis, from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 9	`Router(config-if)#` **`tunnel source`** `{`*`ip-address`* `|` *`type number`*`}`	Sets source address for a tunnel interface. • *ip-address*—IP address to use as the source address for packets in the tunnel. • *type number*—Interface type and number; for example, VLAN 2.
Step 10	`Router(config-if)#` **`tunnel mode gre multipoint`**	Sets the encapsulation mode to mGRE for the tunnel interface. Use this command if data traffic can use dynamic spoke-to-spoke traffic.
Step 11	`Router(config-if)#` **`tunnel protection ipsec profile`** *`name`*	Associates a tunnel interface with an IPsec profile. • *name*—Name of the IPsec profile; this value must match the name specified in the **crypto ipsec profile** command.

	Command	Purpose
Step 12	`Router(config-if)# crypto engine slot slot/subslot`	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located.
Step 13	`Router(config)# interface vlan ifvlan`	Configures the DMVPN inside VLAN.
Step 14	`Router(config-if)# ip address address mask`	Sets a primary or secondary IP address for an interface. • *address*—IP address. Enter the value specified in Step 7. • *mask*—Subnet mask.
Step 15	`Router(config-if)# crypto engine slot slot/subslot`	Assigns the specified crypto engine to the interface. • *slot/subslot*—The slot where the IPsec VPN SPA is located.
Step 16	`Router(config-if)# interface type slot/subslot/port`	Configures the DMVPN physical egress interface.
Step 17	`Router(config-if)# no ip address`	Assigns no IP address to the interface.
Step 18	`Router(config-if)# crypto connect vlan ifvlan`	Connects the outside access port VLAN to the inside interface VLAN and enters crypto-connect mode. • *ifvlan*—DMVPN inside VLAN identifier.

Verifying the DMVPN Configuration

To verify that your DMVPN configuration is working, use the **show crypto isakmp sa**, **show crypto map**, and **show ip nhrp** commands.

The **show crypto isakmp sa** command displays all current IKE security associations (SAs) at a peer.

The following sample output is displayed after IKE negotiations have successfully completed between a hub and two spokes and between the two spokes, as shown in Figure 27-1 on page 27-2:

```
HUB# show crypto isakmp sa
dst             src             state          conn-id slot status
10.0.0.1        11.0.0.1        QM_IDLE          68001 ACTIVE
10.0.0.1        21.0.0.1        QM_IDLE          68002 ACTIVE


SPOKE1# show crypto isakmp sa
dst             src             state          conn-id slot status
11.0.0.1        21.0.0.1        QM_IDLE          68002 ACTIVE
21.0.0.1        11.0.0.1        QM_IDLE          68003 ACTIVE
10.0.0.1        11.0.0.1        QM_IDLE          68001 ACTIVE

SPOKE2# show crypto isakmp sa
dst             src             state          conn-id slot status
10.0.0.1        21.0.0.1        QM_IDLE          68001 ACTIVE
11.0.0.1        21.0.0.1        QM_IDLE          68003 ACTIVE
21.0.0.1        11.0.0.1        QM_IDLE          68002 ACTIVE
```

The **show crypto map** command displays the crypto map configuration.

The following sample output is displayed after a crypto map has been configured:

```
HUB# show crypto map
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
```

```
        Profile name: VPN-PROF
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 11.0.0.1
        Extended IP access list
            access-list  permit gre host 10.0.0.1 host 11.0.0.1
        Current peer: 11.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65538 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 21.0.0.1
        Extended IP access list
            access-list  permit gre host 10.0.0.1 host 21.0.0.1
        Current peer: 21.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0
 using crypto engine SPA-IPSEC-2G[4/0]



SPOKE1# show crypto map
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: VPN-PROF
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 10.0.0.1
        Extended IP access list
            access-list  permit gre host 11.0.0.1 host 10.0.0.1
        Current peer: 10.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65538 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 21.0.0.1
        Extended IP access list
            access-list  permit gre host 11.0.0.1 host 21.0.0.1
        Current peer: 21.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0
 using crypto engine SPA-IPSEC-2G[4/0]



SPOKE2# show crypto map
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: VPN-PROF
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 10.0.0.1
        Extended IP access list
            access-list  permit gre host 21.0.0.1 host 10.0.0.1
        Current peer: 10.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }

Crypto Map "Tunnel0-head-0" 65538 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 11.0.0.1
        Extended IP access list
            access-list  permit gre host 21.0.0.1 host 11.0.0.1
        Current peer: 11.0.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0
 using crypto engine SPA-IPSEC-2G[4/0]
```

The **show ip nhrp** command displays the NHRP cache.

The following sample output shows that NHRP registration occurred. Note that NHRP between the hub and a spoke is static, while NHRP between spokes is dynamic:

```
Router# show ip nhrp
HUB# show ip nhrp
30.1.0.1/32 via 30.1.0.1, Tunnel0 created 00:18:13, expire 01:41:46
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 11.0.0.1
30.2.0.1/32 via 30.2.0.1, Tunnel0 created 00:11:55, expire 01:48:04
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 21.0.0.1


SPOKE1# show ip nhrp
30.0.0.1/32 via 30.0.0.1, Tunnel0 created 00:23:39, never expire
  Type: static, Flags: authoritative used
  NBMA address: 10.0.0.1
30.2.0.1/32 via 30.2.0.1, Tunnel0 created 00:04:27, expire 01:47:59
  Type: dynamic, Flags: router
  NBMA address: 21.0.0.1


SPOKE2# show ip nhrp
30.0.0.1/32 via 30.0.0.1, Tunnel0 created 00:12:02, never expire
  Type: static, Flags: authoritative used
  NBMA address: 10.0.0.1
30.1.0.1/32 via 30.1.0.1, Tunnel0 created 00:04:29, expire 01:41:40
  Type: dynamic, Flags: router
  NBMA address: 11.0.0.1
```

For DMVPN configuration examples, see the "DMVPN Configuration Examples" section on page 27-18.

Configuring the Easy VPN Server

The Easy VPN server provides server support for the Cisco VPN Client Release 4.x and later software clients and Cisco VPN hardware clients. The feature allows a remote end user to communicate using IP Security (IPsec) with any Cisco IOS Virtual Private Network (VPN) gateway. Centrally managed IPsec policies are pushed to the client by the server, minimizing configuration by the end user.

Easy VPN Server features include:

- Mode configuration and Xauth support
- User-based policy control
- Session monitoring for VPN group access
- RADIUS server support
- **backup-gateway** command
- **pfs** command
- Virtual IPsec interface support
- Banner, auto-update, and browser proxy
- Configuration management enhancements (pushing a configuration URL through a mode-configuration exchange)
- Per-user AAA policy download with PKI
- Syslog message enhancements
- Network admission control support

Easy VPN Server Configuration Guidelines and Restrictions

When configuring the Easy VPN server, follow these guidelines and restrictions:

- The following IPsec protocol options and attributes currently are not supported by Cisco VPN clients, so these options and attributes should not be configured on the switch for these clients:
  - Authentication with public key encryption
  - Digital Signature Standard (DSS)

  - Diffie-Hellman (DH) groups (1)
  - IPsec Protocol Identifier (IPSEC_AH)
  - IPsec Protocol Mode (Transport mode)
  - Manual keys
  - Perfect Forward Secrecy (PFS)
- Enhanced Easy VPN, which uses Dynamic Virtual Tunnel Interfaces (DVTI) instead of dynamic crypto maps, is not supported.

For complete configuration information about the Easy VPN Server feature and the enhancements, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/ftunity.html

Configuring the Easy VPN Remote

The Easy VPN remote feature allows Cisco routers and security appliances to establish a site-to-site VPN connection to a Cisco Easy VPN Server without complex remote-side configuration. Centrally managed IPsec policies are pushed to the client by the server, minimizing configuration by the end user.

Easy VPN Remote features include the following:

- Virtual IPsec interface support
- Banner, auto-update, and browser proxy
- Dual tunnel support
- Configuration management enhancements (pushing a configuration URL through a mode-configuration exchange)
- Reactivate primary peer

Easy VPN Remote Configuration Guidelines

Follow these guidelines when configuring Easy VPN for the IPsec VPN SPA:

Caution You must clear all other crypto configurations from your running configuration on the Cisco IOS-based Easy VPN client that you are using to connect to the IPsec VPN SPA. If an ISAKMP policy is configured, it takes precedence over the preinstalled Easy VPN ISAKMP policies and the connection will fail. Other clients such as the VPN3000 and PIX systems running Easy VPN will prevent you from configuring Easy VPN unless all crypto configurations are removed. For complete configuration information for Easy VPN client support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ftezvpnr.html

For an Easy VPN server configuration example, see the "Easy VPN Server (Router Side) Configuration Example" section on page 27-22.

Configuring Easy VPN Remote RSA Signature Storage

The Easy VPN remote RSA signature support feature provides for the support of Rivest, Shamir, and Adelman (RSA) signatures on Easy VPN remote devices. The support is provided through RSA certificates that can be stored on or off the remote device.

Note The Easy VPN remote RSA signature support feature supported in Cisco IOS Release 12.2(33)SXH and later releases.

Easy VPN Remote RSA Signature Support Configuration Guidelines and Restrictions

When configuring Easy VPN remote RSA signature support, follow these guidelines and restrictions:

- You must have a Cisco Virtual Private Network (VPN) remote device and be familiar with configuring the device.
- You must have a certificate authority (CA) available to your network before you configure this interoperability feature. The CA must support the public key infrastructure (PKI) protocol of Cisco Systems, which is the Simple Certificate Enrollment Protocol (SCEP) (formerly called Certificate Enrollment Protocol [CEP]).
- This feature should be configured only when you also configure both IPsec and Internet Key Exchange (IKE) in your network.
- The Cisco IOS software does not support CA server public keys greater than 2048 bits.

Configuring Easy VPN Remote RSA Signature Support

The RSA signatures for an Easy VPN remote device are configured the same way that you would configure RSA signatures for any other Cisco device.

For information about configuring RSA signatures, refer to the *Cisco IOS Security Configuration Guide*.

To enable the RSA signatures, when you are configuring the Easy VPN remote and assigning the configuration to the outgoing interface, you must omit the **group** command. The content of the first Organizational Unit (OU) field will be used as the group.

For information about configuring Cisco Easy VPN remote devices, refer to the feature document, *Easy VPN Remote RSA Signature Support*, at the following location:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gtevcrsa.html

Configuration Examples

This section provides examples of the following configurations:

- DMVPN Configuration Examples, page 27-18
- Easy VPN Server (Router Side) Configuration Example, page 27-22

Note The following examples use commands at the level of Cisco IOS Release 12.2(33)SXH.

As of Cisco IOS Release 12.2(33)SXH, the **crypto engine subslot** command used in previous releases has been replaced with the **crypto engine slot** command (of the form **crypto engine slot** *slot* {**inside** | **outside**}). The **crypto engine subslot** command is no longer supported. When upgrading, ensure that this command has been modified in your start-up configuration to avoid extended maintenance time.

DMVPN Configuration Examples

The following sections provide examples of DMVPN configuration:

- DMVPN Hub with VRF Mode Configuration Example, page 27-18
- DMVPN Spoke with VRF Mode Configuration Example, page 27-20
- DMVPN Spoke with Crypto-Connect Mode Configuration Example, page 27-21

The DMVPN examples are based on the implementation shown in Figure 27-1 on page 27-2, using the following configuration parameters:

- The hub switch (HUB) is configured in VRF mode with inside VRF (IVRF) and front-door VRF (FVRF).
- One spoke switch (SPOKE1) is configured in VRF mode with IVRF but no FVRF.
- One spoke switch (SPOKE2) is configured in crypto-connect mode.
- EIGRP is configured to distribute routes over the tunnels.
- In all switches, interface gi3/1 is the interface to the provider network.
- In all switches, interface gi3/13 is the interface to the private LAN.

Note The tunnel source can be the same as the physical egress port. If the tunnel source is not the physical egress port, make sure that traffic to and from the tunnel source passes through the physical egress port.

DMVPN Hub with VRF Mode Configuration Example

The following is a configuration example of the IPsec VPN SPA serving as a DMVPN hub using VRF mode with inside VRF and front-door VRF (FVRF):

```
hostname HUB
!
ip vrf fvrf
 rd 1000:1
!
ip vrf ivrf
 rd 1:1
!
crypto engine mode vrf
!
crypto keyring RING1 vrf fvrf
  pre-shared-key address 0.0.0.0 0.0.0.0 key abcdef
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
!
crypto ipsec transform-set ts esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile VPN-PROF
 set transform-set ts
!
!
interface Tunnel0
! EIGRP uses the configured bandwidth to allocate bandwidth for its routing update 
mechanism
 bandwidth 1000000
 ip vrf forwarding ivrf
 ip address 30.0.0.1 255.0.0.0
 ip nhrp authentication cisco123
 ip nhrp map multicast dynamic
 ip nhrp network-id 1000
! For a large number of tunnels, the following two commands are recommended
! EIGRP timers are adjusted to match the default timers for a WAN interface
 ip hello-interval eigrp 200 60
 ip hold-time eigrp 200 180
! The following two EIGRP commands are necessary to allow spoke-to-spoke communication
 no ip next-hop-self eigrp 200
 no ip split-horizon eigrp 200
 tunnel source Vlan10
 tunnel mode gre multipoint
 tunnel vrf fvrf
 tunnel protection ipsec profile VPN-PROF
 crypto engine slot 4/0 inside
!
interface Vlan10
 ip vrf forwarding fvrf
 ip address 10.0.0.1 255.255.255.0
 crypto engine outside
!
interface GigabitEthernet3/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 10
 switchport mode trunk

interface GigabitEthernet3/13
 description Local LAN interface
 ip vrf forwarding ivrf
 ip address 70.0.0.1 255.255.255.0

router eigrp 10
 no auto-summary
 !
 address-family ipv4 vrf ivrf
 redistribute connected
 network 30.0.0.0
 network 70.0.0.0
 no auto-summary
 autonomous-system 200
 exit-address-family
!
! In this example, tunnel destination reachability is provided by static routes
! A routing protocol could also be used
ip route vrf fvrf 11.0.0.0 255.0.0.0 10.0.0.2
ip route vrf fvrf 21.0.0.0 255.0.0.0 10.0.0.2

end
```

DMVPN Spoke with VRF Mode Configuration Example

The following is a configuration example of the IPsec VPN SPA serving as a DMVPN spoke using VRF mode with inside VRF but no front-door VRF:

```
hostname SPOKE1
!
ip vrf ivrf
 rd 1:1
!
crypto engine mode vrf
!
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key abcdef address 0.0.0.0 0.0.0.0
crypto isakmp keepalive 60
!
!
crypto ipsec transform-set ts esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile VPN-PROF
 set transform-set ts
!
interface Tunnel0
 bandwidth 100000
 ip vrf forwarding ivrf
 ip address 30.1.0.1 255.0.0.0
 ip nhrp authentication cisco123
 ip nhrp map 30.0.0.1 10.0.0.1
 ip nhrp map multicast 10.0.0.1
 ip nhrp network-id 1000
 ip nhrp nhs 30.0.0.1
 ip hello-interval eigrp 200 60
 ip hold-time eigrp 200 180
 tunnel source Loopback0
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN-PROF
 crypto engine slot 4/0 inside
!
interface Loopback0
 ip address 11.0.0.1 255.255.255.0
!

interface GigabitEthernet3/1
 ip address 11.255.255.1 255.255.255.0
 crypto engine outside
!
interface GigabitEthernet3/13
 ip vrf forwarding ivrf
 ip address 80.0.0.1 255.255.255.0

router eigrp 10
 no auto-summary
 !
 address-family ipv4 vrf ivrf
 autonomous-system 200
 network 30.0.0.0
 network 70.0.0.0
 no auto-summary
 redistribute connected
 exit-address-family

ip route 10.0.0.0 255.0.0.0 11.255.255.2
ip route 21.0.0.0 255.0.0.0 11.255.255.2

end
```

DMVPN Spoke with Crypto-Connect Mode Configuration Example

The following is a configuration example of the IPsec VPN SPA serving as a DMVPN spoke using crypto-connect mode:

```
hostname SPOKE2
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key abcdef address 0.0.0.0 0.0.0.0
crypto isakmp keepalive 60
!
!
crypto ipsec transform-set ts esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile VPN-PROF
 set transform-set ts
!
interface Tunnel0
 bandwidth 1000000
 ip address 30.2.0.1 255.0.0.0
 ip nhrp authentication cisco123
 ip nhrp map 30.0.0.1 10.0.0.1
 ip nhrp map multicast 10.0.0.1
 ip nhrp network-id 1000
 ip nhrp nhs 30.0.0.1
 ip hello-interval eigrp 200 60
 ip hold-time eigrp 200 180
 tunnel source Vlan10
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN-PROF
 crypto engine slot 4/0 inside
!
interface Vlan10
 ip address 21.0.0.1 255.255.255.0
 no mop enabled
 crypto engine slot 4/0 inside
!
interface GigabitEthernet3/1
 no ip address
 crypto connect vlan 10
!
interface GigabitEthernet3/13
 ip address 90.0.0.1 255.255.255.0
!
router eigrp 200
 redistribute connected
 network 30.0.0.0
 network 90.0.0.0
 no auto-summary

ip route 10.0.0.0 255.0.0.0 21.0.0.2
ip route 11.0.0.0 255.0.0.0 21.0.0.2

end
```

Easy VPN Server (Router Side) Configuration Example

The following is an example of an Easy VPN server router-side configuration:

```
!
version 12.2
!
hostname sanjose
!
logging snmp-authfail
logging buffered 1000000 debugging
aaa new-model
aaa authentication login authen local
aaa authorization network author local
!
username unity password 0 uc
ip subnet-zero
no ip source-route
!
mpls ldp logging neighbor-changes
mls flow ip destination
mls flow ipx destination
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 12345 address 0.0.0.0 0.0.0.0
crypto isakmp keepalive 10 2
!
crypto isakmp client configuration group group1
 key 12345
 domain cisco.com
 pool pool1
!
crypto isakmp client configuration group default
 key 12345
 domain cisco.com
 pool pool2
!
crypto ipsec transform-set myset3 esp-3des esp-md5-hmac
!
crypto dynamic-map test_dyn 1
 set transform-set myset3
 reverse-route
!
! Static client mapping
crypto map testtag client authentication list authen
crypto map testtag isakmp authorization list author
crypto map testtag client configuration address respond
crypto map testtag 10 ipsec-isakmp
 set peer 10.5.1.4
 set security-association lifetime seconds 900
 set transform-set myset3
 match address 109
!
! Dynamic client mapping
crypto map test_dyn client authentication list authen
crypto map test_dyn isakmp authorization list author
crypto map test_dyn client configuration address respond
crypto map test_dyn 1 ipsec-isakmp dynamic test_dyn
!
!
no spanning-tree vlan 513
!
redundancy
  main-cpu
  auto-sync running-config
  auto-sync standard
!
interface GigabitEthernet2/1
 no ip address
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,513,1002-1005
 switchport mode trunk
!
interface GigabitEthernet2/2
 no ip address
 shutdown
!
interface GigabitEthernet6/1/1
 no ip address
 flowcontrol receive on
 flowcontrol send off
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,513,1002-1005
 switchport mode trunk
 cdp enable
!
interface GigabitEthernet6/1/2
 no ip address
 flowcontrol receive on
 flowcontrol send off
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 cdp enable
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan2
 no ip address
 crypto connect vlan 513
!
interface Vlan513
 ip address 10.5.1.1 255.255.0.0
 crypto map test_dyn
 crypto engine slot 6/1 inside
!
ip local pool pool1 22.0.0.2
ip local pool pool2 23.0.0.3
ip classless
ip pim bidir-enable
!
access-list 109 permit ip host 10.5.1.1 host 22.0.0.2
arp 127.0.0.12 0000.2100.0000 ARPA
!
snmp-server enable traps tty
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
!
line con 0
line vty 0 4
 password lab
 transport input lat pad mop telnet rlogin udptn nasi
!
end
```

CHAPTER 28

Configuring Duplicate Hardware and IPsec Failover Using the IPsec VPN SPA

This chapter provides information about configuring duplicate hardware and IPsec failover using the IPsec VPN SPA on the Catalyst 6500 Series switch. It includes the following sections:

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For detailed information on Cisco IOS IPsec cryptographic operations and policies, refer to the *Cisco IOS Security Configuration Guide, Release 12.2* and *Cisco IOS Security Command Reference, Release 12.2*.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Tip To ensure a successful configuration of your VPN using the IPsec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Overview of Duplicate Hardware Configurations and IPsec Failover

For critical VPN communications, you can deploy redundant VPN hardware and configure your system for failover in case of hardware failure. The following topics provide information about configuring for IPsec failover using the IPsec VPN SPA:

- Configuring Multiple IPsec VPN SPAs in a Chassis, page 28-2
- Understanding Stateless Failover Using HSRP, page 28-3
- Understanding Stateful Failover Using HSRP and SSP, page 28-3.

Configuring Multiple IPsec VPN SPAs in a Chassis

You can deploy up to ten IPsec VPN SPAs in a single chassis, with the restriction that no more than one IPsec VPN SPA can be used to perform IPsec services for any given interface VLAN.

Multiple IPsec VPN SPAs in a Chassis Configuration Guidelines

When configuring multiple IPsec VPN SPAs in a chassis, follow these guidelines:

- If you enter the **no switchport** command followed by the **switchport** command, all VLANs are readded to a trunk port (this situation occurs when you are first switching to a routed port and then back to a switch port). For detailed information on configuring trunk ports, see the "Configuring a Trunk Port" section on page 21-14 of Chapter 21, "Configuring VPNs in Crypto-Connect Mode."
- As with single IPsec VPN SPA deployments, you must properly configure each IPsec VPN SPA's inside and outside port. You can add an interface VLAN only to the inside port of one IPsec VPN SPA. Do not add the same interface VLAN to the inside port of more than one IPsec VPN SPA.

  Assigning interface VLANs to the inside ports of the IPsec VPN SPAs allows you to decide which IPsec VPN SPA can be used to provide IPsec services for a particular interface VLAN.

  Note You do not need to explicitly add interface VLANs to the inside trunk ports of the IPsec VPN SPAs. Entering the **crypto engine slot** command achieves the same results.

  Note There is no support for using more than one IPsec VPN SPA to do IPsec processing for a single interface VLAN.

- SA-based load balancing is not supported.
- If you assign the same crypto map to multiple interfaces, then you must use the **crypto map local address** command, and all interfaces must be assigned to the same crypto engine.

For a configuration example of multiple IPsec VPN SPAs in a chassis, see the "Multiple IPsec VPN SPAs in a Chassis Configuration Example" section on page 28-24.

Understanding Stateless Failover Using HSRP

The IPsec failover (VPN high availability) feature allows you to employ a secondary (standby) switch that automatically takes over the primary (active) switch's tasks in the event of an active switch failure. IPsec failover, stateless or stateful, is designed to work in conjunction with the Hot Standby Routing Protocol (HSRP) and Reverse Route Injection (RRI).

HSRP is used between the active and standby switch in either stateless or stateful mode, tracking the state of switch interfaces and providing a failover mechanism between primary and secondary devices. An HSRP group shares a single virtual IP address as its crypto peer address so that the remote crypto peer requires no reconfiguration after a failover. The configured HSRP timers determine the time that it takes for the standby switch to take over.

RRI uses information derived from the negotiated IPsec SAs to create static routes to the networks identified in those SAs. During an HSRP and IPsec failover, RRI allows dynamic routing information updates.

In an IPsec stateless failover, the HSRP group's virtual IP address transfers over to the standby switch, but no IPsec or ISAKMP SA state information is transferred to the standby switch. The remote crypto peer detects the failure using Dead Peer Detection (DPD) or a keepalive mechanism. The remote crypto peer then communicates with the standby switch at the HSRP group address to renegotiate the dropped ISAKMP SAs and IPsec SAs before traffic transmission can resume.

When used together, HSRP and RRI provide a reliable network design for VPNs and reduce configuration complexity on remote peers.

For complete HSRP configuration information, refer to this URL:

http://www.cisco.com/en/US/docs/ios/ipapp/configuration/guide/ipapp_hsrp_ps6922_TSD_Products_Configuration_Guide_Chapter.html

Understanding Stateful Failover Using HSRP and SSP

Note Support for IPsec stateful failover using HSRP and SSP is removed in Cisco IOS Release 12.2(33)SXH and later releases. The feature is supported in Release 12.2SXF.

IPsec stateful failover enables a switch to continue processing and forwarding IPsec packets after a planned or unplanned outage. The failover process is transparent to users and to remote IPsec peers.

As with IPsec stateless failover, IPsec stateful failover is designed to work with HSRP and RRI, but IPsec stateful failover also uses the State Synchronization Protocol (SSP). During an HSRP and IPsec failover, SSP transfers IPsec and ISAKMP SA state information between the active and standby switches, allowing existing VPN connections to be maintained after a switch failover.

IPsec Stateful Failover Configuration Guidelines and Restrictions

When configuring IPsec stateful failover, follow these guidelines and restrictions:

- When configuring IPsec stateful failover with the IPsec VPN SPA, all IPsec VPN SPA configuration rules apply. You must apply crypto maps to interface VLANs.
- When configuring IPsec stateful failover with an IPsec VPN SPA in two chassis, the hardware configurations of both chassis must be exactly the same. For example, in one chassis if the IPsec VPN SPA that is in slot 2 is used to protect interface VLAN 100 and the IPsec VPN SPA that is in

slot 3 is used to protect interface VLAN 101, the exact same configuration must be reflected in the second chassis. An example of a misconfiguration would be if the IPsec VPN SPA in slot 3 of the second chassis is used to protect interface VLAN 100.

- Do not add nonexistent or inadequately configured HSRP standby groups to the State Synchronization Protocol (SSP) configuration because this action disables high-availability features until the configuration is corrected.
- The recommended HSRP timer values are one second for hello timers and three seconds for hold timers. These values should prevent an undesirable failover that is caused by temporary network congestion or transient, high CPU loads.

  These timer values can be adjusted upward if you are running high loads or have a large number of HSRP groups. Temporary failures and load-related system stability can be positively affected by raising the timer values as needed. The hello timer value should be approximately a third of the hold timer value.
- Use the HSRP delay timers to allow a device to finish booting, initializing, and synchronizing before participating as a high-availability pair. Set the minimum delay at 30 seconds or more to help prevent active/standby flapping and set the reload delay at some value greater than the minimum. You can use the delay timers to reflect the complexity and size of a particular configuration on various hardware. The delay timers tend to vary from platform to platform.
- Sequence number updates from active to standby have a 20-second minimum interval per SA.
- The **standby preempt** command is required, and should be configured with no **priority** or **delay** options.
- To allow dynamic routing information updates during the HSRP and IPsec failover, enable the Reverse Route Injection (RRI) feature using the **reverse-route** command.
- To verify that all processes are running properly after enabling both HSRP and IPsec stateful failover, use the **show ssp**, **show standby**, **show crypto ipsec**, and **show crypto isakmp** commands.
- The following features are not supported with IPsec stateful failover:
  - The **standby use-bia** command—Always use a virtual HSRP MAC address for the switch's MAC address.
  - Easy VPN clients or IKE keepalives— IPsec stateful failover can be used with peers when DPD is used.
  - DMVPN or tunnel protection.
  - Secured WAN ports (for example, IPsec over FlexWAN or SIP module port adapters)— This restriction is due to limitations of HSRP.

Configuring IPsec Failover

The following sections describe how to configure IPsec stateless and stateful failover in crypto-connect and VRF modes:

- Configuring IPsec Stateless Failover Using HSRP with Crypto-Connect Mode, page 28-5
- Configuring IPsec Stateful Failover Using HSRP and SSP with Crypto-Connect Mode, page 28-11
- Configuring IPsec Stateless and Stateful Failover with VRF Mode, page 28-17

Configuring IPsec Stateless Failover Using HSRP with Crypto-Connect Mode

To configure IP stateful failover using HSRP and SSP, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto isakmp policy** *priority* ... Router(config-isakmp) # **exit**	Defines an ISAKMP policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest. For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.
Step 2	Router(config)# **crypto isakmp key** *keystring* **address** *peer_address* [*mask*]	Configures a preshared authentication key. • *keystring*—Preshared key. • *peer_address*—IP address of the remote peer. • *mask*—(Optional) The subnet mask of the remote peer. For details on configuring a preshared key, see the *Cisco IOS Security Configuration Guide*.
Step 3	Router(config)# **crypto ipsec transform-set** *transform_set_name* *transform1*[*transform2*[*transform3*]] ... Router(config-crypto-tran)# **exit**	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform_set_name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. For accepted *transformx* values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference*.
Step 4	Router(config)# **crypto dynamic-map** *dynamic_map_name* *seq_number* **ipsec-isakmp** ... Router(config-crypto-map)# **exit**	Creates or modifies a dynamic crypto map template and enters the crypto map configuration mode. • *dynamic_map_name*—Name that identifies the dynamic crypto map template. • *seq_number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations. For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.

	Command	Purpose

Step 5	`Router(config)# crypto map map_name seq_number [ipsec-isakmp] [dynamic dynamic_map_name]`
	`...`
	`Router(config-crypto-map)# exit`
		Creates a crypto map entry and binds it to the dynamic crypto map template.
		• *map_name*—Name that identifies the crypto map set.
		• *seq_number*—Sequence number you assign to the crypto map entry. Lower values have higher priority.
		• **ipsec-isakmp**—(Optional) Specifies that IKE will be used to establish the IPsec security associations.
		• **dynamic**—(Optional) Specifies that this crypto map entry is to reference a preexisting dynamic crypto map.
		• *dynamic_map_name*—Name that identifies the dynamic crypto map template.

Step 6	`Router(config-if)# interface gigabitethernet slot/subslot/port`	Enters interface configuration mode for the LAN-side Gigabit Ethernet interface.
Step 7	`Router(config-if)# ip address address mask`	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 8	`Router(config-if)# standby [group_number] ip ip_address`	Enables the HSRP. • *group_number*—(Optional) Group number on the interface for which HSRP is being activated. The default is 0. The group number range is from 0 to 255 for HSRP version 1 and from 0 to 4095 for HSRP version 2. • *ip_address*— IP address of the standby switch interface.

	Command	Purpose
Step 9	Router(config-if)# **standby** [*group_number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime*	Configures the time between hello packets and the hold time before other switches declare the active switch to be down. • *group_number*—(Optional) Group number to which the timers apply. • **msec**—(Optional) Interval in milliseconds. Millisecond timers allow for faster failover. • *hellotime*—Hello interval (in seconds). This is an integer from 1 to 254. The default is 3 seconds. If the **msec** option is specified, *hellotime* is in milliseconds. This is an integer from 15 to 999. • *holdtime*—Time (in seconds) before the active or standby switch is declared to be down. This is an integer from x to 255. The default is 10 seconds. If the **msec** option is specified, *holdtime* is in milliseconds. This is an integer from y to 3000.
Step 10	Router(config-if)# **standby** [*group_number*] **priority** *priority*	(Optional) Sets the standby priority used in choosing the active switch. • *group_number*—(Optional) Group number to which this command applies. • *priority*—The priority value range is from 1 to 255, where 1 denotes the lowest priority and 255 denotes the highest priority. Specify that, if the local switch has priority over the current active switch, the local switch should attempt to take its place as the active switch.
Step 11	Router(config-if)# **standby** [*group_number*] **preempt**	Configure HSRP preemption. • *group_number*—(Optional) Group number to which this command applies. **Note** In software releases earlier than Cisco IOS Release 12.2(33)SXH, **preempt** is a keyword of the **standby priority** command. In Cisco IOS Release 12.2(33)SXH and later releases, **standby preempt** and **standby priority** are separate commands.

	Command	Purpose
Step 12	`Router(config-if)# standby [group_number] track type number [interface_priority]`	Configures the interface to track other interfaces, so that if one of the other interfaces goes down, the device's Hot Standby priority is lowered. • *group_number*—(Optional) Group number on the interface for which HSRP is being activated. • *type*—Interface type (combined with interface number) that will be tracked. • *number*—Interface number (combined with interface type) that will be tracked. • *interface_priority*—(Optional) Amount by which the Hot Standby priority for the switch is decremented (or incremented) when the interface goes down (or comes back up). Range is from 0 to 255. Default is 10.
Step 13	`Router(config-if)# standby [group_number] name`	Configures the standby group name for the interface. • *group_number*—(Optional) Group number to which the name is being applied. • *name*—Name of the HSRP standby group.
Step 14	`Router(config-if)# interface vlan vlan_ID`	Enters interface configuration mode for the specified crypto interface VLAN.
Step 15	`Router(config-if)# ip address address mask`	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 16	`Router(config-if)# standby [group_number] ip ip_address`	Enables the HSRP. • *group_number*—(Optional) Group number on the interface for which HSRP is being activated. The default is 0. The group number range is from 0 to 255 for HSRP version 1 and from 0 to 4095 for HSRP version 2. • *ip_address*—Virtual IP address of the HSRP standby group.

	Command	Purpose
Step 17	Router(config-if)# **standby** [*group_number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime*	Configures the time between hello packets and the hold time before other switches declare the active switch to be down. • *group_number*—(Optional) Group number to which the timers apply. • **msec**—(Optional) Interval in milliseconds. Millisecond timers allow for faster failover. • *hellotime*—Hello interval (in seconds). This is an integer from 1 to 254. The default is 3 seconds. If the **msec** option is specified, *hellotime* is in milliseconds. This is an integer from 15 to 999. • *holdtime*—Time (in seconds) before the active or standby switch is declared to be down. This is an integer from x to 255. The default is 10 seconds. If the **msec** option is specified, *holdtime* is in milliseconds. This is an integer from y to 3000.
Step 18	Router(config-if)# **standby** [*group_number*] **priority** *priority*	(Optional) Sets the standby priority used in choosing the active switch. • *group_number*—(Optional) Group number to which this command applies. • *priority*—The priority value range is from 1 to 255, where 1 denotes the lowest priority and 255 denotes the highest priority. Specify that, if the local switch has priority over the current active switch, the local switch should attempt to take its place as the active switch.
Step 19	Router(config-if)# **standby** [*group_number*] **preempt**	Configure HSRP preemption. • *group_number*—(Optional) Group number to which this command applies. **Note** In software releases earlier than Cisco IOS Release 12.2(33)SXH, **preempt** is a keyword of the **standby priority** command. In Cisco IOS Release 12.2(33)SXH and later releases, **standby preempt** and **standby priority** are separate commands.

	Command	Purpose
Step 20	`Router(config-if)# standby [group_number] track type number [interface_priority]`	Configures the interface to track other interfaces, so that if one of the other interfaces goes down, the device's hot standby priority is lowered. • *group_number*—(Optional) Group number on the interface for which HSRP is being activated. • *type*—Interface type (combined with interface number) that will be tracked. • *number*—Interface number (combined with interface type) that will be tracked. • *interface_priority*—(Optional) Amount by which the Hot Standby priority for the switch is decremented (or incremented) when the interface goes down (or comes back up). Range is from 0 to 255. Default is 10.
Step 21	`Router(config-if)# standby [group_number] name`	Configures the standby group name for the interface. • *group_number*—(Optional) Group number to which the name is being applied. • *name*—Name of the standby switch.
Step 22	`Router(config-if)# crypto map map_name redundancy name`	Defines a backup IPsec peer. Both routers in the standby group are defined by the redundancy standby name and share the same virtual IP address. • *map_name*—Name of the crypto map set. • *name*—Name of the HSRP standby group.
Step 23	`Router(config-if)# crypto engine slot slot/subslot`	Assigns the crypto engine to the inside interface VLAN. • *slot/subslot*—The slot and subslot where the IPsec VPN SPA is located.
Step 24	`Router(config-if)# interface gigabitethernet slot/subslot/port`	Enters interface configuration mode for the outside Gigabit Ethernet interface.
Step 25	`Router(config-if)# crypto connect vlan vlan_ID`	Connects the outside access port to the inside interface VLAN and enters crypto-connect mode. • *vlan_ID*—Interface VLAN identifier.

For examples of IPsec stateless failover configurations using HSRP, see the "IPsec Stateless Failover Using HSRP with Crypto-Connect Mode Configuration Examples" section on page 28-26.

Configuring IPsec Stateful Failover Using HSRP and SSP with Crypto-Connect Mode

The configuration of IPsec stateful failover using HSRP is very similar to the configuration of IPsec stateless failover using HSRP with the addition of the SSP-related commands.

Note Support for IPsec stateful failover using HSRP and SSP is removed in Cisco IOS Release 12.2(33)SXH and later releases. The feature is supported in Release 12.2SXF.

To configure IP stateful failover using HSRP and SSP, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ssp group group`	Indicates channel used to communicate high availability (HA) information and enters SSP configuration mode. • *group*—Integer between 1 and 100.
Step 2	`Router(config-ssp)# redundancy name`	Identifies the HSRP group. • *name*—Valid IP redundancy group name.
Step 3	`Router(config-ssp)# remote ipaddr`	Identifies peer that will receive high availability (HA) transmissions. • *ipaddr*—IP address of the standby switch.

Step 4	`Router(config)# crypto isakmp policy priority`
	`...`
	`Router(config-isakmp) # exit`
		Defines an ISAKMP policy and enters ISAKMP policy configuration mode.
		• *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
		For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.

Step 5	`Router(config)# crypto isakmp key keystring address peer_address`	Configures a preshared authentication key. • *keystring*—Preshared key. • *peer_address*—IP address of the remote peer. For details on configuring a preshared key, see the *Cisco IOS Security Configuration Guide*.
Step 6	`Router(config)# crypto isakmp ssp id`	Enables ISAKMP state to be transferred by the SSP channel described by the ID. If this feature is disabled, all dormant SA entries bound to that ID on the standby switch will be removed and any new state entries will not be added. • *id*—Channel used to transfer SA entries.

	Command	Purpose
Step 7	Router(config)# **crypto ipsec transform-set** *transform_set_name* *transform1*[*transform2*[*transform3*]] ... Router(config-crypto-tran)# **exit**	Defines a transform set (an acceptable combination of security protocols and algorithms) and enters crypto transform configuration mode. • *transform_set_name*—Name of the transform set. • *transform1*[*transform2*[*transform3*]]—Defines IPsec security protocols and algorithms. For accepted *transformx* values, and more details on configuring transform sets, see the *Cisco IOS Security Command Reference*.
Step 8	Router(config)# **crypto map** *name* **ha replay-interval inbound** *inbound-interval* **outbound** *outbound-interval*	(Optional) Specifies the intervals at which the active switch should update the standby switch with anti-replay sequence numbers. • *name*—Tag name of the crypto map described in the configuration. • *inbound-interval*—The interval at which the active switch sends packet sequence updates for incoming packets. The range is 0 to 10000 (packets); the default is 1000. • *outbound-interval*—The interval at which the active switch sends packet sequence updates for outgoing packets. The range is 1 to 10 (in millions of packets); the default is 1.
Step 9	Router(config)# **access-list** *access_list_number* {**deny** | **permit**} **ip** *source source_wildcard destination destination_wildcard*	Defines an extended IP access list. • *access_list_number*—Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699. • {**deny** | **permit**}—Denies or permits access if the conditions are met. • *source*—Address of the host from which the packet is being sent. • *source_wildcard*—Wildcard bits to be applied to the source address. • *destination*—Address of the host to which the packet is being sent. • *destination_wildcard*—Wildcard bits to be applied to the destination address. For details on configuring an access list, see the *Cisco IOS Security Configuration Guide*.

	Command	Purpose

Step 10	`Router(config)# `**`crypto dynamic-map`** *`dynamic_map_name seq_number`* **`ipsec-isakmp`**
	`...`
	`Router(config-crypto-map)# `**`exit`**
		Creates or modifies a dynamic crypto map template and enters the crypto map configuration mode.
		• *dynamic_map_name*—Name that identifies the dynamic crypto map template.
		• *seq_number*—Sequence number you assign to the crypto map entry. Lower values have higher priority.
		• **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations.
		For details on configuring a crypto map, see the *Cisco IOS Security Configuration Guide*.

Step 11	`Router(config)# `**`crypto map`** *`map_name seq_number`* **`ipsec-isakmp dynamic`** *`dynamic_map_name`*	Creates a crypto map entry and binds it to the dynamic crypto map template. • *map_name*—Name that identifies the crypto map set. • *seq_number*—Sequence number you assign to the crypto map entry. Lower values have higher priority. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPsec security associations. • *dynamic_map_name*—Name that identifies the dynamic crypto map template.
Step 12	`Router(config-if)# `**`interface gigabitethernet`** *`slot/subslot/port`*	Enters interface configuration mode for the LAN-side Gigabit Ethernet interface.
Step 13	`Router(config-if)# `**`ip address`** *`address mask`*	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 14	`Router(config-if)# `**`standby delay minimum`** *`min-seconds`* **`reload`** *`reload-seconds`*	Specifies the delay period before the initialization of HSRP groups. • *min-seconds*—Minimum time (in seconds) to delay HSRP group initialization after an interface comes up. This minimum delay period applies to all subsequent interface events. The valid range is 0 to 300 seconds. The default is 1 second. The recommended value is 30 seconds. • *reload-seconds*—Time (in seconds) to delay after the switch has reloaded. This delay period applies only to the first interface-up event after the switch has reloaded. The valid range is 0 to 300 seconds. The default is 5 seconds. The recommended value is 60 seconds.

	Command	Purpose
Step 15	Router(config-if)# **standby** [*group_number*] **ip** *ip_address*	Enables the HSRP. • *group_number*—(Optional) Group number on the interface for which HSRP is being activated. The default is 0. The group number range is from 0 to 255 for HSRP version 1 and from 0 to 4095 for HSRP version 2. • *ip_address*—Virtual IP address of the HSRP standby group.
Step 16	Router(config-if)# **standby** [*group_number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime*	(Optional) Configures the time between hello packets and the hold time before other switches declare the active switch to be down. • *group_number*—(Optional) Group number to which the timers apply. • **msec**—(Optional) Interval in milliseconds. Millisecond timers allow for faster failover. • *hellotime*—Hello interval (in seconds). This is an integer from 1 to 254. The default is 3 seconds. If the **msec** option is specified, *hellotime* is in milliseconds. This is an integer from 15 to 999. • *holdtime*—Time (in seconds) before the active or standby switch is declared to be down. This is an integer from x to 255. The default is 10 seconds. If the **msec** option is specified, *holdtime* is in milliseconds. This is an integer from y to 3000.
Step 17	Router(config-if)# **standby** [*group_number*] **preempt**	Configure HSRP preemption. • *group_number*—(Optional) Group number to which this command applies. **Note** In software releases earlier than Cisco IOS Release 12.2(33)SXH, **preempt** is a keyword of the **standby priority** command. In Cisco IOS Release 12.2(33)SXH and later releases, **standby preempt** and **standby priority** are separate commands.

	Command	Purpose
Step 18	`Router(config-if)# standby [group_number] track type number [interface_priority]`	Configures the interface to track other interfaces, so that if one of the other interfaces goes down, the device's Hot Standby priority is lowered. • *group_number*—(Optional) Group number on the interface for which HSRP is being activated. • *type*—Interface type (combined with interface number) that will be tracked. • *number*—Interface number (combined with interface type) that will be tracked. • *interface_priority*—(Optional) Amount by which the Hot Standby priority for the switch is decremented (or incremented) when the interface goes down (or comes back up). Range is from 0 to 255. Default is 10.
Step 19	`Router(config-if)# standby [group_number] name`	Configures the standby group name for the interface. • *group_number*—(Optional) Group number to which the name is being applied. • *name*—Name of the HSRP standby group.
Step 20	`Router(config-if)# interface vlan vlan_ID`	Enters interface configuration mode for the specified crypto interface VLAN.
Step 21	`Router(config-if)# ip address address mask`	Specifies the IP address and subnet mask for the interface. • *address*—IP address. • *mask*—Subnet mask.
Step 22	`Router(config-if)# standby delay minimum min-seconds reload reload-seconds`	Specifies the delay period before the initialization of HSRP groups. • *min-seconds*—Minimum time (in seconds) to delay HSRP group initialization after an interface comes up. This minimum delay period applies to all subsequent interface events. The valid range is 0 to 300 seconds. The default is 1 second. The recommended value is 30 seconds. • *reload-seconds*—Time (in seconds) to delay after the switch has reloaded. This delay period applies only to the first interface-up event after the switch has reloaded. The valid range is 0 to 300 seconds. The default is 5 seconds. The recommended value is 60 seconds.

	Command	Purpose
Step 23	Router(config-if)# **standby** [*group_number*] **ip** *ip_address*	Enables the HSRP. • *group_number*—(Optional) Group number on the interface for which HSRP is being activated. The default is 0. The group number range is from 0 to 255 for HSRP version 1 and from 0 to 4095 for HSRP version 2. • *ip_address*—(Optional) Virtual IP address of the HSRP standby group.
Step 24	Router(config-if)# **standby** [*group_number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime*	(Optional) Configures the time between hello packets and the hold time before other switches declare the active switch to be down. • *group_number*—(Optional) Group number to which the timers apply. • **msec**—(Optional) Interval in milliseconds. Millisecond timers allow for faster failover. • *hellotime*—Hello interval (in seconds). This is an integer from 1 to 254. The default is 3 seconds. If the **msec** option is specified, *hellotime* is in milliseconds. This is an integer from 15 to 999. • *holdtime*—Time (in seconds) before the active or standby switch is declared to be down. This is an integer from x to 255. The default is 10 seconds. If the **msec** option is specified, *holdtime* is in milliseconds. This is an integer from y to 3000.
Step 25	Router(config-if)# **standby** [*group_number*] **preempt**	Configure HSRP preemption. • *group_number*—(Optional) Group number to which this command applies. **Note** In software releases earlier than Cisco IOS Release 12.2(33)SXH, **preempt** is a keyword of the **standby priority** command. In Cisco IOS Release 12.2(33)SXH and later releases, **standby preempt** and **standby priority** are separate commands.

	Command	Purpose
Step 26	`Router(config-if)# standby [group_number] track type number [interface_priority]`	Configures the interface to track other interfaces, so that if one of the other interfaces goes down, the device's hot standby priority is lowered. • *group_number*—(Optional) Group number on the interface for which HSRP is being activated. • *type*—Interface type (combined with interface number) that will be tracked. • *number*—Interface number (combined with interface type) that will be tracked. • *interface_priority*—(Optional) Amount by which the Hot Standby priority for the switch is decremented (or incremented) when the interface goes down (or comes back up). Range is from 0 to 255. Default is 10.
Step 27	`Router(config-if)# standby [group_number] name`	Configures the standby group name for the interface. • *group_number*—(Optional) Group number to which the name is being applied. • *name*—Name of the HSRP standby group.
Step 28	`Router(config-if)# crypto map map_name ssp id`	Enables IPsec state information to be transferred by the SSP channel described by the ID. If this feature is disabled, all standby entries bound to that interface will be removed.
Step 29	`Router(config-if)# crypto engine slot slot`	Assigns the crypto engine to the inside interface VLAN. • *slot*—The slot where the IPsec VPN SPA is located.
Step 30	`Router(config-if)# interface gigabitethernet slot/subslot/port`	Enters interface configuration mode for the outside Gigabit Ethernet interface.
Step 31	`Router(config-if)# crypto connect vlan vlan_ID`	Connects the outside access port to the inside interface VLAN and enters crypto-connect mode. • *vlan_ID*—interface VLAN identifier.

For an example of IPsec stateful failover configuration using HSRP and SSP, see the "IPsec Stateful Failover Using HSRP and SSP with Crypto-Connect Mode Configuration Example" section on page 28-29.

Configuring IPsec Stateless and Stateful Failover with VRF Mode

Note Support for IPsec stateful failover is removed in Cisco IOS Release 12.2(33)SXH. The feature is supported in Release 12.2SXF.

Chassis-to- chassis failover with VRF mode is configured differently than in non-VRF (crypto-connect) mode. In VRF mode, the HSRP configuration goes on the physical interface, but the crypto map is added to the interface VLAN. In non-VRF mode, both the HSRP configuration and the crypto map are on the same interface. RRI dynamically inserts and removes routes from the active and standby switch VRF routing tables.

For a configuration example of VRF mode with stateless failover, see the "IPsec Stateless Failover Using HSRP with VRF Mode Configuration Example" section on page 28-32.

For a configuration example of VRF mode with stateful failover, see the "IPsec Stateful Failover Using HSRP with VRF Mode Configuration Example" section on page 28-33

Verifying HSRP Configurations

To verify the IPsec stateful failover HSRP configuration, use the **show crypto isakmp ha standby**, **show crypto ipsec ha**, **show crypto ipsec sa**, and **show crypto ipsec sa standby** commands.

To view your ISAKMP standby or active SAs, enter the **show crypto isakmp ha standby** command:

```
Router# show crypto isakmp ha standby

dst             src             state       I-Cookie          R-Cookie

172.16.31.100   20.3.113.1      QM_IDLE     796885F3 62C3295E  FFAFBACD EED41AFF

172.16.31.100   20.2.148.1      QM_IDLE     5B78D70F 3D80ED01  FFA03C6D 09FC50BE

172.16.31.100   20.4.124.1      QM_IDLE     B077D0A1 0C8EB3A0  FF5B152C D233A1E0

172.16.31.100   20.3.88.1       QM_IDLE     55A9F85E 48CC14DE  FF20F9AE DE37B913

172.16.31.100   20.1.95.1       QM_IDLE     3881DE75 3CF384AE  FF192CAB 795019AB
```

To view your IPsec HA Manager state, enter the **show crypto ipsec ha** command:

```
Router# show crypto ipsec ha

Interface              VIP              SAs     IPSec Ha State

GigabitEthernet5/0/1   172.16.31.100    1800    Active since 13:00:16 EDT Tue Oct 1 2002
```

To view HA status of the IPsec SA (standby or active), enter the **show crypto ipsec sa** command:

```
Router# show crypto ipsec sa

interface: GigabitEthernet5/0/1
   Crypto map tag: mymap, local addr. 172.168.3.100

    local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
    remote ident (addr/mask/prot/port): (5.6.0.0/255.255.0.0/0/0)
    current_peer: 172.168.3.1
    PERMIT, flags={}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

    local crypto endpt.: 172.168.3.100, remote crypto endpt.: 172.168.3.1
    path mtu 1500, media mtu 1500
    current outbound spi: 132ED6AB
    inbound esp sas:
    spi: 0xD8C8635F(3637011295)
    transform: esp-des esp-md5-hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 2006, flow_id: 3, crypto map: mymap
    sa timing: remaining key lifetime (k/sec): (4499/59957)
    IV size: 8 bytes
    replay detection support: Y
    HA Status: STANDBY

  inbound ah sas:
    spi: 0xAAF10A60(2867923552)
    transform: ah-sha-hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 2004, flow_id: 3, crypto map: mymap
    sa timing: remaining key lifetime (k/sec): (4499/59957)
    replay detection support: Y
    HA Status: STANDBY

  inbound pcp sas:

  outbound esp sas:
    spi: 0x132ED6AB(321836715)
    transform: esp-des esp-md5-hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 2007, flow_id: 4, crypto map: mymap
    sa timing: remaining key lifetime (k/sec): (4499/59957)
    IV size: 8 bytes
    replay detection support: Y
    HA Status: STANDBY

  outbound ah sas:
    spi: 0x1951D78(26549624)
    transform: ah-sha-hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 2005, flow_id: 4, crypto map: mymap
    ssa timing: remaining key lifetime (k/sec): (4499/59957)
    replay detection support: Y
    HA Status: STANDBY

  outbound pcp sas:
```

Enter the **show crypto ipsec sa standby** command to view your standby SAs:

```
Router# show crypto ipsec sa standby

interface: GigabitEthernet5/0/1
    Crypto map tag: mymap, local addr. 172.168.3.100

    local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
    remote ident (addr/mask/prot/port): (5.6.0.0/255.255.0.0/0/0)
    current_peer: 172.168.3.1
    PERMIT, flags={}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

    local crypto endpt.: 172.168.3.100, remote crypto endpt.: 172.168.3.1
    path mtu 1500, media mtu 1500
    current outbound spi: 132ED6AB
```

```
inbound esp sas:
spi: 0xD8C8635F(3637011295)
transform: esp-des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2006, flow_id: 3, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4499/59957)
IV size: 8 bytes
replay detection support: Y
HA Status: STANDBY

inbound ah sas:
spi: 0xAAF10A60(2867923552)
transform: ah-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2004, flow_id: 3, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4499/59957)
replay detection support: Y
HA Status: STANDBY

inbound pcp sas:

outbound esp sas:
spi: 0x132ED6AB(321836715)
transform: esp-des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2007, flow_id: 4, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4499/59957)
IV size: 8 bytes
replay detection support: Y
HA Status: STANDBY

outbound ah sas:
spi: 0x1951D78(26549624)
transform: ah-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2005, flow_id: 4, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4499/59957)
replay detection support: Y
HA Status: STANDBY

outbound pcp sas:
```

Displaying SSP Information

To verify the IPsec stateful failover SSP configuration, use the **show ssp client**, **show ssp packet**, **show ssp peers**, and **show ssp redundancy** commands.

To view SSP client information, enter the **show ssp client** command:

```
Router# show ssp client

SSP Client Information

    DOI   Client Name                      Version   Running Ver

      1   IPSec HA Manager                  1.0       1.0

      2   IKE HA Manager                    1.0       1.0
```

To view SSP packet information, enter the **show ssp packet** command:

```
Router# show ssp packet

SSP packet Information

    Socket creation time: 01:01:06

    Local port: 3249      Server port: 3249

    Packets Sent = 38559, Bytes Sent = 2285020

    Packets Received = 910, Bytes Received = 61472
```

To view SSP peer information, enter the **show ssp peers** command:

```
Router# show ssp peers

SSP Peer Information

    IP Address      Connection State   Local Interface

    40.0.0.1        Connected          FastEthernet0/1
```

To view redundancy information, enter the **show ssp redundancy** command:

```
Router# show ssp redundancy

SSP Redundancy Information

  Device has been ACTIVE for 02:55:34

    Virtual IP      Redundancy Name             Interface

    172.16.31.100   KNIGHTSOFNI                 GigabitEthernet5/0/1GigabitEthernet0/0
```

For complete configuration information for Cisco IOS IPsec stateful failover support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2/12_2y/12_2yx11/feature/guide/ft_vpnha.html

For IPsec stateful failover configuration examples, see the "IPsec Stateful Failover Using HSRP and SSP with Crypto-Connect Mode Configuration Example" section on page 28-29.

Configuring Intrachassis IPsec Stateful Failover Using a Blade Failure Group

This section describes how to configure IPsec stateful failover within a chassis using a blade failure group (BFG).

When one or more pairs of IPsec VPN SPAs are installed in a chassis, each pair can be configured as a blade failure group (BFG). The two modules do not need to reside within the same SSC. Within the BFG, each IPsec VPN SPA serves as a backup for the other IPsec VPN SPA. A BFG may be in either an active/active or an active/standby configuration.

Each IPsec tunnel is associated with only one active IPsec VPN SPA. In a BFG, the other IPsec VPN SPA will act as a backup for that IPsec tunnel. For each IKE SA or IPsec tunnel, there is an active IPsec VPN SPA and its backup. For example, in a system that supports 1000 tunnels with two IPsec VPN SPAs, 500 of the tunnels may be active on one SPA and the remaining 500 may be active on the second SPA. Both SPAs then replicate data to each other so that either one can take over in the event of a failure.

IPsec Stateful Failover Using a BFG Configuration Guidelines

When configuring IPsec stateful failover using a BFG, follow these guidelines:

- You can install or remove one of the IPsec VPN SPAs comprising a BFG without disrupting any of the tunnels on the other IPsec VPN SPA.
- We recommend deploying a BFG in an active/standby configuration to avoid oversubscription in the case of a failover.
- When deploying a BFG in an active/active configuration, we recommend that you limit each IPsec VPN SPA to no more than 50% utilization to avoid oversubscription in the case of a failover.

Configuring a BFG for IPsec Stateful Failover

To configure IPsec stateful failover using a BFG, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`redundancy`**	Enters redundancy configuration mode.
Step 2	`Router(config-red)# `**`linecard-group`** *`group_number`* **`feature-card`**	Identifies the line card group ID for a Blade Failure Group and enters redundancy line card configuration mode. • *group_number*—Specifies a group ID for the BFG.
Step 3	`Router(config-r-lc)# `**`subslot`** *`slot/subslot`*	Adds the first SPA to the group. • *slot*—Specifies the chassis slot number where the SSC is installed. • *subslot*—Specifies the secondary slot number on an SSC where a SPA is installed.
Step 4	`Router(config-r-lc)# `**`subslot`** *`slot/subslot`*	Adds the second SPA to the group.

For an IPsec stateful failover using a BFG configuration example, see the "IPsec Stateful Failover Using a Blade Failure Group Configuration Example" section on page 28-37.

Verifying the IPsec Stateful Failover Using a BFG Configuration

To verify the IPsec stateful failover using a BFG configuration, use the **show redundancy linecard group** and **show crypto ace redundancy** commands.

To display the components of a Blade Failure Group, enter the **show redundancy linecard group** command:

```
Router# show redundancy linecard-group 1

Line Card Redundancy Group:1 Mode:feature-card
Class:load-sharing
Cards:
Slot:3 Sublot:0
Slot:5 Sublot:0
```

To display information about a Blade Failure Group, enter the **show crypto ace redundancy** command:

```
Router# show crypto ace redundancy

--------------------------------------
LC Redundancy Group ID            :1
Pending Configuration Transactions:0
Current State                     :OPERATIONAL
Number of blades in the group     :2
Slots
--------------------------------------
Slot:3 subslot:0
Slot state:0x36
Booted
Received partner config
Completed Bulk Synchronization
Crypto Engine in Service
Rebooted 22 times
Initialization Timer not running
Slot:5 subslot:0
Slot state:0x36
Booted
Received partner config
Completed Bulk Synchronization
Crypto Engine in Service
Rebooted 24 times
Initialization Timer not running
```

Configuration Examples

This section provides examples of the following configurations:

- Multiple IPsec VPN SPAs in a Chassis Configuration Example, page 28-24
- IPsec Stateless Failover Using HSRP with Crypto-Connect Mode Configuration Examples, page 28-26
- IPsec Stateful Failover Using HSRP and SSP with Crypto-Connect Mode Configuration Example, page 28-29
- IPsec Stateless Failover Using HSRP with VRF Mode Configuration Example, page 28-32
- IPsec Stateful Failover Using HSRP with VRF Mode Configuration Example, page 28-33
- IPsec Stateful Failover Using a Blade Failure Group Configuration Example, page 28-37

Note The following examples use commands at the level of Cisco IOS Release 12.2(33)SXH.

As of Cisco IOS Release 12.2(33)SXH, the **crypto engine subslot** command used in previous releases has been replaced with the **crypto engine slot** command (of the form **crypto engine slot** *slot* {**inside** | **outside**}). The **crypto engine subslot** command is no longer supported. When upgrading, ensure that this command has been modified in your startup configuration to avoid extended maintenance time.

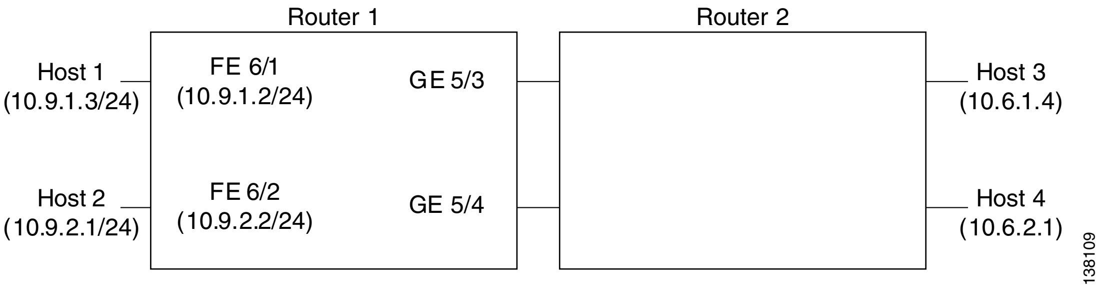
Multiple IPsec VPN SPAs in a Chassis Configuration Example

This section provides an example of a configuration using multiple IPsec VPN SPAs in a chassis as shown in Figure 28-1. Note the following in these examples:

- An IPsec VPN SPA is in slot 2, subslot 0 and slot 3, subslot 0 of router 1.
- In the configuration example, three exclamation points (!!!) precede descriptive comments.

Note In the following figure, the router with the IPsec VPN SPA could be a Cisco 7600 series router or a Catalyst 6500 series switch.

Figure 28-1 Multiple IPsec VPN SPAs in a Chassis Configuration Example



```
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key mykey address 10.8.1.1
crypto isakmp key mykey address 10.13.1.1
!
crypto ipsec transform-set xform1 ah-md5-hmac esp-des esp-sha-hmac
crypto ipsec transform-set xform2 esp-3des esp-sha-hmac
!
!!! crypto map applied to VLAN 12, which is
!!! assigned to "inside" port of IPsec VPN SPA in slot 3
crypto map cmap2 10 ipsec-isakmp
 set peer 10.8.1.1
 set transform-set xform1
 match address 102
!
!!! crypto map applied to VLAN 20, which is
!!! assigned to "inside" port of IPsec VPN SPA in slot 2/0
crypto map cmap3 10 ipsec-isakmp
 set peer 10.13.1.1
 set transform-set xform2
 match address 103
!
!!! "port" VLAN, crypto connected to VLAN 12 by IPsec VPN SPA on slot 3/0
interface Vlan11
 no ip address
 crypto connect vlan 12
!
!!! "interface" VLAN, assigned to IPsec VPN SPA on slot 3/0
interface Vlan12
 ip address 10.8.1.2 255.255.0.0
 crypto map cmap2
 crypto engine slot 3/0
!
!!! "port" VLAN, crypto connected to VLAN 20 by IPsec VPN SPA on slot 2/0
interface Vlan19
 no ip address
 crypto connect vlan 20
!
!!! "interface" VLAN, assigned to IPsec VPN SPA on slot 2/0
interface Vlan20
 ip address 10.13.1.2 255.255.0.0
 crypto map cmap3
 crypto engine slot 2/0
!
!!! connected to Host 1
interface FastEthernet6/1
 ip address 10.9.1.2 255.255.255.0
!
!!! connected to Host 2
interface FastEthernet6/2
 ip address 10.9.2.2 255.255.255.0
!
!!! connected to Router 2
interface GigabitEthernet5/3
 switchport
 switchport mode access
 switchport access vlan 11
!
!!! connected to Router 2
interface GigabitEthernet5/4
 switchport
 switchport mode access
 switchport access vlan 19
!
interface GigabitEthernet2/0/1
 no ip address
 flowcontrol receive on
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 12,1002-1005
 switchport mode trunk
 cdp enable
!
interface GigabitEthernet2/0/2
 no ip address
 flowcontrol receive on
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 11,1002-1005
 switchport mode trunk
 cdp enable
!
interface GigabitEthernet3/0/1
 no ip address
 flowcontrol receive on
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 20,1002-1005
 switchport mode trunk
 cdp enable
!
interface GigabitEthernet3/0/2
 no ip address
 flowcontrol receive on
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 19,1002-1005
 switchport mode trunk
 cdp enable
!
ip classless
!
!!! packets from Host 1 to Host 3 are routed from FastEthernet6/1
!!! to VLAN 12, encrypted with crypto map cmap2
!!! using IPsec VPN SPA in slot 3/0, and forwarded to peer 10.8.1.1
!!! through GigabitEthernet5/3
ip route 10.6.1.4 255.255.255.255 10.8.1.1
!
!!! packets from Host 2 to Host 4 are routed from FastEthernet6/2
!!! to VLAN 20, encrypted with crypto map cmap3
!!! using IPsec VPN SPA in slot 2/0, and forwarded to peer 10.13.1.1
!!! through GigabitEthernet5/4
ip route 10.6.2.1 255.255.255.255 10.13.1.1
!
!!! ACL matching traffic between Host 1 and Host 3
access-list 102 permit ip host 10.9.1.3 host 10.6.1.4
!
!!! ACL matching traffic between Host 2 and Host 4
access-list 103 permit ip host 10.9.2.1 host 10.6.2.1
```

IPsec Stateless Failover Using HSRP with Crypto-Connect Mode Configuration Examples

This section provides the following configuration examples of IPsec stateless failover using HSRP:

- IPsec Stateless Failover for the Active Chassis Configuration Example, page 28-26
- IPsec Stateless Failover for the Remote Switch Configuration Example, page 28-27

IPsec Stateless Failover for the Active Chassis Configuration Example

The following example shows the configuration for an active chassis that is configured for IPsec stateless failover using HSRP:

```
hostname router-1
!
vlan 2-1001
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 1234567890 address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set PYTHON esp-3des
!
crypto dynamic-map dynamap_1 20
 set transform-set PYTHON
 reverse-route
!
!
crypto map MONTY 1 ipsec-isakmp dynamic dynamap_1
```

```
!
interface GigabitEthernet1/3
 switchport
 switchport access vlan 502
 switchport mode access
!
interface GigabitEthernet1/4
 ip address 50.0.0.3 255.0.0.0
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 502
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 172.1.1.3 255.255.255.0
 standby ip 172.1.1.100
 standby preempt
 standby name KNIGHTSOFNI
 standby track GigabitEthernet1/3
 standby track GigabitEthernet1/4
 no mop enabled
 crypto map MONTY redundancy KNIGHTSOFNI
 crypto engine slot 4/0
!
interface Vlan502
 no ip address
 crypto connect vlan 2
!
ip route 10.0.0.0 255.0.0.0 172.1.1.4
ip route 20.0.0.0 255.0.0.0 172.1.1.4
ip route 50.0.0.0 255.0.0.0 50.0.0.13
ip route 50.0.1.1 255.255.255.255 50.0.0.13
ip route 50.0.2.1 255.255.255.255 50.0.0.13
ip route 50.0.3.1 255.255.255.255 50.0.0.13
ip route 50.0.4.1 255.255.255.255 50.0.0.13
ip route 50.0.5.1 255.255.255.255 50.0.0.13
```

IPsec Stateless Failover for the Remote Switch Configuration Example

The following example shows the configuration for a remote switch that is configured for IPsec stateless failover using HSRP.

```
hostname router-remote
!
crypto isakmp policy 1
```

```
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 12345 address 172.1.1.100
!
!
crypto ipsec transform-set ha_transform esp-3des
!
crypto map test_1 local-address Vlan2
crypto map test_1 10 ipsec-isakmp
 set peer 172.1.1.100
 set security-association lifetime seconds 86400
 set transform-set ha_transform
 set pfs group2
 match address test_1
!
interface GigabitEthernet1/1
 ip address 10.0.0.2 255.255.255.0
!
interface GigabitEthernet1/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1-2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 20.0.1.1 255.255.255.0
 crypto map test_1
 crypto engine slot 4/0
!
interface Vlan502
 no ip address
 crypto connect vlan 2
!
ip route 10.0.0.0 255.0.0.0 10.0.0.13
ip route 50.0.1.0 255.255.255.0 20.0.1.2
ip route 172.1.1.0 255.255.255.0 20.0.1.2
!
ip access-list extended test_1
 permit ip host 10.0.1.1 host 50.0.1.1
```

IPsec Stateful Failover Using HSRP and SSP with Crypto-Connect Mode Configuration Example

Note Support for IPsec stateful failover using HSRP and SSP is removed in Cisco IOS Release 12.2(33)SXH and later releases. The feature is supported in Release 12.2SXF.

Note This configuration example does not protect the SSP traffic. To protect the SSP traffic, you will need to define a new crypto map and attach it to the SSP interface without the ssp tag. The ACL for this crypto map can be derived from the remote IP address and the TCP port that are defined in the SSP group.

The following example shows the configuration for an IPsec stateful failover using HSRP and SSP:

```
hostname router-1
!
ssp group 100
 remote 50.0.0.6
 redundancy PUBLIC
 redundancy PRIVATE
!
vlan 502
!
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 1234567890 address 0.0.0.0 0.0.0.0
crypto isakmp ssp 100
!
!
!
crypto ipsec transform-set ha_transform esp-3des
!
crypto dynamic-map ha_dynamic 10
 set security-association lifetime seconds 86400
 set transform-set ha_transform
 set pfs group2
!
!
crypto map ha_dynamic 10 ipsec-isakmp dynamic ha_dynamic
!
!
!
interface GigabitEthernet1/1
 no ip address
 crypto connect vlan 502
!
interface GigabitEthernet1/2
 ip address 50.0.0.5 255.255.255.0
 load-interval 30
 no keepalive
 standby delay minimum 30 reload 60
 standby 2 ip 50.0.0.100
 standby 2 preempt
 standby 2 name PRIVATE
 standby 2 track GigabitEthernet1/1
 standby 2 track Vlan502
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,1002-1005
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan502
 ip address 172.1.1.5 255.255.255.0
 no mop enabled
 standby delay minimum 30 reload 60
 standby 1 ip 172.1.1.100
 standby 1 preempt
 standby 1 name PUBLIC
 standby 1 track GigabitEthernet1/1
 standby 1 track GigabitEthernet1/2
 crypto map ha_dynamic ssp 100
 crypto engine slot 4/0
!
ip route 10.0.0.0 255.0.0.0 172.1.1.4
ip route 20.0.0.0 255.0.0.0 172.1.1.4
ip route 50.0.0.0 255.0.0.0 50.0.0.13
```

The following example shows the configuration for a remote peer switch that is configured for IPsec stateful failover using HSRP and SSP:

```
hostname router-remote
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 12345 address 172.1.1.100
!
!
crypto ipsec transform-set ha_transform esp-3des
!
crypto map test_1 local-address Vlan2
crypto map test_1 10 ipsec-isakmp
 set peer 172.1.1.100
 set security-association lifetime seconds 86400
 set transform-set ha_transform
 set pfs group2
 match address test_1
!
```

```
crypto map test_2 local-address Vlan3
crypto map test_2 10 ipsec-isakmp
 set peer 172.1.1.100
 set security-association lifetime seconds 86400
 set transform-set ha_transform
 set pfs group2
 match address test_2
!
interface GigabitEthernet1/1
 ip address 10.0.0.2 255.255.255.0
!
interface GigabitEthernet1/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,503,1002-1005
 switchport mode trunk
 no ip address
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1-3,1002-1005
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,503,1002-1005
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan2
 ip address 20.0.1.1 255.255.255.0
 crypto map test_1
 crypto engine slot 4/0
!
interface Vlan3
 ip address 20.0.2.1 255.255.255.0
 crypto map test_2
 crypto engine slot 4/0

interface Vlan502
 no ip address
 crypto connect vlan 2
!
interface Vlan503
 no ip address
 crypto connect vlan 3
!
ip route 10.0.0.0 255.0.0.0 10.0.0.13
ip route 50.0.1.0 255.255.255.0 20.0.1.2
ip route 50.0.2.0 255.255.255.0 20.0.2.2
ip route 172.1.1.0 255.255.255.0 20.0.1.2
!
ip access-list extended test_1
```

```
 permit ip host 10.0.1.1 host 50.0.1.1
ip access-list extended test_2
 permit ip host 10.0.2.1 host 50.0.2.1
```

IPsec Stateless Failover Using HSRP with VRF Mode Configuration Example

The following example shows a VRF mode configuration with HSRP chassis-to-chassis stateless failover with crypto maps:

```
hostname router-1
!
ip vrf ivrf
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
crypto engine mode vrf
!
vlan 2,3
!
crypto keyring key1
  pre-shared-key address 14.0.1.1 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp keepalive 10
crypto isakmp profile ivrf
   vrf ivrf
   keyring key1
   match identity address 14.0.1.1 255.255.255.255
!
crypto ipsec transform-set ts esp-3des esp-sha-hmac
!
crypto map map_vrf_1 local-address Vlan3
crypto map map_vrf_1 10 ipsec-isakmp
 set peer 14.0.1.1
 set transform-set ts
 set isakmp-profile ivrf
 match address acl_1
!
interface GigabitEthernet1/1
 !switch inside port
 ip address 13.254.254.1 255.255.255.0
!
interface GigabitEthernet1/1.1
 encapsulation dot1Q 2000
 ip vrf forwarding ivrf
 ip address 13.254.254.1 255.0.0.0
!
interface GigabitEthernet1/2
 !switch outside port
 switchport
 switchport access vlan 3
 switchport mode access
!

interface GigabitEthernet4/0/1
 !IPsec VPN SPA inside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,2,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 !IPsec VPN SPA outside port
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan3
 ip address 15.0.0.2 255.255.255.0
 standby delay minimum 0 reload 0
 standby 1 ip 15.0.0.100
 standby 1 timers msec 100 1
 standby 1 priority 105
 standby 1 preempt
 standby 1 name std-hsrp
 standby 1 track GigabitEthernet1/2
 crypto engine slot 4/0 outside
!
interface Vlan2
 ip vrf forwarding ivrf
 ip address 15.0.0.252 255.255.255.0
 crypto map map_vrf_1 redundancy std-hsrp
 crypto engine slot 4/0 inside

!
ip classless
ip route 12.0.0.0 255.0.0.0 15.0.0.1
ip route 13.0.0.0 255.0.0.0 13.254.254.2
ip route 14.0.0.0 255.0.0.0 15.0.0.1
ip route 223.255.254.0 255.255.255.0 17.1.0.1
ip route vrf ivrf 12.0.0.1 255.255.255.255 15.0.0.1
!
ip access-list extended acl_1
 permit ip host 13.0.0.1 host 12.0.0.1
!
!
arp vrf ivrf 13.0.0.1 0000.0000.2222 ARPA
```

IPsec Stateful Failover Using HSRP with VRF Mode Configuration Example

Note Support for IPsec stateful failover with HSRP is removed in Cisco IOS Release 12.2(33)SXH and later releases. The feature is supported in Release 12.2SXF.

The following example shows a VRF mode configuration with HSRP chassis-to-chassis stateful failover with crypto maps:

```
hostname router-1
!
ip vrf vrf1
 rd 2000:1
 route-target export 2000:1
 route-target import 2000:1
!
ssp group 100
 remote 172.1.1.60
 redundancy PUBLIC
 redundancy PRIVATE
!
crypto engine mode vrf
!
vlan 2-1001
!
crypto keyring key1
  pre-shared-key address 0.0.0.0 0.0.0.0 key 12345
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp ssp 100
!
crypto isakmp profile prof1
   vrf vrf1
   keyring key1
   match identity address 0.0.0.0
!
!
crypto ipsec transform-set ha_transform esp-3des
!
crypto dynamic-map ha_dynamic 10
 set security-association lifetime seconds 86400
 set transform-set ha_transform
 set isakmp-profile prof1
 reverse-route
!
!
crypto map ha_dynamic local-address GigabitEthernet1/3
crypto map ha_dynamic 10 ipsec-isakmp dynamic ha_dynamic
!
!
!
interface GigabitEthernet1/2
 no ip address
!
interface GigabitEthernet1/2.1
 encapsulation dot1Q 2500
 ip vrf forwarding vrf1
 ip address 50.0.0.5 255.0.0.0
 standby delay minimum 30 reload 90
 standby 2 ip 50.0.0.100
 standby 2 preempt
 standby 2 name PRIVATE
 standby 2 track GigabitEthernet1/3
 standby 2 track Vlan100
!
interface GigabitEthernet1/3
```

```
 ip address 172.1.1.50 255.255.255.0
 standby delay minimum 30 reload 90
 standby 1 ip 172.1.1.100
 standby 1 preempt
 standby 1 name PUBLIC
 standby 1 track GigabitEthernet1/2
 standby 1 track Vlan100
 crypto engine slot 2/0
!
interface GigabitEthernet2/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,100,1002-1005
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet2/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface Vlan100
 ip vrf forwarding vrf1
 ip address 172.1.1.6 255.255.255.0
 crypto map ha_dynamic ssp 100
 crypto engine slot 2/0
!
!
ip route 10.0.0.0 255.0.0.0 172.1.1.4
ip route 20.0.0.0 255.0.0.0 172.1.1.4
ip route vrf vrf1 50.0.1.1 255.255.255.255 50.0.0.13
!
```

The following example shows the configuration for a remote peer switch that is configured for IPsec stateful failover in VRF mode:

```
hostname router-remote
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 12345 address 172.1.1.100
!
!
crypto ipsec transform-set ha_transform esp-3des
!
crypto map test_1 local-address Vlan2
crypto map test_1 10 ipsec-isakmp
 set peer 172.1.1.100
 set security-association lifetime seconds 86400
 set transform-set ha_transform
 match address test_1
```

```
!
crypto map test_2 local-address Vlan3
crypto map test_2 10 ipsec-isakmp
 set peer 172.1.1.100
 set security-association lifetime seconds 86400
 set transform-set ha_transform
 match address test_2
!
interface GigabitEthernet1/1
 ip address 10.0.0.2 255.255.255.0
!
interface GigabitEthernet1/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,503,1002-1005
 switchport mode trunk
 no ip address
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1-3,1002-1005
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502,503,1002-1005
 switchport mode trunk
 mtu 9216
 no ip address
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!

interface Vlan2
 ip address 20.0.1.1 255.255.255.0
 crypto map test_1
 crypto engine slot 4/0
!
interface Vlan3
 ip address 20.0.2.1 255.255.255.0
 crypto map test_2
 crypto engine slot 4/0
!
interface Vlan502
 no ip address
 crypto connect vlan 2
!
interface Vlan503
 no ip address
 crypto connect vlan 3
!
ip route 10.0.0.0 255.0.0.0 10.0.0.13
ip route 50.0.1.0 255.255.255.0 20.0.1.2
ip route 50.0.2.0 255.255.255.0 20.0.2.2
ip route 172.1.1.0 255.255.255.0 20.0.1.2
!
```

```
ip access-list extended test_1
 permit ip host 10.0.1.1 host 50.0.1.1
ip access-list extended test_2
 permit ip host 10.0.2.1 host 50.0.2.1
```

IPsec Stateful Failover Using a Blade Failure Group Configuration Example

The following example shows how to configure IPsec stateful failover using a Blade Failure Group (BFG):

```
Router(config)# redundancy
Router(config-red)# line-card-group 1 feature-card
Router(config-r-lc)# subslot 3/1
Router(config-r-lc)# subslot 5/1
```

CHAPTER 29

Configuring Monitoring and Accounting for the IPsec VPN SPA

This chapter provides information about configuring monitoring and accounting using the IPsec VPN SPA on the Catalyst 6500 Series switch. It includes the following sections:

Note For detailed information on Cisco IOS IPsec cryptographic operations and policies, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference*.

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* and *Cisco IOS Configuration Fundamentals Command Reference* publications.

For additional information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* and the related Cisco IOS Release 12.2 software configuration guide and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

Tip To ensure a successful configuration of your VPN using the IPsec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Overview of Monitoring and Accounting for the IPsec VPN SPA

This chapter describes some IPsec features that can be used to monitor and manage the IPsec VPN SPA. These features include:

- The IPsec VPN monitoring feature, which provides VPN session monitoring enhancements that will allow you to troubleshoot the VPN and monitor the end-user interface.
- The IPsec VPN accounting feature, which enables session accounting records to be generated by indicating when the session starts and when it stops.

- The IPsec and IKE MIB support for Cisco VRF-aware IPsec feature, which provides manageability of VPN routing and forwarding- (VRF-) aware IPsec using MIBs.

Monitoring and Managing IPsec VPN Sessions

The IPsec VPN monitoring feature provides VPN session monitoring enhancements that will allow you to troubleshoot the Virtual Private Network (VPN) and monitor the end-user interface. A crypto session is a set of IPsec connections (flows) between two crypto endpoints. If the two crypto endpoints use IKE as the keying protocol, they are IKE peers to each other. Typically, a crypto session consists of one IKE security association (for control traffic) and at least two IPsec security associations (for data traffic, one per each direction). There may be duplicated IKE security associations (SAs) and IPsec SAs or duplicated IKE SAs or IPsec SAs for the same session in the duration of rekeying or because of simultaneous setup requests from both sides.

Session monitoring enhancements include the following:

- Ability to specify an Internet Key Exchange (IKE) peer description in the configuration file
- Summary listing of crypto session status
- Syslog notification for crypto session up or down status
- Ability to clear both IKE and IPsec security associations (SAs) using one command-line interface (CLI)

Adding the Description of an IKE Peer

To add the description of an IKE peer to an IPsec VPN session, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto isakmp peer`** `{`**`ip-address`** *`ip-address`*`}`	Enables an IPsec peer for IKE querying of authentication, authorization, and accounting (AAA) for tunnel attributes in aggressive mode and enters ISAKMP peer configuration mode. • *ip-address*—IP address of the peer.
Step 2	`Router(config-isakmp-peer)#` **`description`** *`description`*	Adds a description for an IKE peer. • *description*—Description identifying the peer.

This example shows how to add a description of an IKE peer:

```
Router(config)# show crypto isakmp peer 10.2.2.9
Router(config-isakmp-peer)# description connection from site A
```

Verifying Peer Descriptions

To verify peer descriptions, enter the **show crypto isakmp peer** command:

```
Router# show crypto isakmp peer
```

```
Peer: 10.2.2.9 Port: 500
Description: connection from site A
flags: PEER_POLICY
```

When the peer at address 10.2.2.9 connects and the session comes up, the syslog status will be shown as follows:

```
%CRYPTO-5-SESSION_STATUS: Crypto tunnel is UP. Peer 10.2.2.9:500 Description: connection
from site A Id: ezvpn
```

Getting a Summary Listing of Crypto Session Status

You can get a list of all the active VPN sessions by entering the **show crypto session** command. The listing will include the following:

- Interface
- IKE peer description, if available
- IKE SAs that are associated with the peer by which the IPsec SAs are created
- IPsec SAs serving the flows of a session

Multiple IKE or IPsec SAs may be established for the same peer, in which case IKE peer descriptions will be repeated with different values for the IKE SAs that are associated with the peer and for the IPsec SAs that are serving the flows of the session.

You can also use the **show crypto session detail** variant of this command to obtain more detailed information about the sessions.

The following is sample output for the **show crypto session** command without the **detail** keyword:

```
Router# show crypto session

Crypto session current status

Interface: FastEthernet0/1
Session status: UP-ACTIVE
Peer: 172.0.0.2/500
IKE SA: local 172.0.0.1/500 remote 172.0.0.2/500 Active
IPSEC FLOW: permit ip 10.10.10.0/255.255.255.0 10.30.30.0/255.255.255.0
Active SAs: 2, origin: crypto map
```

The following is sample output using the **show crypto session** command with the **detail** keyword:

```
Router# show crypto session detail

Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.1.1.3 port 500 fvrf: (none) ivrf: (none)
Desc: this is my peer at 10.1.1.3:500 Green
Phase1_id: 10.1.1.3
IKE SA: local 10.1.1.4/500 remote 10.1.1.3/500 Active
Capabilities:(none) connid:3 lifetime:22:03:24
IPSEC FLOW: permit 47 host 10.1.1.4 host 10.1.1.3
Active SAs: 0, origin: crypto map
Inbound: #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0
IPSEC FLOW: permit ip host 10.1.1.4 host 10.1.1.3
Active SAs: 4, origin: crypto map
Inbound: #pkts dec'ed 4 drop 0 life (KB/Sec) 4605665/2949
Outbound: #pkts enc'ed 4 drop 1 life (KB/Sec) 4605665/2949
```

Syslog Notification for Crypto Session Up or Down Status

The syslog notification for crypto session up or down status function provides syslog notification every time the crypto session comes up or goes down. To enable syslog logging of the session status, enter the **crypto logging session** and **crypto logging ezvpn** commands in configuration mode.

The following is a sample syslog notification showing that a crypto session is up:

```
%CRYPTO-5-SESSION_STATUS: Crypto session is UP. Peer 10.6.6.1:500 fvrf=name10 ivrf=name20
Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

The following is a sample syslog notification showing that a crypto session is down:

```
%CRYPTO-5-SESSION_STATUS: Crypto session is DOWN. Peer 10.6.6.1:500 fvrf=name10
ivrf=name20 Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

Clearing a Crypto Session

In previous Cisco IOS software releases, there was no single command to clear both IKE and IPsec security associations (SAs). Instead, you entered the **clear crypto isakmp** command to clear IKE and the **clear crypto ipsec** command to clear IPsec. The **clear crypto session** command allows you to clear both IKE and IPsec with a single command. To clear a specific crypto session or a subset of all the sessions (for example, a single tunnel to one remote site), you must provide session-specific parameters, such as a local or remote IP address, a local or remote port, a front-door VPN routing and forwarding (FVRF) name, or an inside VRF (IVRF) name. Typically, the remote IP address will be used to specify a single tunnel to be deleted.

If a local IP address is provided as a parameter when you enter the **clear crypto session** command, all the sessions (and their IKE SAs and IPsec SAs) that share the IP address as a local crypto endpoint (IKE local address) will be cleared. If you do not provide a parameter when you enter the **clear crypto session** command, all IPsec SAs and IKE SAs in the switch will be deleted.

To clear a crypto session, enter the **clear crypto session** command in privileged EXEC mode from the switch command line. No configuration statements are required in the configuration file to use this command:

```
Router# clear crypto session
```

For complete configuration information for IPsec VPN Monitoring, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t4/feature/guide/gt_ipsvm.html

For IPsec VPN monitoring configuration examples, see the "IPsec VPN Monitoring Configuration Example" section on page 29-11.

Configuring SPAN Monitoring for the IPsec VPN SPA

You can monitor IPsec VPN SPA port traffic using the local Switched Port Analyzer (SPAN) or remote SPAN (RSPAN). By configuring two SPAN sessions, one on the inside port and one on the outside port, you can monitor clear traffic and encrypted traffic simultaneously.

For detailed information on using SPAN, see the "Configuring Local SPAN, RSPAN, and ERSPAN" chapter of the *Catalyst 6500 Release 12.2SXH and Later Software Configuration Guide* at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/span.html

Note Do not configure one IPsec VPN SPA port as a source for more than one SPAN session.

Configuring a SPAN Session

To configure a local SPAN session using an IPsec VPN SPA port as a source, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# monitor session session_number source [interface type slot/subslot/port | vlan vlan_number] rx`	Associates the local SPAN source session number with the source port or VLAN and selects the traffic direction to be monitored. - *session_number*—A user-defined identifying number for the session. Range is 1 to 66.
Step 2	`Router(config)# monitor session session_number destination interface type slot/port`	Specifies the destination for local SPAN session traffic.

This example shows how to configure a local SPAN session to capture inbound traffic before decryption from an IPsec VPN SPA in subslot 0 of module 2 and send the captured traffic to port 16 of module 5:

```
Router(config)# monitor session 1 source interface gi2/0/2 tx
Router(config)# monitor session 1 destination interface gi5/16
```

This example shows how to capture inbound traffic after decryption:

```
Router(config)# monitor session 1 source interface gi2/0/1 rx
```

This example shows how to capture outbound traffic before encryption:

```
Router(config)# monitor session 1 source interface gi2/0/1 tx
```

This example shows how to capture outbound traffic after encryption:

```
Router(config)# monitor session 1 source interface gi2/0/2 rx
```

Configuring IPsec VPN Accounting

The IPsec VPN accounting feature enables session accounting records to be generated by indicating when the session starts and when it stops.

A VPN session is defined as an Internet Key Exchange (IKE) security association (SA) and the one or more SA pairs that are created by the IKE SA. The session starts when the first IP Security (IPsec) pair is created and stops when all IPsec SAs are deleted. If IPsec accounting is configured, after IKE phases are complete, an accounting start record is generated for the session. New accounting records are not generated during a rekeying.

Session-identifying information and session-usage information is passed to the Remote Authentication Dial-In User Service (RADIUS) server by standard RADIUS attributes and vendor-specific attributes (VSAs).

To enable IPsec VPN accounting, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables periodic interim accounting records to be sent to the accounting server.
Step 2	`Router(config)# aaa authentication login list-name group radius`	Sets authentication, authorization, and accounting (AAA) authentication at login using RADIUS servers. • *list-name*—Character string used to name the list of authentication methods activated when a user logs in. • **group radius**—Uses the list of all RADIUS servers for authentication.
Step 3	`Router(config)# aaa authorization network list-name group radius`	Runs authorization for all network-related service requests, including Serial Line Internet Protocol (SLIP), PPP, PPP Network Control Programs (NCPs), and AppleTalk Remote Access (ARA). • *list-name*—Character string used to name the list of authorization methods activated when a user logs in. • **group radius**—Uses the list of all RADIUS servers for authentication.
Step 4	`Router(config)# aaa accounting network list-name start-stop [broadcast] group radius`	Enables AAA accounting of network-related requested services for billing or security purposes when you use RADIUS. • *list-name*—Character string used to name the list of the accounting methods. • **start-stop**—Sends a start accounting notice at the beginning of a process and a stop accounting notice at the end of a process. The start accounting record is sent in the background. The requested user process begins regardless of whether the start accounting notice was received by the accounting server. • **broadcast**—(Optional) Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group. • **group radius**—Uses the list of all RADIUS servers for authentication as defined by the **aaa group server radius** command.

	Command	Purpose
Step 5	`Router(config)# aaa accounting update periodic minutes`	(Optional) Sends accounting updates to the accounting server while a session is up. • *minutes* — Specifies the interval (in number of minutes) at which accounting records are to be sent to the accounting server.
Step 6	`Router(config)# aaa session-id common`	Specifies whether the same session ID will be used for each AAA accounting service type within a call or whether a different session ID will be assigned to each accounting service type. • **common**—Ensures that all session identification (ID) information that is sent out for a given call will be made identical. The default behavior is common.
Step 7	`Router(config)# crypto isakmp profile profile-name`	Audits IP security (IPsec) user sessions and enters isakmp-profile configuration mode. • *profile-name*—Name of the user profile. To associate a user profile with the RADIUS server, the user profile name must be identified.
Step 8	`Router(conf-isa-prof)# vrf ivrf`	Associates the on-demand address pool with a Virtual Private Network (VPN) routing and forwarding (VRF) instance name. • *ivrf*—VRF to which the IPsec tunnel will be mapped.
Step 9	`Router(conf-isa-prof)# match identity group group-name`	Matches an identity from a peer in an ISAKMP profile. • *group-name*—A unity group that matches identification (ID) type ID_KEY_ID. If unity and main mode Rivest, Shamir, and Adelman (RSA) signatures are used, the *group-name* argument matches the Organizational Unit (OU) field of the Distinguished Name (DN).
Step 10	`Router(conf-isa-prof)# client authentication list list-name`	Configures Internet Key Exchange (IKE) extended authentication (XAUTH) in an Internet Security Association and Key Management Protocol (ISAKMP) profile. • *list-name*—Character string used to name the list of authentication methods activated when a user logs in. The list name must match the list name that was defined during the authentication, authorization, and accounting (AAA) configuration.

	Command	Purpose
Step 11	`Router(conf-isa-prof)# isakmp authorization list list-name`	Configures an IKE shared secret and other parameters using the AAA server in an ISAKMP profile. The shared secret and other parameters are generally pushed to the remote peer via mode configuration (MODECFG). • *list-name*—AAA authorization list used for configuration mode attributes or preshared keys for aggressive mode.
Step 12	`Router(conf-isa-prof)# client configuration address [initiate | respond]`	Configures IKE mode configuration (MODECFG) in the ISAKMP profile. • **initiate**—(Optional) Switch will attempt to set IP addresses for each peer. • **respond**—(Optional) Switch will accept requests for IP addresses from any requesting peer.
Step 13	`Router(conf-isa-prof)# accounting list-name`	Enables AAA accounting services for all peers that connect via this ISAKMP profile. • *list-name*— Name of a client accounting list.
Step 14	`Router(conf-isa-prof)# exit`	Exits isakmp profile configuration mode and returns to global configuration mode.
Step 15	`Router(config)# crypto dynamic-map dynamic-map-name dynamic-seq-num`	Creates a dynamic crypto map template and enters the crypto map configuration command mode. • *dynamic-map-name*—Name of the dynamic crypto map set that should be used as the policy template. • *dynamic-seq-num*—Sequence number you assign to the dynamic crypto map entry.
Step 16	`Router(config-crypto-map)# set transform-set transform-set-name`	Specifies which transform sets can be used with the crypto map template. A transform set defines IPsec security protocols and algorithms. Transform sets and their accepted values are described in the *Cisco IOS Security Command Reference*. • *transform-set-name*—Name of the transform set.
Step 17	`Router(config-crypto-map)# set isakmp-profile profile-name`	Sets the ISAKMP profile name. • *profile-name*—Name of the ISAKMP profile.
Step 18	`Router(config-crypto-map)# reverse-route [remote-peer]`	Allows routes (IP addresses) to be injected for destinations behind the VPN remote tunnel endpoint and may include a route to the tunnel endpoint itself (using the **remote-peer** keyword for the crypto map). • **remote-peer**—(Optional) Routes of public IP addresses and IP security (IPsec) tunnel destination addresses are inserted into the routing table.

	Command	Purpose
Step 19	`Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.
Step 20	`Router(config)# crypto map map-name ipsec-isakmp dynamic dynamic-map-name`	Creates a crypto profile that provides a template for configuration of dynamically created crypto maps. • *map-name*—Name that identifies the crypto map set. • *dynamic-map-name*—Name of the dynamic crypto map set that should be used as the policy template.
Step 21	`Router(config)# radius-server host ip-address [auth-port auth-port-number] [acct-port acct-port-number]`	Specifies a RADIUS server host. • *ip-address* —IP address of the RADIUS server host. • *auth-port-number*—(Optional) UDP destination port number for authentication requests; the host is not used for authentication if set to 0. If unspecified, the port number defaults to 1645. • *acct-port-number*—(Optional) UDP destination port number for accounting requests; the host is not used for accounting if set to 0. If unspecified, the port number defaults to 1646.
Step 22	`Router(config)# radius-server key string`	Sets the authentication and encryption key for all RADIUS communications between the switch and the RADIUS daemon. • *string*—The unencrypted (cleartext) shared key.
Step 23	`Router(config)# interface type slot/[subslot]/port`	Configures an interface type and enters interface configuration mode. • *slot*/[*subslot*]/ *port*—Number of the slot, subslot (optional), and port to be configured.
Step 24	`Router(config-if)# crypto map map-name`	Applies a previously defined crypto map set to an interface. • *map-name*—Name that identifies the crypto map set.

For complete configuration information for IPsec VPN Accounting, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ft_evpna.html

For IPsec VPN accounting configuration examples, see the "IPsec VPN Accounting Configuration Example" section on page 29-10.

Configuration Examples

This section provide examples of the following configurations:

- IPsec VPN Accounting Configuration Example, page 29-10

- IPsec VPN Monitoring Configuration Example, page 29-11

Note The following examples use commands at the level of Cisco IOS Release 12.2(33)SXH.

As of Cisco IOS Release 12.2(33)SXH, the **crypto engine subslot** command used in previous releases has been replaced with the **crypto engine slot** command (of the form **crypto engine slot** *slot* {**inside** | **outside**}). The **crypto engine subslot** command is no longer supported. When upgrading, ensure that this command has been modified in your start-up configuration to avoid extended maintenance time.

IPsec VPN Accounting Configuration Example

The following example shows how to enable the IPsec VPN accounting feature:

```
aaa new-model
!
!
aaa group server radius r1
 server-private 10.30.1.52 auth-port 1812 acct-port 1813 key allegro
!
aaa authentication login test_list group r1
aaa authorization network test_list group r1
aaa accounting update periodic 10 jitter maximum 0
aaa accounting network test_list start-stop group r1!
!
ip vrf ivrf1
 rd 1:2
!
crypto engine mode vrf
!
crypto isakmp policy 5
 encr 3des
 authentication pre-share
 group 2
 lifetime 14400
!
crypto isakmp client configuration group test
 key world
 pool pool1
!
crypto isakmp profile test_pro
   vrf ivrf1
   match identity group test
   client authentication list test_list
   isakmp authorization list test_list
   client configuration address respond
   accounting test_list
!
crypto ipsec transform-set t3 esp-3des esp-sha-hmac
!
!
crypto dynamic-map dyn-ra 10
 set transform-set t3
 set isakmp-profile test_pro
 reverse-route
!
!
crypto map map-ra local-address GigabitEthernet3/15
crypto map map-ra 1 ipsec-isakmp dynamic dyn-ra
!
```

```
!
interface GigabitEthernet1/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,100,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
interface GigabitEthernet1/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust ip-precedence
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast edge trunk
!
!

interface GigabitEthernet3/15
 mtu 9216
 ip address 120.0.0.254 255.255.255.0
crypto engine outside
!
!
!
interface Vlan100
 ip vrf forwarding ivrf1
 ip address 120.0.0.100 255.255.255.0
 ip flow ingress
 crypto map map-ra
crypto engine slot 1/0 inside
!
!
!
ip local pool pool1 100.0.1.1 100.0.5.250
```

IPsec VPN Monitoring Configuration Example

The following example shows how to configure an IKE peer for IPsec VPN monitoring:

```
!
upgrade fpd auto
version 12.2
service timestamps debug datetime
service timestamps log datetime
no service password-encryption
service counters max age 5
!
hostname Ez-DCM-CC
!
boot-start-marker
boot system disk1:s72033-adventerprisek9_wan-mz.122-33.SXH
boot-end-marker
!
```

```
logging buffered 1000000 debugging
enable secret 5 $1$i5FZ$47ybx5dEaUKc3eRaDIZ/z.
!
username cisco password 0 cisco
username t1 password 0 t1
username t2 password 0 t2
username t3 password 0 t3
username t4 password 0 t4
username t5 password 0 t5
username t6 password 0 t6
username t7 password 0 t7
username t8 password 0 t8
username user1 password 0 letmein
aaa new-model
aaa authentication login myuserlist local
aaa authorization network myuserlist local
!
aaa session-id common
clock timezone PST -7
call-home
  alert-group configuration
  alert-group diagnostic
  alert-group environment
  alert-group inventory
  alert-group syslog
 profile "CiscoTAC-1"
   no active
   no destination transport-method http
   destination transport-method email
   destination address email callhome@cisco.com
   destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService
   subscribe-to-alert-group diagnostic severity minor
   subscribe-to-alert-group environment severity minor
   subscribe-to-alert-group syslog severity major pattern ".*"
   subscribe-to-alert-group configuration periodic monthly 10 15:08
   subscribe-to-alert-group inventory periodic monthly 10 14:53
ip subnet-zero
!
no ip domain-lookup
ip domain-name cisco.com
ipv6 mfib hardware-switching replication-mode ingress
vtp mode transparent
no mls acl tcam share-global
mls netflow interface
no mls flow ip
no mls flow ipv6
mls cef error action freeze
!
redundancy
 keepalive-enable
 mode sso
 linecard-group 0 feature-card
  class load-sharing
  subslot 4/0
 main-cpu
  auto-sync running-config
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
diagnostic monitor syslog
diagnostic cns publish cisco.cns.device.diag_results
diagnostic cns subscribe cisco.cns.device.diag_commands
!
power redundancy-mode combined
```

```
port-channel per-module load-balance
!
vlan internal allocation policy descending
vlan access-log ratelimit 2000
!
vlan 2-3,16-17
!
crypto logging session
crypto logging ezvpn
!
crypto logging ezvpn group mygroup
!
crypto isakmp policy 10
 encr aes
 authentication pre-share
 group 2
 lifetime 43200
crypto isakmp key WorldCup2006 address 0.0.0.0 0.0.0.0
!
crypto isakmp client configuration group mygroup
 key mykey
 pool mypool
!
crypto isakmp peer address 16.0.0.3
 description first-ezvpn-client
!
crypto isakmp peer address 16.0.0.4
 description second-ezvpn-client
!
crypto ipsec security-association lifetime seconds 21600
!
crypto ipsec transform-set MyTranSet esp-aes esp-sha-hmac
no crypto ipsec nat-transparency udp-encaps
!
crypto call admission limit ike in-negotiation-sa 10
!
crypto dynamic-map DynMap1 10
 set transform-set MyTranSet
 reverse-route
!
crypto map MyMap1 client authentication list myuserlist
crypto map MyMap1 isakmp authorization list myuserlist
crypto map MyMap1 client configuration address respond
crypto map MyMap1 500 ipsec-isakmp dynamic DynMap1
!
interface GigabitEthernet1/25
 no ip address
 crypto connect vlan 16
!
interface GigabitEthernet1/27
 no ip address
 crypto connect vlan 17
!
interface GigabitEthernet1/29
 ip address 26.0.0.2 255.255.255.0
!
interface GigabitEthernet4/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 16,17,1002-1005
 switchport mode trunk
 mtu 9216
 mls qos vlan-based
 mls qos trust cos
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet4/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1002-1005
 switchport mode trunk
 mtu 9216
 mls qos trust cos
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet5/2
 ip address 44.0.111.114 255.0.0.0
 media-type rj45
!
interface Vlan1
 no ip address
 ip flow ingress
 ip igmp snooping querier
 shutdown
!
interface Vlan16
 ip address 16.0.0.2 255.255.224.0
 no mop enabled
 crypto map MyMap1
 crypto engine slot 4/0
!
interface Vlan17
 ip address 16.0.32.2 255.255.224.0
 no mop enabled
 crypto map MyMap1
 crypto engine slot 4/0
!
ip local pool mypool 36.0.0.1 36.0.15.254
ip local pool mypool 36.0.16.1 36.0.31.254
ip local pool mypool 36.0.32.1 36.0.47.254
ip local pool mypool 36.0.48.1 36.0.63.254
ip default-gateway 44.0.100.1
ip classless
ip route 43.0.0.0 255.0.0.0 44.0.100.1
ip route 45.0.0.0 255.0.0.0 44.0.100.1
ip route 223.255.254.53 255.255.255.255 44.0.100.1
ip route 223.255.254.54 255.255.255.255 44.0.100.1
!
no ip http server
no ip http secure-server
!
radius-server source-ports 1645-1646
!
control-plane
!
dial-peer cor custom
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password cisco
 transport input lat pad mop udptn telnet rlogin ssh nasi acercon
line vty 5 15
 transport input lat pad mop udptn telnet rlogin ssh nasi acercon
```

```
!
monitor event-trace platform cmfi lc agg-label
monitor event-trace platform cmfi lc error
ntp clock-period 17280219
ntp update-calendar
ntp server 223.255.254.254
ntp server 223.255.254.53
mac-address-table aging-time 0
!
end
```

CHAPTER 30

Troubleshooting the IPsec VPN SPA

This chapter describes techniques that you can use to troubleshoot the operation of your IPsec VPN SPAs in a Catalyst 6500 Series switch.

It includes the following sections:

- General Troubleshooting Information, page 30-1
- Monitoring the IPsec VPN SPA, page 30-3
- Troubleshooting Specific Problems on the IPsec VPN SPA, page 30-24
- Using Crypto Conditional Debug, page 30-27
- Preparing for Online Insertion and Removal of a SPA, page 30-30

Note For detailed information on Cisco IOS IPsec cryptographic operations and policies, refer to the *Cisco IOS Security Configuration Guide, Release 12.2* and *Cisco IOS Security Command Reference, Release 12.2*.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xlv.

General Troubleshooting Information

This section describes general information for troubleshooting the IPsec VPN SPA and the Cisco 7600 SSC-400 SIP. It includes the following sections:

- Interpreting Console Error Messages, page 30-2
- Using debug Commands, page 30-2
- Using show Commands, page 30-2

Interpreting Console Error Messages

The Catalyst 6500 Series switch can generate error messages and other system messages to inform the operator of events that might require attention. These messages can be displayed on the console, or sent to a logging host using the System Logging (Syslog) protocol or Simple Network Management Protocol (SNMP).

System error messages are organized in the documentation according to the particular system facility that produces the messages. The IPsec VPN SPA and Cisco 7600 SSC-400 SIP use the following facility names in error messages:

- IPsec VPN SPA—SPA_IPSEC_2G (also VPNSPA)
- Cisco 7600 SSC-400—CAT6000_SSC (also C7600_SSC400)

To view the explanations and recommended actions for Catalyst 6500 Series switch error messages, including messages related to service modules, refer to the following documents:

- *Cisco IOS Release 12.2SX System Message Guide* at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2sx/system/messages/122sxsms.html
- *System Messages for 12.2S* (for error messages in Release 12.2S) at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2s/system/messages/122sdebu.html

Using debug Commands

For information about **debug** commands specific to the Cisco IOS software release 12.2SX, see the *Cisco IOS Master Command List, Release 12.2SX* at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/122sxmcl/12_2sx_mcl_book.html

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support personnel. We recommend that you use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

For information about available crypto conditional debugging commands, see the "Using Crypto Conditional Debug" section on page 30-27.

For more information about other **debug** commands that can be used on a Catalyst 6500 Series switch, see the *Cisco IOS Debug Command Reference, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/debug/command/reference/122debug.html

Using show Commands

You can use several **show** commands to monitor and troubleshoot the IPsec VPN SPA on the Catalyst 6500 Series switch.

For more information about **show** commands to verify and monitor the IPsec VPN SPA, see the "Displaying IPsec VPN SPA Configuration Information" section on page 30-6 and the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX*.

For more information about security-related **show** commands, see the *Cisco IOS Security Command Reference*.

Monitoring the IPsec VPN SPA

This section describes commands that can be used to display information about the IPsec VPN SPA hardware and configuration. It consists of the following subsections:

- Displaying IPsec VPN SPA Hardware and System Information, page 30-3
- Displaying IPsec VPN SPA Configuration Information, page 30-6

Displaying IPsec VPN SPA Hardware and System Information

To display hardware and system information, use the following commands:

- **show diagbus, show module, show crypto eli**—See the "Displaying Information About IPsec VPN SPA Ports" section on page 30-3.
- **show crypto engine accelerator statistic slot**—See the "Displaying Platform and Network Interface Controller Statistics for the IPsec VPN SPA" section on page 30-4.
- **show hw-module slot fpd**—See the "Displaying Information About Hardware Revision Levels" section on page 30-6.

Displaying Information About IPsec VPN SPA Ports

To display information about the type of SPAs that are installed in the switch, use the **show diagbus** command.

The following example shows output from the **show diagbus** command on a Catalyst 6500 Series switch with an IPsec VPN SPA installed in subslot 1 of a Cisco 7600 SSC-400 that is installed in slot 5:

```
Router# show diagbus

Slot 5: Logical_index 10
        2-subslot Services SPA Carrier-400 controller
        Board is analyzed ipc ready
        HW rev 0.3, board revision A01
        Serial Number: abc Part number: 73-6348-01

        Slot database information:
        Flags: 0x2004   Insertion time: 0x3DB5F4BC (4d20h ago)

        Controller Memory Size:
                248 MBytes CPU Memory
                8 MBytes Packet Memory
                256 MBytes Total on Board SDRAM
        IOS (tm) cwlc Software (smsc-DWDBG-M), Experimental Version 12.2(20050623:231413)

        SPA Information:
        subslot 5/1: SPA-IPSEC-2G (0x3D7), status: ok
```

For information about the **show module** and **show crypto eli** commands, see the "Displaying the SPA Hardware Type" section on page 20-20.

Displaying Platform and Network Interface Controller Statistics for the IPsec VPN SPA

To display platform statistics and optionally display network interface controller statistics, use the **show crypto engine accelerator statistic slot** command.

Note The **show crypto engine accelerator statistic** command is supported in Cisco IOS Release 12.2(33)SXH and later releases.

The following example shows output from the **show crypto engine accelerator statistic** command on a Catalyst 6500 Series switch with an IPsec VPN SPA in subslot 0 of a Cisco 7600 SSC-400 that is installed in slot 1. The output displays platform statistics for the IPsec VPN SPA and also displays the network interface controller statistics.

```
Router# show crypto engine accelerator statistic slot 1/0 detail

VPN module in slot 1/0


Decryption Side Data Path Statistics
====================================
Packets RX...............: 454260
Packets TX...............: 452480

IPSec Transport Mode.....: 0
IPSec Tunnel Mode........: 452470
AH Packets...............: 0
ESP Packets..............: 452470
GRE Decapsulations.......: 0
NAT-T Decapsulations.....: 0
Clear....................: 8
ICMP.....................: 0

Packets Drop.............: 193
Authentication Errors....: 0
Decryption Errors........: 0
Replay Check Failed......: 0
Policy Check Failed......: 0
Illegal CLear Packet.....: 0
GRE Errors...............: 0
SPD Errors...............: 0
HA Standby Drop..........: 0

Hard Life Drop...........: 0
Invalid SA...............: 191
SPI No Match.............: 0
Destination No Match.....: 0
Protocol No Match........: 0

Reassembly Frag RX.......: 0
IPSec Fragments..........: 0
IPSec Reasm Done.........: 0
Clear Fragments..........: 0
Clear Reasm Done.........: 0
Datagrams Drop...........: 0
Fragments Drop...........: 0
```

```
Decryption Side Controller Statistics

=====================================
Frames RX................: 756088
Bytes RX.................: 63535848
Mcast/Bcast Frames RX....: 2341
RX Less 128Bytes.........: 756025
RX Less 512Bytes.........: 58
RX Less 1KBytes..........: 2
RX Less 9KBytes..........: 3
RX Frames Drop...........: 0

Frames TX................: 452365
Bytes TX.................: 38001544
Mcast/Bcast Frames TX....: 9
TX Less 128Bytes.........: 452343
TX Less 512Bytes.........: 22
TX Less 1KBytes..........: 0
TX Less 9KBytes..........: 0


Encryption Side Data Path Statistics
====================================
Packets RX...............: 756344
Packets TX...............: 753880
IPSec Transport Mode.....: 0
IPSec Tunnel Mode........: 753869
GRE Encapsulations.......: 0
NAT-T Encapsulations.....: 0
LAF prefragmented........: 0

Fragmented...............: 0
Clear....................: 753904
ICMP.....................: 0

Packets Drop.............: 123
IKE/TED Drop.............: 27
Authentication Errors....: 0
Encryption Errors........: 0
HA Standby Drop..........: 0

Hard Life Drop...........: 0
Invalid SA...............: 191

Reassembly Frag RX.......: 0
Clear Fragments..........: 0
Clear Reasm Done.........: 0
Datagrams Drop...........: 0
Fragments Drop...........: 0


Encryption Side Controller Statistics
=====================================
Frames RX................: 454065
Bytes RX.................: 6168274/
Mcast/Bcast Frames RX....: 1586
RX Less 128Bytes.........: 1562
RX Less 512Bytes.........: 452503
RX Less 1KBytes..........: 0
RX Less 9KBytes..........: 0
RX Frames Drop...........: 0

Frames TX................: 753558
```

```
Bytes TX.................: 100977246
Mcast/Bcast Frames TX....: 2
TX Less 128Bytes.........: 3
TX Less 512Bytes.........: 753555
TX Less 1KBytes..........: 0
TX Less 9KBytes..........: 0

Router#
```

Displaying Information About Hardware Revision Levels

To display information about the hardware revision of the Cisco 7600 SSC-400 and the IPsec VPN SPA as well as the version of the field-programmable devices (FPDs) that are on the carrier card and the SPA, use the **show hw-module slot fpd** command. Cisco technical engineers might need this information to debug or troubleshoot problems with a SPA installation.

The following example shows output from the **show hw-module slot** command on a Catalyst 6500 Series switch with an IPsec VPN SPA installed in subslot 0 of a Cisco 7600 SSC-400 that is installed in slot 6:

```
Router# show hw-module slot 2 fpd

==== ====================== ====== =============================================
                             H/W   Field Programmable   Current   Min. Required
Slot Card Type               Ver.  Device: "ID-Name"    Version      Version
==== ====================== ====== ================== =========== ==============
   2 7600-SSC-400              0.5 1-I/O FPGA             1.0         1.0
---- ---------------------- ------ ------------------ ----------- --------------
 2/0 SPA-IPSEC-2G              0.3 1-PROM                 1.1         1.1
==== ====================== ====== =============================================
```

Displaying IPsec VPN SPA Configuration Information

To display information about the IPsec VPN SPA configuration, use the following commands:

- **show crypto vlan**—See the "Displaying Information About Access and Routed Ports That Are Connected" section on page 30-7, "Displaying the VPN Running State" section on page 30-8, and "Displaying Information About IP Multicast Over a GRE Tunnel" section on page 30-23.
- **show interfaces trunk**—See the "Displaying Information About the VLANs Allowed by a Trunk Port" section on page 30-7.
- **show crypto isakmp policy**—See the "Displaying Information About IKE Policies" section on page 30-8.
- **show crypto ipsec transform-set**—See the "Displaying Information About IPsec Transform Sets" section on page 30-9.
- **show crypto map**—See the "Displaying Information About Crypto Maps" section on page 30-9.
- **show crypto isakmp sa**—See the "Displaying Information About SAs at a Peer" section on page 30-11.
- **show crypto isakmp ha standby**—See the "Displaying HSRP Information" section on page 30-11.
- **show crypto ipsec ha**—See the "Displaying HSRP Information" section on page 30-11.
- **show crypto ipsec sa**—See the "Displaying Information About IPsec Security Associations" section on page 30-9 and the "Displaying HSRP Information" section on page 30-11.
- **show crypto ipsec sa standby**—See the "Displaying HSRP Information" section on page 30-11.
- **show ssp client**—See the "Displaying SSP Information" section on page 30-14.

- **show ssp packet**—See the "Displaying SSP Information" section on page 30-14.
- **show ssp peers**—See the "Displaying SSP Information" section on page 30-14.
- **show ssp redundancy**—See the "Displaying SSP Information" section on page 30-14.
- **show redundancy linecard-grou**p—See the "Displaying Information About a BFG Configuration" section on page 30-15.
- **show crypto ace redundancy**—See the "Displaying Information About a BFG Configuration" section on page 30-15.
- **show crypto key mypubkey rsa**—See the "Displaying Information About RSA Public Keys" section on page 30-15.
- **show crypto key pubkey-chain rsa**—See the "Displaying Information About RSA Public Keys" section on page 30-15.
- **show crypto pki certificates**—See the "Displaying Information About Certificates" section on page 30-16.
- **show crypto pki trustpoints**—See the "Displaying Information About Trustpoints" section on page 30-17.
- **show ip nhrp**—See the "Displaying Information About the NHRP Cache" section on page 30-18.
- **show crypto session**—See the "Displaying Information About Crypto Sessions" section on page 30-18.
- **show interfaces tunnel**—See the "Displaying Tunnel Interface Information" section on page 30-19.

For a detailed description of the information displayed by the **show** commands, refer to the "IP Security and Encryption" chapter of the *Cisco IOS Security Command Reference*.

Displaying Information About Access and Routed Ports That Are Connected

To verify that an access or routed port is connected, use the **show crypto vlan** command. The following is sample output from the command:

```
Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port GigabitEthernet2/0/1 connected to VLAN 502
with crypto map set mymap1


Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port GigabitEthernet2/0/1 connected to Gi2/8 with
crypto map set mymap2
```

Displaying Information About the VLANs Allowed by a Trunk Port

To display information about the VLANs allowed by a trunk port, use the **show interfaces trunk** command. The following is sample output from the command:

```
Router# show interfaces trunk

Port                Mode         Encapsulation  Status        Native vlan
Gi2/0/1             on           802.1q         trunking      1
Gi2/0/2             on           802.1q         trunking      1

Port                Vlans allowed on trunk
```

```
Gi2/0/1             2
Gi2/0/2             502

Port                Vlans allowed and active in management domain
Gi2/0/1             2
Gi2/0/2             502

Port                Vlans in spanning tree forwarding state and not pruned
Gi2/0/1             2
Gi2/0/2             502
```

Displaying the VPN Running State

To display the VPN running state, use the **show crypto vlan** command. The following is sample output from the command:

In the following example, the interface VLAN belongs to the IPsec VPN SPA inside port:

```
Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port GigabitEthernet2/0/1 connected to Fa8/3
```

In the following example, VLAN 2 is the interface VLAN and VLAN 2022 is the hidden VLAN:

```
Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port GigabitEthernet2/0/1 connected to VLAN 2022
with crypto map set mymap2
```

In the following example, either the interface VLAN is missing on the IPsec VPN SPA inside port, the IPsec VPN SPA is removed from the chassis, or the IPsec VPN SPA was moved to a different subslot:

```
Router# show crypto vlan

  Interface VLAN 2 connected to VLAN 3 (no IPSec Service Module attached)
```

Displaying Information About IKE Policies

To display information about IKE policies, use the **show crypto isakmp policy** command. The following is sample output from the command:

```
Router# show crypto isakmp policy

Global IKE policy
Protection suite of priority 1
        encryption algorithm:   Three key triple DES
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit
Default protection suite
        encryption algorithm:   DES - Data Encryption Standard (56 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit
```

Note If a user enters an IKE encryption method that the hardware does not support, a warning message will be displayed in the **show crypto isakmp policy** command output:

```
WARNING:encryption hardware does not support the configured encryption method for ISAKMP
policy value
```

Displaying Information About IPsec Transform Sets

To display information about transform set configurations, use the **show crypto ipsec transform-set** command. The following is sample output from the command:

```
Router# show crypto ipsec transform-set

Transform set combined-des-md5: {esp-des esp-md5-hmac}
will negotiate = {Tunnel,},
Transform set t1: {esp-des esp-md5-hmac}
will negotiate = {Tunnel,},
Transform set t100: {ah-sha-hmac}
will negotiate = {Transport,},
```

Note If a user enters an IPsec transform that the hardware (the IPsec peer) does not support, a warning message will be displayed in the **show crypto ipsec transform-set** command output:

```
WARNING:encryption hardware does not support transform.
```

Displaying Information About Crypto Maps

To display information about crypto map configurations, use the **show crypto map** command. The following is sample output from the command:

```
Router# show crypto map

Crypto Map "test" 10 ipsec-isakmp
        Peer = 11.1.0.1
        Extended IP access list 101
            access-list 101 permit ip host 1.0.0.1 host 2.0.0.1
        Current peer: 11.1.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                tset:  { esp-3des  } ,
        }
        Interfaces using crypto map test:
                Vlan2
 using crypto engine SPA-IPSEC-2G[2/0]
```

Displaying Information About IPsec Security Associations

To display information about IPsec security associations, use the **show crypto ipsec sa** command.

Note When you first enter the **show crypto ipsec sa** command, the packet counters will not show the correct values. Subsequent instances of the command will display the correct values.

The following is sample output from the command:

```
Router# show crypto ipsec sa

interface: Ethernet0

    Crypto map tag: router-alice, local addr. 172.21.114.123
    local ident (addr/mask/prot/port): (172.21.114.123/255.255.255.255/0/0)
    remote ident (addr/mask/prot/port): (172.21.114.67/255.255.255.255/0/0)
    current_peer: 172.21.114.67
    PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify 10
    #send errors 10, #recv errors 0

    local crypto endpt.: 172.21.114.123, remote crypto endpt.: 172.21.114.67
    path mtu 1500, media mtu 1500
    current outbound spi: 20890A6F

    inbound esp sas:
    spi: 0x257A1039(628756537)
        transform: esp-des esp-md5-hmac,
        in use settings ={Tunnel,}
        slot: 0, conn id: 26, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y

    inbound ah sas:

    outbound esp sas:
    spi: 0x20890A6F(545852015)
        transform: esp-des esp-md5-hmac,
        in use settings ={Tunnel,}
        slot: 0, conn id: 27, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y
     outbound ah sas:

interface: Tunnel0

    Crypto map tag: router-alice, local addr. 172.21.114.123
    local ident (addr/mask/prot/port): (172.21.114.123/255.255.255.255/0/0)
    remote ident (addr/mask/prot/port): (172.21.114.67/255.255.255.255/0/0)
    current_peer: 172.21.114.67
    PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify 10
    #send errors 10, #recv errors 0
    local crypto endpt.: 172.21.114.123, remote crypto endpt.: 172.21.114.67
    path mtu 1500, media mtu 1500
    current outbound spi: 20890A6F

    inbound esp sas:
    spi: 0x257A1039(628756537)
        transform: esp-des esp-md5-hmac,
        in use settings ={Tunnel,}
        slot: 0, conn id: 26, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y

    inbound ah sas:

    outbound esp sas:
    spi: 0x20890A6F(545852015)
        transform: esp-des esp-md5-hmac,
        in use settings ={Tunnel,}
        slot: 0, conn id: 27, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y

    outbound ah sas:
```

Displaying Information About SAs at a Peer

To display information about all current IKE SAs at a peer, use the **show crypto isakmp sa** command. The following is sample output from the command:

```
Router# show crypto isakmp sa
dst             src             state          conn-id slot status
11.0.0.1        21.0.0.1        QM_IDLE          68002 ACTIVE
21.0.0.1        11.0.0.1        QM_IDLE          68003 ACTIVE
10.0.0.1        11.0.0.1        QM_IDLE          68001 ACTIVE
```

Displaying HSRP Information

To display information about HSRP configurations, use the **show crypto isakmp ha standby**, **show crypto ipsec ha**, **show ipsec sa**, and **show crypto ipsec sa standby** commands.

Enter the **show crypto isakmp ha standby** command to view your ISAKMP standby or active SAs. The following is sample output from the command:

```
Router# show crypto isakmp ha standby

dst             src             state       I-Cookie           R-Cookie

172.16.31.100   20.3.113.1      QM_IDLE     796885F3  62C3295E  FFAFBACD

EED41AFF

172.16.31.100   20.2.148.1      QM_IDLE     5B78D70F  3D80ED01  FFA03C6D

09FC50BE

172.16.31.100   20.4.124.1      QM_IDLE     B077D0A1  0C8EB3A0  FF5B152C

D233A1E0

172.16.31.100   20.3.88.1       QM_IDLE     55A9F85E  48CC14DE  FF20F9AE

DE37B913

172.16.31.100   20.1.95.1       QM_IDLE     3881DE75  3CF384AE  FF192CAB
```

Enter the **show crypto ipsec ha** command to view your IPsec high availability (HA) manager state. The following is sample output from the command:

```
Router# show crypto ipsec ha

Interface           VIP                SAs   IPSec HA State

FastEthernet0/0     172.16.31.100     1800   Active since 13:00:16 EDT Tue Oct 1 2002
```

Enter the **show crypto ipsec sa** command to view HA status of the IPsec SA (standby or active). The following is sample output from the command:

```
Router# show crypto ipsec sa

interface: FastEthernet0/0

    Crypto map tag: mymap, local addr. 172.168.3.100
    local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
    remote ident (addr/mask/prot/port): (5.6.0.0/255.255.0.0/0/0)
    current_peer: 172.168.3.1
    PERMIT, flags={}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

    local crypto endpt.: 172.168.3.100, remote crypto endpt.: 172.168.3.1
    path mtu 1500, media mtu 1500
    current outbound spi: 132ED6AB

    inbound esp sas:
    spi: 0xD8C8635F(3637011295)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2006, flow_id: 3, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4499/59957)
        IV size: 8 bytes
        replay detection support: Y
        HA Status: STANDBY

    inbound ah sas:
    spi: 0xAAF10A60(2867923552)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2004, flow_id: 3, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4499/59957)
        replay detection support: Y
        HA Status: STANDBY

    inbound pcp sas:

    outbound esp sas:
    spi: 0x132ED6AB(321836715)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2007, flow_id: 4, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4499/59957)
        IV size: 8 bytes
        replay detection support: Y
        HA Status: STANDBY

    outbound ah sas:
    spi: 0x1951D78(26549624)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2005, flow_id: 4, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4499/59957)
        replay detection support: Y
        HA Status: STANDBY

    outbound pcp sas:
```

Enter the **show crypto ipsec sa standby** command to view your standby SAs. The following is sample output from the command:

```
Router# show crypto ipsec sa standby

interface: FastEthernet0/0
    Crypto map tag: mymap, local addr. 172.168.3.100
    local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
    remote ident (addr/mask/prot/port): (5.6.0.0/255.255.0.0/0/0)
    current_peer: 172.168.3.1
    PERMIT, flags={}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

    local crypto endpt.: 172.168.3.100, remote crypto endpt.: 172.168.3.1
    path mtu 1500, media mtu 1500
    current outbound spi: 132ED6AB

    inbound esp sas:
    spi: 0xD8C8635F(3637011295)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2006, flow_id: 3, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4499/59957)
        IV size: 8 bytes
        replay detection support: Y
        HA Status: STANDBY

    inbound ah sas:
    spi: 0xAAF10A60(2867923552)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2004, flow_id: 3, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4499/59957)
        replay detection support: Y
        HA Status: STANDBY

    inbound pcp sas:

    outbound esp sas:
    spi: 0x132ED6AB(321836715)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2007, flow_id: 4, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4499/59957)
        IV size: 8 bytes
        replay detection support: Y
        HA Status: STANDBY

    outbound ah sas:
    spi: 0x1951D78(26549624)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2005, flow_id: 4, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4499/59957)
        replay detection support: Y
        HA Status: STANDBY

    outbound pcp sas:
```

Displaying SSP Information

To display information about an SSP configuration, use the **show ssp client**, **show ssp packet**, **show ssp peers**, and **show ssp redundancy** commands.

Enter the **show ssp client** command to display the domain of interpretation (DOI), name, running version and available version of each client that is registered with SSP. The following is sample output from the command:

```
Router# show ssp client

SSP Client Information

    DOI   Client Name                       Version   Running Ver

      1   IPSec HA Manager                   1.0       1.0

      2   IKE HA Manager                     1.0       1.0
```

Enter the **show ssp packet** command to display the byte count and packet count for the current socket, the creation time of the socket, the server port number, and the port number used for SSP communication. The following is sample output from the command:

```
Router# show ssp packet

SSP packet Information

    Socket creation time: 01:01:06

    Local port: 3249      Server port: 3249

    Packets Sent = 38559, Bytes Sent = 2285020

    Packets Received = 910, Bytes Received = 61472
```

Enter the **show ssp peers** command to display the IP address of the remote peer, the interface used, and the connection state. The following is sample output from the command:

```
Router# show ssp peers

SSP Peer Information

    IP Address      Connection State   Local Interface

    40.0.0.1        Connected          FastEthernet0/1
```

Enter the **show ssp redundancy** command to display the current SSP state, the HSRP group name, interface used, and the elapsed time since last state change. The following is sample output from the command:

```
Router# show ssp redundancy

SSP Redundancy Information

  Device has been ACTIVE for 02:55:34
```

```
    Virtual IP      Redundancy Name             Interface

    172.16.31.100   KNIGHTSOFNI                 FastEthernet0/0
```

Displaying Information About a BFG Configuration

To display information about a BFG configuration, use the **show redundancy linecard-group** and **show crypto ace redundancy** commands. The following is sample output from the commands:

```
Router# show redundancy linecard-group 1

Line Card Redundancy Group:1 Mode:feature-card
Class:load-sharing
Cards:
Slot:3 Subslot:0
Slot:5 Subslot:0

Router# show crypto ace redundancy
--------------------------------------
LC Redundancy Group ID            :1
Pending Configuration Transactions:0
Current State                     :OPERATIONAL
Number of blades in the group     :2
Slots
--------------------------------------
Slot:3 Subslot:0
Slot state:0x36
Booted
Received partner config
Completed Bulk Synchronization
Crypto Engine in Service
Rebooted 22 times
Initialization Timer not running
Slot:5 Subslot:0
Slot state:0x36
Booted
Received partner config
Completed Bulk Synchronization
Crypto Engine in Service
Rebooted 24 times
Initialization Timer not running
```

Displaying Information About RSA Public Keys

To display information the RSA public keys configured for your switch, use the **show crypto key mypubkey rsa** command. The following is sample output from the command:

```
Router# show crypto key mypubkey rsa

% Key pair was generated at: 06:07:50 UTC Jan 13 1996

Key name: myrouter.example.com

 Usage: Encryption Key

 Key Data:

  00302017 4A7D385B 1234EF29 335FC973 2DD50A37 C4F4B0FD 9DADE748 429618D5
  18242BA3 2EDFBDD3 4296142A DDF7D3D8 08407685 2F2190A0 0B43F1BD 9A8A26DB

  07953829 791FCDE9 A98420F0 6A82045B 90288A26 DBC64468 7789F76E EE21
```

To display a list of all the RSA public keys stored on your switch (including the public keys of peers that have sent your switch their certificates during peer authentication for IPsec), or to display details of a particular RSA public key stored on your switch, use the **show crypto key pubkey-chain rsa** command. The following is sample output from the command:

```
Router# show crypto key pubkey-chain rsa

Codes: M - Manually Configured, C - Extracted from certificate

Code  Usage        IP-address     Name

M     Signature    10.0.0.l       myrouter.example.com

M     Encryption   10.0.0.1       myrouter.example.com

C     Signature    172.16.0.1     routerA.example.com

C     Encryption   172.16.0.1     routerA.example.com

C     General      192.168.10.3   routerB.domain1.com
```

Displaying Information About Certificates

To display information about your certificate, the certificate of the CA, and any RA certificates, use the **show crypto pki certificates** command. The following is sample output from the command:

```
Router# show crypto pki certificates

CA Certificate

Status: Available
    Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
    Certificate Usage: Signature

    Issuer:
        CN = new-user
        OU = pki new-user
        O = cisco
        L = santa cruz2
        ST = CA
        C = US
        EA = user@example.com

    Subject:
        CN = new-user
        OU = pki new-user
        O = cisco
        L = santa cruz2
        ST = CA
        C = US
        EA = user@example.com

    CRL Distribution Point:
        http://new-user.example.com/CertEnroll/new-user.crl

    Validity Date:
        start date: 14:19:29 PST Oct 31 2002
        end date: 14:27:27 PST Oct 31 2017

    Associated Trustpoints: MS

Certificate

Status: Available
    Certificate Serial Number: 193E28D20000000009F7
    Certificate Usage: Signature

    Issuer:
        CN = new-user
        OU = pki new-user
        O = cisco
        L = santa cruz2
        ST = CA
        C = US
        EA = user@example.com

    Subject:
        Name: User1.Example.Com

    CRL Distribution Point:
        http://new-user.example.com/CertEnroll/new-user.crl

    Validity Date:
        start date: 12:40:14 PST Feb 26 2003
        end   date: 12:50:14 PST Mar 5 2003
        renew date: 16:00:00 PST Dec 31 1969

    Associated Trustpoints: MS
```

Displaying Information About Trustpoints

To display the trustpoints that are configured in the switch, use the **show crypto pki trustpoints** command. The following is sample output from the command:

```
Router# show crypto pki trustpoints

Trustpoint bo:

    Subject Name:

    CN = ACSWireless Certificate Manager

     O = cisco.com

     C = US

          Serial Number:01

    Certificate configured.

    CEP URL:http://ACSWireless

    CRL query url:ldap://ACSWireless
```

Displaying Information About the NHRP Cache

To display information about the Next Hop Resolution Protocol (NHRP) cache, use the **show ip nhrp** and the **show crypto sockets** commands. The following is sample output from the commands:

```
Router# show ip nhrp

10.10.1.75/32 via 10.10.1.75, Tunnel5 created 00:32:11, expire 00:01:46

  Type: dynamic, Flags: authoritative unique registered

  NBMA address: 172.16.175.75

10.10.1.76/32 via 10.10.1.76, Tunnel5 created 00:26:41, expire 00:01:37

  Type: dynamic, Flags: authoritative unique registered

  NBMA address: 172.16.175.76

10.10.1.77/32 via 10.10.1.77, Tunnel5 created 00:31:26, expire 00:01:33

  Type: dynamic, Flags: authoritative unique registered

  NBMA address: 172.17.63.20

Router# show crypto sockets

Number of Crypto Socket connections 1

   Tu0 Peers (local/remote): 9.1.1.1/11.1.1.1
       Local Ident  (addr/mask/port/prot): (9.1.1.1/255.255.255.255/0/47)
       Remote Ident (addr/mask/port/prot): (11.1.1.1/255.255.255.255/0/47)
       IPSec Profile: "MyIpsecProf"
       Socket State: Open
       Client: "TUNNEL SEC" (Client State: Active)

Crypto Sockets in Listen state:
Client: "TUNNEL SEC" Profile: "MyIpsecProf" Map-name: "Tunnel0-head-0"

Router#
```

Displaying Information About Crypto Sessions

To display status information for active crypto sessions, use the **show crypto session** command. The output will include the following:

- Interface
- IKE peer description, if available
- IKE SAs that are associated with the peer by which the IPsec SAs are created
- IPsec SAs serving the flows of a session

The following is sample output from the command:

```
Router# show crypto session detail

Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, X - IKE Extended Authentication
```

```
Interface: Ethernet1/0
Session status: UP-NO-IKE
Peer: 10.2.80.179/500 fvrf: (none) ivrf: (none)
Desc: My-manual-keyed-peer
Phase1_id: 10.2.80.179
IPSEC FLOW: permit ip host 10.2.80.190 host 10.2.80.179
Active SAs: 4, origin: manual-keyed crypto map
Inbound: #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0

Interface: Ethernet1/2
Session status: DOWN
Peer: 10.1.1.1/500 fvrf: (none) ivrf: (none)
Desc: SJC24-2-VPN-Gateway
Phase1_id: 10.1.1.1
IPSEC FLOW: permit ip host 10.2.2.3 host 10.2.2.2
Active SAs: 0, origin: crypto map
Inbound: #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0
IPSEC FLOW: permit ip 10.2.0.0/255.255.0.0 10.4.0.0/255.255.0.0
Active SAs: 0, origin: crypto map
Inbound: #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0

Interface: Serial2/0.17
Session status: UP-ACTIVE
Peer: 10.1.1.5/500 fvrf: (none) ivrf: (none)
Desc: (none)
Phase1_id: 10.1.1.5
IKE SA: local 10.1.1.5/500 remote 10.1.1.5/500 Active
Capabilities:(none) connid:1 lifetime:00:59:51
IPSEC FLOW: permit ip host 10.1.1.5 host 10.1.2.5
Active SAs: 2, origin: dynamic crypto map
Inbound: #pkts dec'ed 4 drop 0 life (KB/Sec) 20085/171
Outbound: #pkts enc'ed 4 drop 0 life (KB/Sec) 20086/171
```

Displaying Tunnel Interface Information

To display tunnel interface information, use the **show interfaces tunnel** command. The following is sample output from the command:

```
Router# show interfaces tunnel 1

Tunnel4 is up, line protocol is down
Hardware is Routing Tunnel
Internet address is 10.1.1.1/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
Encapsulation TUNNEL, loopback not set
Keepalive set (10 sec)
Tunnel source 9.2.2.1, destination 6.6.6.2
Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
Tunnel TOS 0xF, Tunnel TTL 128
Checksumming of packets disabled, fast tunneling enabled
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Queueing strategy, fifo
Output queue 0/0, 1 drops; input queue 0/75, 0 drops
30 second input rate 0 bits/sec, 0 packets/sec
30 second output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
```

```
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

Table 30-1 describes significant fields shown in the display.

Table 30-1 show interfaces tunnel Field Descriptions

Field	Description
Tunnel is {up | down}	Interface is currently active and inserted into ring (up) or inactive and not inserted (down).
line protocol is {up | down | administratively down}	Shows line protocol up if a valid route is available to the tunnel destination. Shows line protocol down if no route is available, or if the route would be recursive.
Hardware	Specifies the hardware type.
MTU	Maximum transmission unit of the interface.
BW	Bandwidth of the interface in kilobits per second.
DLY	Delay of the interface in microseconds.
rely	Reliability of the interface as a fraction of 255 (255/255 is 100 percent reliability), calculated as an exponential average over 5 minutes.
load	Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes.
Encapsulation	Encapsulation method is always TUNNEL for tunnels.
loopback	Indicates whether loopback is set or not.
Keepalive	Indicates whether keepalives are set or not.
Tunnel source	IP address used as the source address for the tunnel packets.
destination	IP address of the tunnel destination.
Tunnel protocol	Tunnel transport protocol (the protocol the tunnel is using). This is based on the **tunnel mode** command, which defaults to GRE.
key	(Optional) ID key for the tunnel interface.
sequencing	(Optional) Indicates whether the tunnel interface drops datagrams that arrive out of order.
Last input	Number of hours, minutes, and seconds (or never) since the last packet was successfully received by an interface and processed locally on the switch. Useful for knowing when a dead interface failed. This field is not updated by fast-switched traffic.
output	Number of hours, minutes, and seconds (or never) since the last packet was successfully transmitted by an interface.

Table 30-1 show interfaces tunnel Field Descriptions (continued)

Field	Description
output hang	Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is displayed. If that field overflows, asterisks are displayed.
Last clearing	Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared. Three asterisks (***) indicate the elapsed time is too large to be displayed. 0:00:00 indicates the counters were cleared more than 231 ms (and less than 232 ms) ago.
Output queue, drops Input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped because of a full queue.
30 second input rate, 30 second output rate	Average number of bits and packets transmitted per second in the last 30 seconds. The 30-second input and output rates should be used only as an approximation of traffic per second during a given 30-second period. These rates are exponentially weighted averages with a time constant of 30 seconds. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period.
packets input	Total number of error-free packets received by the system.
bytes	Total number of bytes, including data and MAC encapsulation, in the error-free packets received by the system.
no buffer	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernet networks and bursts of noise on serial lines are often responsible for no input buffer events.
broadcasts	Total number of broadcast or multicast packets received by the interface.

Table 30-1 show interfaces tunnel Field Descriptions (continued)

Field	Description
runts	Number of packets that are discarded because they are smaller than the minimum packet size of the medium.
giants	Number of packets that are discarded because they exceed the maximum packet size of the medium.
CRC	Cyclic redundancy checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of a station transmitting bad data.
frame	Number of packets received incorrectly having a CRC error and a noninteger number of octets.
overrun	Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.
ignored	Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased.
abort	Illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment.
packets output	Total number of messages transmitted by the system.
bytes	Total number of bytes, including data and MAC encapsulation, transmitted by the system.
underruns	Number of times that the far-end transmitter has been running faster than the near-end switch's receiver can handle. This may never be reported on some interfaces.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories.

Table 30-1 show interfaces tunnel Field Descriptions (continued)

Field	Description
collisions	Number of messages retransmitted because of an Ethernet collision. This usually is the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). Some collisions are normal. However, if your collision rate climbs to around 4 or 5 percent, you should consider verifying that there is no faulty equipment on the segment and moving some existing stations to a new segment. A packet that collides is counted only once in output packets.
interface resets	Number of times an interface has been reset. The interface may be reset by the administrator or automatically when an internal error occurs.
restarts	Number of times that the controller was restarted because of errors.

Displaying Information About IP Multicast Over a GRE Tunnel

To display information about an IP multicast over a GRE tunnel configuration, enter the **show crypto vlan** and **show ip mroute** commands.

Enter the **show crypto vlan** command to check that the tunnel has been taken over by the IPsec VPN SPA. The following is sample output from the command:

```
Router# show crypto vlan

Interface VLAN 100 on IPSec Service Module port Gi7/0/1 connected to Po1 with crypto map
set map_t3
Tunnel15 is accelerated via IPSec SM in subslot 7/0
```

Enter the **show ip mroute** command and look for the H flag to check that the IP multicast traffic is hardware-switched. The following is sample output from the command:

```
Router# show ip mroute 230.1.1.5

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 230.1.1.5), 01:23:45/00:03:16, RP 15.15.1.1, flags: SJC
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Tunnel15, Forward/Sparse-Dense, 00:25:47/00:03:16
```

```
(120.1.0.3, 230.1.1.5), 01:23:46/00:03:25, flags: T
Incoming interface: GigabitEthernet8/1, RPF nbr 0.0.0.0, RPF-MFD
Outgoing interface list:
Tunnel15, Forward/Sparse-Dense, 00:25:47/00:03:16, H
```

Troubleshooting Specific Problems on the IPsec VPN SPA

This section provides additional information about troubleshooting specific problems related to the IPsec VPN SPA. It includes the following subsections:

- Clearing IPsec Security Associations, page 30-24
- Troubleshooting Trunk Port Configurations, page 30-24
- Troubleshooting IPsec Stateful Failover (VPN High Availability), page 30-25
- Troubleshooting a Blade Failure Group, page 30-27
- Troubleshooting IKE Policy and Transform Sets, page 30-27

Clearing IPsec Security Associations

You can clear (and reinitialize) IPsec security associations by using the **clear crypto sa** command.

Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, refer to the **clear crypto sa** command in the *Cisco IOS Security Command Reference*, *Release 12.2*.

If you want to also remove the IKE (phase 1) SAs, follow the **clear crypto sa** command with the **clear crypto isa** command. Alternatively, you can use the **clear crypto session** command to achieve the same result as the **clear crypto sa** and the **clear crypto isa** commands. The **clear crypto session** command supports many of the same parameters as the **clear crypto sa** command.

Troubleshooting Trunk Port Configurations

Caution When you configure an Ethernet port as a trunk port, all the VLANs are allowed on the trunk port by default. This default configuration does not work well with the IPsec VPN SPA and causes network loops. To avoid this problem, you must explicitly specify only the desirable VLANs.

For more information on trunk configuration guidelines, review the "Configuring a Trunk Port" section on page 21-14.

To verify which ports are assigned to the VLAN, enter the **show vlan id** *number* command, using the interface VLAN identifier. Following is an example of a trunk port configuration and the output of the **show vlan id** command:

```
Router# show run interface gi 1/3
Building configuration...

Current configuration : 175 bytes
!
interface GigabitEthernet1/3
```

```
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,502-504,1002-1005
 switchport mode trunk
 no ip address
end


Router# show crypto vlan

Interface VLAN 2 on IPSec Service Module port Gi7/0/1 connected to VLAN 502 with crypto
map set testtag_1
Interface VLAN 3 on IPSec Service Module port Gi7/0/1 connected to VLAN 503 with crypto
map set testtag_2
Interface VLAN 4 on IPSec Service Module port Gi7/0/1 connected to VLAN 504 with crypto
map set testtag_3

Router# show vlan id 2

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
2    VLAN0002                         active    Gi7/0/1

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
2    enet  100002     1500  -      -      -        -    -        0      0

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

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

Router# show vlan id 502

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
502  VLAN0502                         active    Gi1/3, Gi7/0/2

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
502  enet  100502     1500  -      -      -        -    -        0      0

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

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

Router#
```

Troubleshooting IPsec Stateful Failover (VPN High Availability)

If you find that either the active or standby IPsec stateful failover (VPN high availability) processes do not function as expected, you can perform the following checks:

- Use the **show ssp** command to verify the SSP process is running.
- Make sure that both switches share identical IPsec configurations. This is critical. If switches are configured differently, IPsec stateful failover (VPN high availability) will not work.

Note Support for IPsec stateful failover is removed in Cisco IOS Release 12.2SXH. The feature is supported in Cisco IOS Release 12.2SXF.

- Verify that an IPsec connection can be formed with existing maps, transforms, and access lists.
- Configure HSRP on the inside and outside interfaces and make the HSRP groups track one another. Verify this works properly by performing a **shut** command on either of the interfaces, then observe that the HSRP standby switch takes active control from the active switch.
- Verify that SSP peers can see each other by performing a **show ssp peer** command on both the active and standby switches.
- Bind the IKE and IPsec to SSP and send traffic over the tunnels. You can view high availability (HA) messages on the standby switch as both the active and standby switches synchronize.
- HSRP settings may require adjustments depending on the interface employed, such as Fast Ethernet or Gigabit Ethernet.

Checking HSRP Settings

To check HSRP settings, perform this task:

	Command	Purpose
Step 1	`Router# show standby brief`	Ensures that the interfaces are synchronized.
Step 2	`Router# no standby delay timer`	Leaves the delay timers at their default settings
Step 3	`Router# show standby brief`	When the other switch comes online, enter the **show standby brief** command once again. If the output shows an interface on standby, you must set the standby switch's delay timer.

Clearing Dormant SAs on Standby Switches

To clear associated SA entries, perform this task:

	Command	Purpose
Step 1	`Router# clear crypto isakmp ha [standby][resync]`	Clears all dormant (standby) entries from the device. If the **resync** keyword is used, all standby IKE SAs will be removed, and a resynchronization of state will occur.
Step 2	`Router# clear crypto sa ha standby [peer ip address | resync]`	Clears all standby SAs for the device if **peer** is specified.

Enabling Debugging for HA

To enable debugging for HA, perform this task:

	Command	Purpose
Step 1	Router# **debug crypto isakmp ha** [**detail** | **fsm** | **update**]	Enables basic debug messages related to the IKE HA Manager.
Step 2	Router# **debug crypto ipsec ha** [**detail** | **fsm** | **update**]	Enables IPsec HA debugging
Step 3	Router# **debug ssp** [**fsm** | **socket** | **packet** | **peers** | **redundancy** | **config**]	Enables SSP debugging.

Troubleshooting a Blade Failure Group

To enable IPsec VPN SPA debugging for a blade failure group, enter the **debug crypto ace b2b** command:

```
Router# debug crypto ace b2b

ACE B2B Failover debugging is on
```

Troubleshooting IKE Policy and Transform Sets

Any IPsec transforms or IKE encryption methods that the current hardware does not support should be disabled; they are ignored whenever an attempt to negotiate with the peer is made.

If a user enters an IPsec transform or an IKE encryption method that the hardware does not support, a warning message will be generated. These warning messages are also generated at boot time. When an encrypted card is inserted, the current configuration is scanned. If any IPsec transforms or IKE encryption methods are found that are not supported by the hardware, a warning message will be generated.

Using Crypto Conditional Debug

The crypto conditional debug feature provides three command-line interface (CLI) commands that allow you to debug an IP Security (IPsec) tunnel on the basis of predefined crypto conditions such as the peer IP address, connection-ID of a crypto engine, and security parameter index (SPI). By limiting debug messages to specific IPsec operations and reducing the amount of debug output, you can better troubleshoot a switch with a large number of tunnels.

The crypto conditional debug commands (**debug crypto condition**, **debug crypto condition unmatched**, and **show crypto debug-condition**) allow you to specify conditions (filter values) in which to generate and display debug messages related only to the specified conditions.

Table 30-2 lists the supported condition types.

Table 30-2 Supported Condition Types for Crypto Conditional Debug Commands

Condition Type (Keyword)	Description
connid	An integer between 1 and 32766. Relevant debug messages will be shown if the current IPsec operation uses this value as the connection-ID to interface with the crypto engine.
flowid	An integer between 1 and 32766. Relevant debug messages will be shown if the current IPsec operation uses this value as the flow-ID to interface with the crypto engine.
fvrf	The name string of a virtual private network (VPN) routing and forwarding (VRF) instance. Relevant debug messages will be shown if the current IPsec operation uses this VRF instance as its front-door VRF (FVRF).
ivrf	The name string of a VRF instance. Relevant debug messages will be shown if the current IPsec operation uses this VRF instance as its inside VRF (IVRF).
peer group	A Unity group name string. Relevant debug messages will be shown if the peer is using this group name as its identity.
peer hostname	A fully qualified domain name (FQDN) string. Relevant debug messages will be shown if the peer is using this string as its identity.
peer ipv4	A single IP address. Relevant debug messages will be shown if the current IPsec operation is related to the IP address of this peer.
peer subnet	A subnet and a subnet mask that specify a range of peer IP addresses. Relevant debug messages will be shown if the IP address of the current IPsec peer falls into the specified subnet range.
peer username	A username string. Relevant debug messages will be shown if the peer is using this username as its identity.
spi	A 32-bit unsigned integer. Relevant debug messages will be shown if the current IPsec operation uses this value as the SPI.

Note If **connid**, **flowid**, or **spi** is used as a debug condition, the debug messages for a related IPsec flow are generated. An IPsec flow has two connection-IDs, flow-IDs, and SPI values—one inbound and one outbound. Either one of the two connection-IDs, flow-IDs, and SPI values can be used as the debug condition that triggers debug messages for the IPsec flow.

Crypto Conditional Debug Configuration Guidelines and Restrictions

When configuring crypto conditional debug, follow these guidelines and restrictions:

- This feature does not support debug message filtering for hardware crypto engines.
- Although conditional debugging is useful for troubleshooting peer-specific or functionality-related Internet Key Exchange (IKE) and IPsec problems, conditional debugging may not be able to define and check large numbers of debug conditions.
- Because extra space is needed to store the debug condition values, additional processing overhead is added to the CPU and memory usage is increased. Thus, enabling crypto conditional debugging on a switch with heavy traffic should be used with caution.
- Your switch will perform conditional debugging only after at least one of the global crypto debug commands (**debug crypto isakmp**, **debug crypto ipsec**, or **debug crypto engine**) has been enabled. This requirement helps to ensure that the performance of the switch will not be impacted when conditional debugging is not being used.

Enabling Crypto Conditional Debug Filtering

To enable crypto conditional debug filtering, perform the following tasks:

	Command	Purpose
Step 1	`Router#` **`enable`**	Enables privileged EXEC mode.
Step 2	`Router#` **`debug crypto condition`** `[`**`connid`** *`integer`* **`engine-id`** *`integer`*`] [`**`flowid`** *`integer`* **`engine-id`** *`integer`*`] [`**`fvrf`** *`string`*`] [`**`ivrf`** *`string`*`] [`**`peer`** `[`**`group`** *`string`*`] [`**`hostname`** *`string`*`] [`**`ipv4`** `ipaddress] [`**`subnet`** *`subnet mask`*`] [`**`username`** *`string`*`]] [`**`spi`** *`integer`*`] [`**`reset`**`]`	Defines conditional debug filters. See Table 30-2 for descriptions of values.
Step 3	`Router#` **`show crypto debug-condition`** `{[`**`peer`**`] [`**`connid`**`] [`**`spi`**`] [`**`fvrf`**`] [`**`ivrf`**`] [`**`unmatched`**`]}`	Displays crypto debug conditions that have already been enabled in the switch.
Step 4	`Router#` **`debug crypto isakmp`**	Enables global IKE debugging.
Step 5	`Router#` **`debug crypto ipsec`**	Enables global IPsec debugging.
Step 6	`Router#` **`debug crypto engine`**	Enables global crypto engine debugging.
Step 7	`Router#` **`debug crypto condition unmatched`** `[`**`isakmp`** `|` **`ipsec`** `|` **`engine`**`]`	(Optional) Displays debug conditional crypto messages when no context information is available to check against debug conditions. If none of the optional keywords are specified, all crypto-related information will be shown.

Disabling Crypto Conditional Debugging

Before you disable crypto conditional debugging, you must first disable any crypto global debug CLIs that you have issued. You can then disable crypto conditional debugging. To disable crypto conditional debugging, enter the following command:

```
Router# debug crypto condition reset
```

Enabling Crypto Error Debug Messages

Enabling the **debug crypto error** command displays only error-related debug messages, which allows you to easily determine why a crypto operation, such as an IKE negotiation, has failed within your system. To enable crypto error debug messages, enter the following command from privileged EXEC mode:

```
Router# debug crypto {isakmp | ipsec | engine} error
```

Note When enabling this command, ensure that global crypto debug commands are not enabled; otherwise, the global commands will override any possible error-related debug messages.

For complete configuration information for crypto conditional debug support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t2/feature/guide/gt_dbcry.html

Preparing for Online Insertion and Removal of a SPA

The Catalyst 6500 Series switch supports online insertion and removal (OIR) of the SSC, in addition to each of the SPAs. You can remove an SSC with its SPAs still intact, or you can remove a SPA independently from the SSC, leaving the SSC installed in the switch.

An SSC can remain installed in the switch with one SPA remaining active while you remove another SPA from one of the SSC subslots. If you are not planning to immediately replace a SPA into the SSC, then be sure to install a blank filler plate in the subslot. The SSC should always be fully installed with either functional SPAs or blank filler plates.

For more information about activating and deactivating SPAs in preparation for OIR, see the “Preparing for Online Insertion and Removal of SIPs and SPAs” topic in the “Troubleshooting the SIPs and SSC” chapter in this guide.

PART 8

Field-Programmable Devices

CHAPTER 31

Upgrading Field-Programmable Devices

In general terms, field-programmable devices (FPDs) are hardware devices implemented on switch cards that support separate upgrades. The term “FPD” has been introduced to collectively and generically describe any type of programmable hardware device on SIPs and SPAs. FPDs were introduced on the Catalyst 6500 Series switch to support SPAs and SIPs.

This chapter describes the information that you need to know to verify image versions and to perform SIP and SPA FPD upgrades.

This chapter includes the following sections:

Release History

Table 31-1 provides the release and modification history for all FPD-related features on the Catalyst 6500 Series switch.

Table 31-1 FPD Release History

Release	Modification
Cisco IOS Release 12.2(18)SXE	SIPs and SPAs were released on the Cisco 7600 series router and Catalyst 6500 series switch for the first time. FPD images were introduced to support these SPAs. The Fast Software Upgrade (FSU) procedure supported by Route Processor Redundancy (RPR) for supervisor engines was added to the documentation.

FPD Quick Upgrade

This section provides information if you simply want to upgrade FPDs for SIPs and SPAs as quickly as possible. These instructions are not always feasible for operating network environments and are not the only methods available for upgrading FPDs. If these methods of upgrade are not suitable for your situation, see the various other sections of this document for other methods of upgrading FPDs.

This section addresses the following topics:

- FPD Quick Upgrade Before Upgrading Your Cisco IOS Release (Recommended), page 31-2
- FPD Quick Upgrade After Upgrading Your Cisco IOS Release, page 31-2

FPD Quick Upgrade Before Upgrading Your Cisco IOS Release (Recommended)

Step 1 When getting your Cisco IOS image, download the FPD image package for the Cisco IOS release that you are upgrading to any flash disk on your switch before booting the new version of Cisco IOS. The FPD image package can be retrieved from the same site where you went to get your Cisco IOS image. Do not change the name of the FPD image package.

Step 2 Boot using the new version of Cisco IOS. When the new Cisco IOS boots, it by default searches for the FPD image package in the switch flash file systems and the FPD images will be updated automatically as part of the IOS boot process.

FPD Quick Upgrade After Upgrading Your Cisco IOS Release

Step 1 An FPD upgrade is not always necessary after Cisco IOS is reloaded. If you have already reloaded your Cisco IOS Software, enter the **show hw-module all fpd** command to see if all system FPDs are compatible. If the FPDs are compatible, no further action is necessary. If at least one FPD needs an upgrade, proceed to Step 2.

Step 2 Go to the cisco.com site where you downloaded your specific Cisco IOS software and locate the FPD image package, if you have not already.

Step 3 Download this FPD image package to a flash disk on your switch. Do not change the name of the FPD image package.

Do not change any FPD-related settings on your system (if **upgrade fpd auto** or **upgrade fpd path** has been changed, change the settings back to the default settings using the **no** form of the command). Reboot your Cisco IOS release software. When the new Cisco IOS boots, it by default searches for the FPD image package in the flash file systems and the FPD images will be updated automatically as part of the IOS boot process.

Overview of FPD Images and Packages

An FPD image package is used to upgrade FPD images. Whenever a Cisco IOS image is released that supports carrier cards and SPAs, a companion FPD image package is also released for that Cisco IOS software release. The FPD image package is available from Cisco.com and is accessible from the Cisco Software Center page where you also go to download your Cisco IOS software image.

If you are running SIPs and SPAs on your switch and are upgrading your Cisco IOS image, you should download the FPD image package file before booting the switch using the new Cisco IOS release. If the SIP or SPA requires an FPD upgrade and the Cisco IOS image is unable to locate an FPD image package, the system messages will indicate that the FPD image is incompatible and you will need to go to the Cisco Software Center on Cisco.com to download the FPD image package for your Cisco IOS software release. An FPD incompatibility on a SPA disables all interfaces on that SPA until the incompatibility is addressed; an FPD incompatibility on a SIP disables all interfaces for all SPAs in the SIP until the incompatibility is addressed.

Note The FPD automatic upgrade feature only searches for the FPD image package file that is the same version number as the Cisco IOS release being used by the system. For example, if the Cisco IOS release being used is Cisco IOS Release 12.2(18)SXE, then the system will search for the FPD image package file that supports the specific Cisco IOS release (c7600-fpd-pkg.122-18.SXE.pkg). Therefore, ensure the FPD image package file on your system is compatible with your Cisco IOS release and do not change the name of the FPD image package file.

Upgrading FPD Images

This section documents some of the common scenarios where FPD image updates are necessary. It discusses the following scenarios:

- Migrating to a Newer Cisco IOS Release, page 31-3
- Upgrading FPD Images in a Production System, page 31-5

Migrating to a Newer Cisco IOS Release

This section discusses the following topics:

- Upgrading FPD Images Before Upgrading the Cisco IOS Release (Recommended), page 31-3
- Upgrading FPD Images in a Production System, page 31-5
- Upgrading FPD Images Using Fast Software Upgrade, page 31-6

Upgrading FPD Images Before Upgrading the Cisco IOS Release (Recommended)

If you are still running your old Cisco IOS release but are preparing to load a newer version of Cisco IOS, you can upgrade FPD for the new Cisco IOS release using the following method:

- Placing FPD Image Package on Flash Disk Before Upgrading Cisco IOS (Recommended), page 31-4

Placing FPD Image Package on Flash Disk Before Upgrading Cisco IOS (Recommended)

Placing the FPD image package for the Cisco IOS release that you are upgrading to before upgrading Cisco IOS is the recommended method for upgrading FPD because it is simple in addition to being fast. To perform this type of FPD upgrade, follow these steps:

Step 1 While still running the Cisco IOS release that will be upgraded, place the FPD image package for the new version of Cisco IOS onto one of your switch's flash file systems. For instance, if you are running Cisco IOS Release 12.2(18)SXE and are upgrading to Cisco IOS Release 12.2(19)SXE, place the FPD image package for Cisco IOS Release 12.2(19)SXE onto a flash file system while still running Cisco IOS Release 12.2(18)SXE. You can locate the FPD image package for a specific IOS release on cisco.com from the same area where you download that Cisco IOS software image. Your switch and SPAs should continue to operate normally since this action will have no impact on the current FPDs.

Caution Do not change the filename of the FPD image package file. The Cisco IOS software searches for the FPD image package file by filename, so the FPD image package file cannot be found if it has been renamed.

Step 2 Reboot your switch using the new upgraded Cisco IOS image. As part of the bootup process, the switch will search for the FPD image package. Since the default settings for the FPD image package search are to check for the FPD image package for the specific Cisco IOS release in a flash file system, the FPD image package will be located during the bootup procedure and all FPDs that required upgrades will be upgraded.

Step 3 When the switch has booted, verify the upgrade was successful by entering the **show hw-module all fpd** command.

Upgrading FPD Images After Upgrading the Cisco IOS Release

The following steps explain how to upgrade FPD images if you have already upgraded your Cisco IOS release but still need to upgrade your FPD images.

To perform an FPD upgrade after the new Cisco release has been booted, follow these steps:

Step 1 If you are unsure if your FPD images for your SIPs and SPAs are compatible, enter the **show hw-module all fpd** command to verify compatibility of all SIPs and SPAs. If all of your SIPs and SPAs are compatible, there is no reason to perform this upgrade.

Step 2 If an FPD upgrade is necessary, place the FPD image package for the new version of Cisco IOS onto the switch's flash disk or on an accessible FTP or TFTP server. You can locate the FPD image package on cisco.com from the same area where you downloaded your Cisco IOS software image.

Step 3 Enter the **upgrade hw-module** [**slot** *slot-number* | **subslot** *slot-number*/*subslot-number*] *file-url* [**force**] command. The *file-url* command should direct users to the location of the FPD image package. For instance, if you had placed the FPD image package for Release 12.2(18)SXE on the TFTP server abrick/muck/myfolder, you would enter **upgrade hw-module** [**slot** *slot-number* | **subslot** *slot-number*/*subslot-number*] **tftp://abrick/muck/myfolder/c7600-fpd-pkg.122-18.SXE.pkg** to complete this step.

If multiple SIPs or SPAs require upgrades, the different pieces of hardware will have to be updated individually.

Note the **force** option is used in this command. This option will force an FPD upgrade even if no FPD mismatch is detected. In instances where the **upgrade hw-module** command is entered, this option is almost never necessary and should only be entered if requested by a technical support representative.

Step 4 Verify the upgrade was successful by entering the **show hw-module all fpd** command.

Upgrading FPD Images in a Production System

Adding a SIP or SPA to a production system presents the possibility that the SIP or SPA may contain versions of FPD images that are incompatible with the Cisco IOS release currently running the switch. In addition, the FPD upgrade operation can be a very CPU-intensive operation and therefore the upgrade operation may take more time when it is performed on a production system. The performance impact will vary depending on various factors, including network traffic load, the type of processing engine used, type of SPA, and the type of service configured.

For these reasons, we recommend that one of the following alternatives be used to perform the FPD upgrade on a production system if possible:

- Using a Non-Production System to Upgrade the SIP or SPA FPD Image, page 31-5
- Verifying System Compatibility First, page 31-6

Using a Non-Production System to Upgrade the SIP or SPA FPD Image

Before beginning the upgrade, ensure the following:

- The spare system is running the same version of the Cisco IOS software release that the target production system is running.
- The automatic upgrade feature is enabled on the spare system (the automatic upgrade feature is enabled by default. It can also be enabled using the **upgrade fpd auto** command).

To perform an upgrade on a spare system, follow these steps:

Step 1 Download the FPD image package file to the switch's flash file system or TFTP or FTP server accessible by the spare system. In most cases, it is preferable to place the file in a flash file system since the switch, by default, searches for the FPD image package in the flash file systems. If the flash file systems are full, use the **upgrade fpd path** command to direct the switch to search for the FPD image package in the proper location.

Step 2 Insert the SIP or SPA into the spare system.

If an upgrade is required, the system will perform the necessary FPD image updates so that when this SIP or SPA is inserted to the target production system it will not trigger an FPD upgrade operation there.

Step 3 Verify the upgrade was successful by entering the **show hw-module all fpd** command.

Step 4 Remove the SIP or SPA from the spare system after the upgrade.

Step 5 Insert the SIP or SPA into the target production system.

Verifying System Compatibility First

If a spare system is not available to perform an upgrade, you can check for system compatibility by disabling the automatic upgrade feature before inserting the SIP or SPA (the automatic upgrade feature is enabled by default. It can be disabled using the **no upgrade fpd auto** command).

- If the FPD images on the SIP or SPA are compatible with the system, you will only need to reenable the automatic upgrade feature (the automatic upgrade feature can be reenabled using the **upgrade fpd auto** command).
- If the FPD images on the SIP or SPA are not compatible with the system, the SIP or SPA is disabled but will not impact system performance by attempting to perform an automatic upgrade.

To check the FPD images on the SIP or SPA for system compatibility, follow these steps:

Step 1 Disable the automatic upgrade feature using the **no upgrade fpd auto** global configuration command.

Step 2 Insert the SIP or SPA into the system.

If the FPD images are compatible, the SIP or SPA will operate successfully after bootup.

If the FPD images are not compatible, the SIP or SPA is disabled. At this point we recommend that you wait for a scheduled maintenance when the system is offline to manually perform the FPD upgrade using one of the procedures outlined in the "Upgrading FPD Images" section on page 31-3.

Step 3 Reenable the automatic upgrade feature using the **upgrade fpd auto** global configuration command.

Upgrading FPD Images Using Fast Software Upgrade

The fast software upgrade (FSU) procedure supported by Route Processor Redundancy (RPR) allows you to upgrade the Cisco IOS image on supervisor engines without reloading the system.

When using FSU to upgrade the Cisco IOS image, remember that Cisco IOS software is configured, by default, to automatically load the new FPD images from a flash file system on the router. Therefore, if the FPD image package for the new Cisco IOS has not been downloaded to the router flash file system, the FPD image that needs to be upgraded will not get upgraded if the new supervisor engine with the upgraded Cisco IOS becomes the primary supervisor engine. To ensure FPD is upgraded at the time of the FSU, place the FPD image package for the new version of Cisco IOS onto the flash file system before upgrading the Cisco IOS and follow the instructions in the "Upgrading FPD Images Before Upgrading the Cisco IOS Release (Recommended)" section on page 31-3.

If a SIP or SPA is disabled after FSU is used to upgrade Cisco IOS and the supervisor engine with the upgraded Cisco IOS has become the primary supervisor engine, follow the instructions in the "Upgrading FPD Images After Upgrading the Cisco IOS Release" section on page 31-4 to verify and, if necessary, upgrade FPD.

Optional FPD Procedures

This section provides information for optional FPD-related functions. None of the topics discussed in this section are necessary for completing FPD upgrades, but may be useful in some FPD-related scenarios. It covers the following topics:

- Manually Upgrading SIP and SPA FPD Images, page 31-7

- Upgrading FPD from an FTP or TFTP Server, page 31-7
- Modifying the Default Path for the FPD Image Package File Location, page 31-9
- Upgrading Multiple FPD Images, page 31-10
- Displaying Current and Minimum Required FPD Image Versions, page 31-10
- Displaying Information About the Default FPD Image Package, page 31-12
- Verifying the FPD Image Upgrade Progress, page 31-12

Manually Upgrading SIP and SPA FPD Images

To manually upgrade the current FPD version on a SIP or SPA, use the following command:

```
Router# upgrade hw-module [slot slot-number | subslot slot-number/subslot-number] file
file-url [force]
```

In this example, *slot-number* is the slot where the SIP is installed, *subslot-number* is the subslot number where the SPA is located, *file-url* is the location and name of the FPD image package file, and **force** is an option that forces the SPA to perform an FPD upgrade even if FPD is compatible (the **force** option is almost never necessary and should only be entered if requested by a technical support representative). Note that **slot** *slot-number* is entered to specify a SIP FPD upgrade, while **subslot** *slot-number/subslot-number* is used to specify a SPA FPD upgrade. The SIP or SPA will automatically be reloaded to complete the FPD upgrade.

Caution An image upgrade can require a long period of time to complete depending on the SIP or SPA.

Upgrading FPD from an FTP or TFTP Server

The generally recommended method to perform an FPD image upgrade is to download the FPD image package to a flash file system and use the FPD automatic upgrade. By default, the system searches the flash file system for the FPD image package file when an FPD incompatibility is detected.

This default behavior of loading an FPD image from flash can be changed using the **upgrade fpd path** global configuration command, which sets the path to search for the FPD image package file to a location other than the switch's flash file systems.

For large deployments where all the systems are being upgraded to a specific Cisco IOS software release, we recommend that the FPD image package file be placed on an FTP or TFTP server that is accessible to all the affected systems, and then use the **upgrade fpd path** global configuration command to configure the switches to look for the FPD image package file from the FTP or TFTP server prior to the reloading of the system with the new Cisco IOS release.

Note This approach can also be used if there is not enough disk space on the system flash card to hold the FPD image package file.

To download an FPD image package file to an FTP or TFTP server, use the following procedure:

Step 1 Copy the FPD image package file to the FTP or TFTP server.

Step 2 Access the switch from a connection that does not use the SPA interface for access, if possible. We recommend not using the SPA interface as your connection to the switch because an FPD incompatibility disables all interfaces on the SPA, making a manual FPD upgrade impossible through a SPA interface.

If access through one of the SPA ports is the only access to the switch you have, do not use the TFTP or FTP upgrade method. Instead, copy the FPD image package to your switch's default flash card before upgrading your Cisco IOS release. This will allow the switch to find the FPD image package during the first IOS bootup and upgrade FPD automatically.

Step 3 From global configuration mode, use the **upgrade fpd path** command to instruct the switch to locate the FPD image package file from the FTP or TFTP server location.

For example, enter one of the following global configuration commands from the target system's console:

```
Router(config)# upgrade fpd path tftp://my_tftpserver/fpd_pkg_dir/
```

or

```
Router(config)# upgrade fpd path ftp://login:password@my_ftpserver/fpd_pkg_dir/
```

Note The final "/" at the end of each of the above examples is required. If the path is specified without the trailing "/" character, the command will not work properly.

In these examples, *my_tftpserver* or *my_ftpserver* is the path to server name, *fpd_pkg_dir* is the directory on the TFTP server where the FPD image package is located, and *login:password* is your FTP login name and password.

Step 4 Make sure that the FPD automatic upgrade feature is enabled by examining the output of the **show running-config** command. (Look for the *upgrade fpd auto* configuration line in the output. If there are no upgrade commands in the output, then **upgrade fpd auto** is enabled because it is the default setting.) If automatic upgrades are disabled, use the **upgrade fpd auto** global configuration command to enable automatic FPD upgrades.

Step 5 Enter the **show upgrade fpd file** command to ensure your switch is connecting properly to the default FPD image package. If you are able to generate output related to the FPD image package using this command, the upgrade should work properly.

In the following example, the switch is able to generate FPD image package information for the FPD image package on the TFTP server.

```
Router# show upgrade fpd file
tftp://mytftpserver/myname/myfpdpkg/c7600-fpd-pkg.122-18.SXE.pkg
Loading myname/myfpdpkg/c7600-fpd-pkg.122-18.SXE.pkg from 124.0.0.0 (via FastEthernet0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK]
Cisco Field Programmable Device Image Package for IOS
C7600 Family FPD Image Package (c7600-fpd-pkg.122-18.SXE.pkg), Version 12.2(SXE)
Copyright (c) 2004-2005 by cisco Systems, Inc.
Built Fri 25-Mar-2005 09:12 by integ


=============================== ================================================
                                        Bundled FPD Image Version Matrix
                                ================================================
                                                                      Min. Req.
Supported Card Types            ID  Image Name               Version  H/W Ver.
=============================== == ======================== ========= =========
2-port T3/E3 Serial SPA          1 T3E3 SPA ROMMON             2.12      0.0
                                 2 T3E3 SPA I/O FPGA           0.24      0.0
                                 3 T3E3 SPA E3 FPGA            0.6       0.0
                                 4 T3E3 SPA T3 FPGA            0.14      0.0
------------------------------- -- ------------------------ --------- ---------
```

```
4-port T3/E3 Serial SPA          1 T3E3 SPA ROMMON              2.12      0.0
                                 2 T3E3 SPA I/O FPGA            0.24      0.0
                                 3 T3E3 SPA E3 FPGA             0.6       0.0
                                 4 T3E3 SPA T3 FPGA             0.14      0.0
------------------------------- -- ------------------------ --------- ---------
...<additional output removed for readability>
```

Step 6 Save the configuration and reload the system with the new Cisco IOS release.

During the system startup after the reload, the necessary FPD image version check for all the SIPs and SPAs will be performed and any upgrade operation will occur automatically if an upgrade is required. In each upgrade operation, the system extracts the necessary FPD images to the SIP or SPA from the FPD image package file located on the FTP or TFTP server.

Modifying the Default Path for the FPD Image Package File Location

By default, the Cisco IOS software looks for the FPD image package file on a flash file system when performing an automatic FPD image upgrade.

Note Be sure there is enough space on one of your flash file systems to accommodate the FPD image package file.

Alternatively, you can store an FPD image package file elsewhere. However, because the system looks on the flash file systems by default, you need to change the FPD image package file location so that the system is directed to search an alternate location (such an FTP or TFTP server) that is accessible by the Cisco IOS software. Enter the **upgrade fpd path** *fpd-pkg-dir-url* global configuration command, where *fpd-pkg-dir-url* is the alternate location, to instruct the switch to search for the FPD image package elsewhere.

When specifying the *fpd-pkg-dir-url*, be aware of the following:

- The *fpd-pkg-dir-url* is the path to the FPD image package, but the FPD image package should not be specified as part of the *fpd-pkg-dir-url*. For instance, if the c7600-fpd-pkg.122-18.SXE.pkg file can be found on the TFTP server using the path mytftpserver/myname/myfpdpkg/c7600-fpd-pkg.122-18.SXE.pkg and you wanted the switch to utilize this FPD image package for FPD upgrades, the **upgrade fpd path tftp://mytftpserver/myname/myfpdpkg/** command should be entered so the switch knows where to find the file. The actual filename should not be specified.
- The final "/" character in the *fpd-pkg-dir-url* is required. In the preceding example, note that the *fpd-pkg-dir-url* is **tftp://mytftpserver/myname/myfpdpkg/.** Entering **tftp://mytftpserver/myname/myfpdpkg** (the final "/" character is missing) as the *fpd-pkg-dir-url* in that scenario would not work.

If the **upgrade fpd path** global configuration command has not been entered to direct the switch to locate an FPD image package file in an alternate location, the system searches the flash file systems on the Catalyst 6500 Series switch for the FPD image package file.

Failure to locate an FPD image package file when an upgrade is required will disable the SIP or SPA. Because SIPs and SPAs will not come online until FPD is compatible, the SIP or SPA will also be disabled if it requires an FPD upgrade and the automatic upgrade feature is disabled.

Upgrading Multiple FPD Images

A single piece of hardware can contain multiple FPD images. The Catalyst 6500 Series switch can upgrade up to four FPD images simultaneously. However, only one FPD upgrade per switch slot can occur at a time, so all FPD images on all SIPs and SPAs in a single slot will have to wait for another FPD upgrade to finish.

Users should note that some FPD images require the SIP or SPA to reload to complete. The FPD upgrade process will perform this step automatically, so users do not have to intervene. However, the other FPDs in the hardware of the specified slot will have to wait for this reload to complete before their upgrade process begins.

During an automatic upgrade, the Catalyst 6500 Series switch will upgrade as many FPDs as possible at a time. No user intervention is possible or necessary. The upgrade process will not stop until all FPD images have been updated.

During manual upgrades, it is important to note that users can only specify upgrades for a single piece of hardware each time the **upgrade hw-module** [**slot** *slot-number* **|** **subslot** *slot-number***/***subslot-number*] is entered. The up to four simultaneous upgrades applies to the manual upgrades as well; if you individually specify multiple manual FPD upgrades, only four FPDs can be upgraded simultaneously and that can only occur when the hardware is in different switch slots. The FPD upgrade process will stop when all FPDs for the specified hardware have been upgraded.

Displaying Current and Minimum Required FPD Image Versions

To display the current version of FPD images on the SIPs and SPAs installed on your switch, use the **show hw-module** [*slot-number***/***subslot-number* **|** **all**] **fpd** command, where *slot-number* is the slot number where the SIP is installed, and *subslot-number* is the number of the SIP subslot where a target SPA is located. Entering the **all** keyword shows information for hardware in all switch slots.

The following examples show the output when using this **show** command.

The output display in this example shows that FPD versions on the SIPs and SPAs in the system meet the minimum requirements:

```
Router# show hw-module all fpd

==== ====================== ====== =============================================
                             H/W   Field Programmable   Current   Min. Required
Slot Card Type               Ver.  Device:"ID-Name"    Version      Version
==== ====================== ====== ================== =========== ==============
   1 7600-SIP-200           0.550 1-I/O FPGA             1.1         1.1
                                   2-EOS FPGA             1.211       1.211
                                   3-PEGASUS TX FPGA      1.129       1.129
                                   4-PEGASUS RX FPGA      1.3         1.3
                                   5-ROMMON               1.2         1.2
---- ---------------------- ------ ------------------ ----------- --------------
 1/1 SPA-2XOC3-ATM          0.225 1-I/O FPGA             1.24        1.24
---- ---------------------- ------ ------------------ ----------- --------------
   4 7600-SIP-200           0.550 1-I/O FPGA             1.1         1.1
                                   2-EOS FPGA             1.211       1.211
                                   3-PEGASUS TX FPGA      1.129       1.129
                                   4-PEGASUS RX FPGA      1.3         1.3
                                   5-ROMMON               1.2         1.2
---- ---------------------- ------ ------------------ ----------- --------------
 4/0 SPA-2XT3/E3            1.0   1-ROMMON               2.12        2.12
                                   2-I/O FPGA             0.24        0.24
                                   3-E3 FPGA              0.6         0.6
                                   4-T3 FPGA              0.14        0.14
---- ---------------------- ------ ------------------ ----------- --------------
```

```
 4/1 SPA-4XOC3-POS           0.209 1-I/O FPGA             3.4         3.4
---- ---------------------- ------ ------------------ ----------- --------------
 4/2 SPA-8XCHT1/E1           0.117 1-ROMMON               2.12        2.12
                                   2-I/O FPGA             1.2         1.2
==== ====================== ====== =============================================
```

This example shows the output when verifying all the FPDs for the carrier card and all the SPAs in a specific slot:

```
Router# show hw-module slot 4 fpd

==== ====================== ====== =============================================
                             H/W   Field Programmable   Current   Min. Required
Slot Card Type               Ver.  Device:"ID-Name"    Version      Version
==== ====================== ====== ================== =========== ==============
   4 7600-SIP-200            0.550 1-I/O FPGA             1.1         1.1
                                   2-EOS FPGA             1.211       1.211
                                   3-PEGASUS TX FPGA      1.129       1.129
                                   4-PEGASUS RX FPGA      1.3         1.3
                                   5-ROMMON               1.2         1.2
---- ---------------------- ------ ------------------ ----------- --------------
 4/0 SPA-2XT3/E3             1.0   1-ROMMON               2.12        2.12
                                   2-I/O FPGA             0.24        0.24
                                   3-E3  FPGA             0.6         0.6
                                   4-T3  FPGA             0.14        0.14
---- ---------------------- ------ ------------------ ----------- --------------
 4/1 SPA-4XOC3-POS           0.209 1-I/O FPGA             3.4         3.4
---- ---------------------- ------ ------------------ ----------- --------------
 4/2 SPA-8XCHT1/E1           0.117 1-ROMMON               2.12        2.12
                                   2-I/O FPGA             1.2         1.2
==== ====================== ====== =============================================
```

This example shows the output when using the *slot-number/subslot-number* argument to identify a particular SPA:

```
Router# show hw-module subslot 4/2 fpd

==== ====================== ====== =============================================
                             H/W   Field Programmable   Current   Min. Required
Slot Card Type               Ver.  Device:"ID-Name"    Version      Version
==== ====================== ====== ================== =========== ==============
 4/2 SPA-8XCHT1/E1           0.117 1-ROMMON               2.12        2.12
                                   2-I/O FPGA             1.2         1.2
==== ====================== ====== =============================================
```

The output display in this example shows that the SIP in slot 4 is disabled because one of the programmable devices does not meet the minimum version requirements. The output also contains a “NOTES” section that provides the name of the FPD image package file needed to upgrade the disabled SIP’s FPD image.

```
Router# show hw-module all fpd

==== ====================== ====== =============================================
                             H/W   Field Programmable   Current   Min. Required
Slot Card Type               Ver.  Device:"ID-Name"    Version      Version
==== ====================== ====== ================== =========== ==============
   1 7600-SIP-200            0.550 1-I/O FPGA             1.1         1.1
                                   2-EOS FPGA             1.211       1.211
                                   3-PEGASUS TX FPGA      1.129       1.129
                                   4-PEGASUS RX FPGA      1.3         1.3
                                   5-ROMMON               1.2         1.2
```

```
---- ---------------------- ------ ------------------ ----------- --------------
 1/1 SPA-2XOC3-ATM           0.225 1-I/O FPGA             1.24        1.24
---- ---------------------- ------ ------------------ ----------- --------------
   4 7600-SIP... <DISABLED>  0.550 1-I/O FPGA             1.1         1.1
                                   2-EOS FPGA             1.211       1.211
                                   3-PEGASUS TX FPGA      1.129       1.129
                                   4-PEGASUS RX FPGA      1.3         1.3
                                   5-ROMMON               1.1         1.2     *
==== ====================== ====== =============================================
 NOTES:
       - FPD images that are required to be upgraded are indicated with a '*'
         character in the "Minimal Required Version" field.
       - The following FPD image package file is required for the upgrade:
         "c7600-fpd-pkg.122-18.SXE.pkg"
```

Displaying Information About the Default FPD Image Package

You can use the **show upgrade fpd package default** command to find out which SIPs and SPAs are supported with your current Cisco IOS release and which FPD image package you need for an upgrade.

```
Router# show upgrade fpd package default

*************************************************************************
This IOS release requires the following default FPD Image Package for
the automatic upgrade of FPD images:
*************************************************************************

Version:12.2(SXE)

Package Filename:c7600-fpd-pkg.122-18.SXE.pkg

   List of card type supported in this package:

                                Minimal
         No. Card Type          HW Ver.
        ---- ------------------ -------
          1) 2 port adapter Enh   1.0
          2) 2xCT3 SPA            0.100
          3) 2xCT3 SPA            0.200
          4) 4xCT3 SPA            0.100
          5) 4xCT3 SPA            0.200

<additional output removed for readability>
```

Verifying the FPD Image Upgrade Progress

You can use the **show upgrade fpd progress** command to view a snapshot of the upgrade progress while an FPD image upgrade is taking place. The following example shows the type of information this command displays:

```
Router# show upgrade fpd progress

FPD Image Upgrade Progress Table:

 ==== ====================
====================================================
                                               Approx.
                          Field Programmable    Time     Elapsed
```

```
Slot Card Type            Device : "ID-Name"   Needed      Time    State
==== =================== ================== ========== =====================
 1/1 SPA-2XOC3-ATM       1-I/O FPGA          00:06:30   00:01:25 Updating...
---- ------------------- ------------------ ----------- --------------------
 4/0 SPA-2XT3/E3         1-ROMMON            00:00:30   00:00:02 Completed
                         2-I/O FPGA          00:01:00   00:00:01 Updating...
                         3-E3 FPGA           00:00:30   --:--:-- Waiting...
                         4-T3 FPGA           00:00:30   --:--:-- Waiting...
---- ------------------- ------------------ ----------- --------------------
 4/2 SPA-8XCHT1/E1       1-ROMMON            --:--:--   --:--:-- Waiting...
                         2-I/O FPGA          --:--:--   --:--:-- Waiting...
==== =======================================================================
```

FPD Image Upgrade Examples

This section provides examples of automatic and manual FPD image upgrades. It includes the following examples:

- System Cannot Locate FPD Image Package File for an Automatic FPD Image Upgrade Example, page 31-13
- Automatic FPD Image Upgrade Example, page 31-13
- Manual FPD Image Upgrade Example, page 31-14
- Pending FPD Upgrade Example, page 31-15

System Cannot Locate FPD Image Package File for an Automatic FPD Image Upgrade Example

The following example displays the output when a SIP-200 requires an FPD upgrade and the **upgrade fpd auto** command is enabled, but the system cannot find the FPD image package file:

```
Mar 25 16:14:13:%FPD_MGMT-3-INCOMP_IMG_VER:Incompatible ROMMON (FPD ID=5) image version
detected for 7600-SIP-200 card in slot 1. Detected version = 1.1, minimum required version
= 1.2. Current HW version = 0.550.
Mar 25 16:14:13:%FPD_MGMT-5-UPGRADE_ATTEMPT:Attempting to automatically upgrade the FPD
image(s) for 7600-SIP-200 card in slot 1. Use 'show upgrade fpd progress' command to view
the upgrade progress ...
Mar 25 16:14:14:%FPD_MGMT-3-PKG_FILE_SEARCH_FAILED:FPD image package
(c7600-fpd-pkg.122-18.SXE.pkg) cannot be found in system's flash card or disk to do FPD
upgrade.
Mar 25 16:14:14:%OIR-6-REMCARD:Card removed from slot 1, interfaces disabled
Mar 25 16:14:14:%FPD_MGMT-5-CARD_DISABLED:7600-SIP-200 card in slot 1 is being disabled
because of an incompatible FPD image version. Note that the c7600-fpd-pkg.122-18.SXE.pkg
package will be required if you want to perform the upgrade operation.
Mar 25 16:14:14:%C6KPWR-SP-4-DISABLED:power to module in slot 1 set off (FPD Upgrade
Failed)
```

Automatic FPD Image Upgrade Example

The following example shows the output displayed when a SIP-200 requires an FPD image upgrade and the **upgrade fpd auto** command is enabled. In this example, the switch has been configured to locate the FPD image package from a TFTP server, but most of the output would be similar regardless of the location of the FPD image package. The required FPD image is automatically upgraded.

```
Mar 25 16:22:48:%FPD_MGMT-3-INCOMP_IMG_VER:Incompatible ROMMON (FPD ID=5) image version
detected for 7600-SIP-200 card in slot 1. Detected version = 1.1, minimum required version
= 1.2. Current HW version = 0.550.
```

```
Mar 25 16:22:48:%FPD_MGMT-5-UPGRADE_ATTEMPT:Attempting to automatically upgrade the FPD
image(s) for 7600-SIP-200 card in slot 1. Use 'show upgrade fpd progress' command to view
the upgrade progress ...
Mar 25 16:22:48:%FPD_MGMT-6-BUNDLE_DOWNLOAD:Downloading FPD image bundle for 7600-SIP-200
card in slot 1 ...
Loading muck/luislu/c7600-fpd-pkg.122-18.SXE.pkg from 223.255.254.254 (via
GigabitEthernet5/1):!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Mar 25 16:23:17:%FPD_MGMT-6-UPGRADE_TIME:Estimated total FPD image upgrade time for
7600-SIP-200 card in slot 1 = 00:02:00.
Mar 25 16:23:17:%FPD_MGMT-6-UPGRADE_START:ROMMON (FPD ID=5) image upgrade in progress for
7600-SIP-200 card in slot 1. Updating to version 1.2. PLEASE DO NOT INTERRUPT DURING THE
UPGRADE PROCESS (estimated upgrade completion time = 00:02:00) ...
Mar 25 16:23:25:%FPD_MGMT-6-UPGRADE_PASSED:ROMMON (FPD ID=5) image in the 7600-SIP-200
card in slot 1 has been successfully updated from version 1.1 to version 1.2. Upgrading
time = 00:00:08.452
Mar 25 16:23:25:%FPD_MGMT-6-OVERALL_UPGRADE:All the attempts to upgrade the required FPD
images have been completed for 7600-SIP-200 card in slot 1. Number of successful/failure
upgrade(s):1/0.
Mar 25 16:23:26:%FPD_MGMT-5-CARD_POWER_CYCLE:7600-SIP-200 card in slot 1 is being power
cycled for the FPD image upgrade to take effect.
Mar 25 16:23:26:%OIR-6-REMCARD:Card removed from slot 1, interfaces disabled
Mar 25 16:23:26:%C6KPWR-SP-4-DISABLED:power to module in slot 1 set off (Reset)
Mar 25 16:24:16:%CWAN_RP-6-CARDRELOAD:Module reloaded on slot 1/0
Mar 25 16:24:18:%DIAG-SP-6-RUN_COMPLETE:Module 1:Running Complete Diagnostics...
Mar 25 16:24:18:%DIAG-SP-6-DIAG_OK:Module 1:Passed Online Diagnostics
Mar 25 16:24:19:%OIR-SP-6-INSCARD:Card inserted in slot 1, interfaces are now online
```

Manual FPD Image Upgrade Example

In the following example, FPD for the T1/E1 SPA in subslot 4/2 is upgraded manually from the FPD image package file that was placed on disk0:

```
Router# upgrade hw-module subslot 4/2 file disk0:c7600-fpd-pkg.122-18.SXE.pkg

% The following FPD(s) will be upgraded for SPA-8XCHT1/E1 (H/W ver = 0.117) in subslot
4/2:

    ================== =========== =========== ============
    Field Programmable   Current     Upgrade   Estimated
    Device:"ID-Name"    Version     Version   Upgrade Time
    ================== =========== =========== ============
    1-ROMMON               2.11        2.12      00:00:20
    2-I/O FPGA             1.1         1.2       00:01:00
    ================== =========== =========== ============


% Are you sure that you want to perform this operation? [no]:y
% Restarting the target card in subslot 4/2 for FPD image upgrade. Please wait ...

Router#
Mar 25 17:01:01:%FPD_MGMT-6-UPGRADE_TIME:Estimated total FPD image upgrade time for
SPA-8XCHT1/E1 card in subslot 4/2 = 00:01:20.
```

```
Mar 25 17:01:01:%FPD_MGMT-6-UPGRADE_START:ROMMON (FPD ID=1) image upgrade in progress for
SPA-8XCHT1/E1 card in subslot 4/2. Updating to version 2.12. PLEASE DO NOT INTERRUPT
DURING THE UPGRADE PROCESS (estimated upgrade completion time = 00:00:20) ...
Router#
Mar 25 17:01:04:%FPD_MGMT-6-UPGRADE_PASSED:ROMMON (FPD ID=1) image in the SPA-8XCHT1/E1
card in subslot 4/2 has been successfully updated from version 2.11 to version 2.12.
Upgrading time = 00:00:03.092
Mar 25 17:01:04:%FPD_MGMT-6-UPGRADE_START:I/O FPGA (FPD ID=2) image upgrade in progress
for SPA-8XCHT1/E1 card in subslot 4/2. Updating to version 1.2. PLEASE DO NOT INTERRUPT
DURING THE UPGRADE PROCESS (estimated upgrade completion time = 00:01:00) ...
Router#
Mar 25 17:01:26:%FPD_MGMT-6-UPGRADE_PASSED:I/O FPGA (FPD ID=2) image in the SPA-8XCHT1/E1
card in subslot 4/2 has been successfully updated from version 1.1 to version 1.2.
Upgrading time = 00:00:22.580
Mar 25 17:01:26:%FPD_MGMT-6-OVERALL_UPGRADE:All the attempts to upgrade the required FPD
images have been completed for SPA-8XCHT1/E1 card in subslot 4/2. Number of
successful/failure upgrade(s):2/0.
Router#
Mar 25 17:01:26:%FPD_MGMT-5-CARD_POWER_CYCLE:SPA-8XCHT1/E1 card in subslot 4/2 is being
power cycled for the FPD image upgrade to take effect.
```

Pending FPD Upgrade Example

In the following example, some FPD images are waiting for upgrades because the FPD upgrade process is upgrading another FPD on the same card (up to four FPD upgrades can occur at once, but the upgrades have to occur on hardware in different line card slots). In this particular example, the FPD upgrade process is happening on a SIP-200.

```
Mar 25 17:04:59:%FPD_MGMT-6-UPGRADE_TIME:Estimated total FPD image
upgrade time for 7600-SIP-200 card in slot 1 = 00:10:00.
Mar 25 17:04:59:%FPD_MGMT-6-UPGRADE_START:ROMMON (FPD ID=5) image
upgrade in progress for 7600-SIP-200 card in slot 1. Updating to version
1.2. PLEASE DO NOT INTERRUPT DURING THE UPGRADE PROCESS (estimated
upgrade completion time = 00:02:00) ...
Mar 25 17:05:08:%FPD_MGMT-6-UPGRADE_PASSED:ROMMON (FPD ID=5) image in
the 7600-SIP-200 card in slot 1 has been successfully updated from
version 1.1 to version 1.2. Upgrading time = 00:00:08.884
Mar 25 17:05:08:%FPD_MGMT-6-PENDING_UPGRADE:4 more FPD image upgrade
operation will be required on 7600-SIP-200 in slot 1 after additional
power-cycle operation on the target card.
Mar 25 17:05:08:%FPD_MGMT-5-CARD_POWER_CYCLE:7600-SIP-200 card in slot
1 is being power cycled for the FPD image upgrade to take effect.
Mar 25 17:05:08:%OIR-6-REMCARD:Card removed from slot 1, interfaces
disabled
Mar 25 17:05:08:%C6KPWR-SP-4-DISABLED:power to module in slot 1 set
off (Reset)
Mar 25 17:05:59:%CWAN_RP-6-CARDRELOAD:Module reloaded on slot 1/0
Mar 25 17:06:02:%FPD_MGMT-6-UPGRADE_TIME:Estimated total FPD image
upgrade time for 7600-SIP-200 card in slot 1 = 00:10:00.
Mar 25 17:06:02:%FPD_MGMT-6-UPGRADE_START:I/O FPGA (FPD ID=1) image
upgrade in progress for 7600-SIP-200 card in slot 1. Updating to version
1.1. PLEASE DO NOT INTERRUPT DURING THE UPGRADE PROCESS (estimated
upgrade completion time = 00:02:00) ...
Mar 25 17:06:21:%FPD_MGMT-6-UPGRADE_PASSED:I/O FPGA (FPD ID=1) image
in the 7600-SIP-200 card in slot 1 has been successfully updated from
version 1.0 to version 1.1. Upgrading time = 00:00:18.592
Mar 25 17:06:21:%FPD_MGMT-6-UPGRADE_START:EOS FPGA (FPD ID=2) image
upgrade in progress for 7600-SIP-200 card in slot 1. Updating to version
1.211. PLEASE DO NOT INTERRUPT DURING THE UPGRADE PROCESS (estimated
upgrade completion time = 00:02:00) ...
Mar 25 17:07:18:%FPD_MGMT-6-UPGRADE_PASSED:EOS FPGA (FPD ID=2) image
in the 7600-SIP-200 card in slot 1 has been successfully updated from
```

```
version 1.210 to version 1.211. Upgrading time = 00:00:56.812
Mar 25 17:07:18:%FPD_MGMT-6-UPGRADE_START:PEGASUS TX FPGA (FPD ID=3)
image upgrade in progress for 7600-SIP-200 card in slot 1. Updating to
version 1.129. PLEASE DO NOT INTERRUPT DURING THE UPGRADE PROCESS
(estimated upgrade completion time = 00:02:00) ...
Mar 25 17:08:17:%FPD_MGMT-6-UPGRADE_PASSED:PEGASUS TX FPGA (FPD ID=3)
image in the 7600-SIP-200 card in slot 1 has been successfully updated
from version 1.120 to version 1.129. Upgrading time = 00:00:59.188
Mar 25 17:08:17:%FPD_MGMT-6-UPGRADE_START:PEGASUS RX FPGA (FPD ID=4)
image upgrade in progress for 7600-SIP-200 card in slot 1. Updating to
version 1.3. PLEASE DO NOT INTERRUPT DURING THE UPGRADE PROCESS
(estimated upgrade completion time = 00:02:00) ...
Mar 25 17:09:03:%FPD_MGMT-6-UPGRADE_PASSED:PEGASUS RX FPGA (FPD ID=4)
image in the 7600-SIP-200 card in slot 1 has been successfully updated
from version 1.2 to version 1.3. Upgrading time = 00:00:45.396
Mar 25 17:09:03:%FPD_MGMT-6-OVERALL_UPGRADE:All the attempts to
upgrade the required FPD images have been completed for 7600-SIP-200
card in slot 1. Number of successful/failure upgrade(s):5/0.
Mar 25 17:09:03:%FPD_MGMT-5-CARD_POWER_CYCLE:7600-SIP-200 card in slot
1 is being power cycled for the FPD image upgrade to take effect.
```

Troubleshooting Problems with FPD Image Upgrades

This section contains information to help troubleshoot problems that can occur during the upgrade process.

Power Failure or Removal of a SIP or SPA During an FPD Image Upgrade

These instructions should only be used if a previous upgrade attempt has failed due to an external factor such as a power failure or a jacket card or SPA removal.

If the FPD upgrade operation is interrupted by a power failure or the removal of the SIP or SPA, it could corrupt the FPD image. This corruption of the FPD image file makes the SIP or SPA unusable by the switch and the system will display the following messages when it tries to power up the SIP or SPA:

Note To find more information about FPD-related messages, check the system error messages guide for your Cisco IOS software release.

```
Mar 29 11:30:36:%SPA_OIR-3-RECOVERY_RELOAD:subslot 4/1:Attempting
recovery by reloading SPA
Mar 29 11:30:51:%SPA_OIR-3-HW_INIT_TIMEOUT:subslot 4/1
Mar 29 11:30:56:%SPA_OIR-3-RECOVERY_RELOAD:subslot 4/1:Attempting
recovery by reloading SPA
Mar 29 11:31:11:%SPA_OIR-3-HW_INIT_TIMEOUT:subslot 4/1
Mar 29 11:31:16:%SPA_OIR-3-RECOVERY_RELOAD:subslot 4/1:Attempting
recovery by reloading SPA
Mar 29 11:31:31:%SPA_OIR-3-HW_INIT_TIMEOUT:subslot 4/1
Mar 29 11:31:31:%SPA_OIR-3-SPA_POWERED_OFF:subslot 4/1:SPA 4xOC3 POS
SPA powered off after 5 failures within 600 seconds
```

The **show hw-module all fpd** command can be used to verify that the SIP or SPA is using a corrupted FPD image. In this example, the SPA in slot 4/1 is corrupted.

```
Router# show hw-module all fpd

==== ===================== ====== =============================================
                             H/W   Field Programmable   Current   Min. Required
Slot Card Type               Ver.  Device:"ID-Name"    Version      Version
==== ====================== ====== ================== ========== ==============
   4 7600-SIP-200            0.550 1-I/O FPGA            1.1         1.1
                                   2-EOS FPGA            1.211       1.211
                                   3-PEGASUS TX FPGA     1.129       1.129
                                   4-PEGASUS RX FPGA     1.3         1.3
                                   5-ROMMON              1.2         1.2
---- ---------------------- ------ ------------------ ---------- --------------
 4/1 SPA-4XOC3<DISABLED>  ?.?   ????????????        ?.?         ?.?
==== ====================== ====== ===========================================
```

Performing a FPD Recovery Upgrade

The recovery upgrade procedure can only be performed on a SIP or SPA that has been powered off by the system after it has failed all of the retries attempted to initialize the SIP or SPA.

The following example displays the output of an attempt to perform a recovery upgrade before all the initialization retries have been attempted for the SPA in subslot 4/1.

Note Other factors can cause the system to ask “Do you want to perform the recovery upgrade operation?” Only answer **y** to this question if you have attempted an FPD upgrade that has failed due to a power failure or a SIP or SPA removal.
If you are prompted for this question without previously failing an upgrade attempt for one of these reasons, contact Cisco Technical Support.

```
Mar 29 11:29:55:%SPA_OIR-3-RECOVERY_RELOAD:subslot 4/1:Attempting
recovery by reloading SPA
Mar 29 11:30:10:%SPA_OIR-3-HW_INIT_TIMEOUT:subslot 4/1
Mar 29 11:30:15:%SPA_OIR-3-RECOVERY_RELOAD:subslot 4/1:Attempting
recovery by reloading SPA
Mar 29 11:30:31:%SPA_OIR-3-HW_INIT_TIMEOUT:subslot 4/1
Router# upgrade hw-module subslot 4/1 file disk0:c7600-fpd-pkg.122-18.SXE.pkg

% Cannot get FPD version information for version checking. If a previous
upgrade attempt has failed for the target card, then a recovery upgrade
would be required to fix the failure.

% The following FPD(s) will be upgraded for SPA-4XOC3-POS (H/W ver =
0.209) in subslot 4/1:

    ================== =========== =========== ============
    Field Programmable   Current     Upgrade   Estimated
    Device:"ID-Name"    Version     Version   Upgrade Time
    ================== =========== =========== ============
    1-I/O FPGA            ?.?         3.4       00:02:00
    ================== =========== =========== ============


% Do you want to perform the recovery upgrade operation? [no]:y
% Cannot perform recovery upgrade operation because the target card is
not in a failed state. Please try again later.
```

Once the following error message is displayed, you can perform the recovery upgrade:

Note You must wait to see this error message before you attempt the upgrade.

```
Mar 29 11:31:31:%SPA_OIR-3-SPA_POWERED_OFF:subslot 4/1:SPA 4xOC3 POS SPA powered off after
5 failures within 600 seconds
```

Perform the manual FPD image upgrade method using the **upgrade hw-module subslot** command to recover from a corrupted image after the SIP or SPA has been powered off by the system. In this command, *slot-number* is the slot where the SIP is installed, *subslot-number* is the subslot of the SIP where the SPA is located, and *file-url* is the location of the FPD image package file.

Note Before proceeding with this operation, make sure that the correct version of the FPD image package file has been obtained for the corresponding Cisco IOS release that the system is using.

The following example displays the console output of a recovery upgrade operation:

```
Router# upgrade hw-module subslot 4/1 file disk0:c7600-fpd-pkg.122-18.SXE.pkg

% Cannot get FPD version information for version checking. If a previous
upgrade attempt has failed for the target card, then a recovery upgrade
would be required to fix the failure.


% The following FPD(s) will be upgraded for SPA-4XOC3-POS (H/W ver =
0.209) in subslot 4/1:

    ================== =========== ========== ============
    Field Programmable   Current     Upgrade   Estimated
    Device:"ID-Name"    Version     Version   Upgrade Time
    ================== =========== ========== ============
    1-I/O FPGA             ?.?         3.4       00:02:00
    ================== =========== ========== ============


% Do you want to perform the recovery upgrade operation? [no]:y
% Proceeding with recovery upgrade operation ...

Router#
Mar 29 11:37:51:%FPD_MGMT-6-UPGRADE_TIME:Estimated total FPD image
upgrade time for SPA-4XOC3-POS card in subslot 4/1 = 00:02:00.
Mar 29 11:37:51:%FPD_MGMT-6-UPGRADE_START:Unknown FPD (FPD ID=1) image
upgrade in progress for SPA-4XOC3-POS card in subslot 4/1. Updating to
version 3.4. PLEASE DO NOT INTERRUPT DURING THE UPGRADE PROCESS
(estimated upgrade completion time = 00:02:00) ...
Router#
Mar 29 11:39:11:%FPD_MGMT-6-UPGRADE_PASSED:Unknown FPD (FPD ID=1)
image in the SPA-4XOC3-POS card in subslot 4/1 has been successfully
updated from version ?.? to version 3.4. Upgrading time = 00:01:19.528
Mar 29 11:39:11:%FPD_MGMT-6-OVERALL_UPGRADE:All the attempts to
upgrade the required FPD images have been completed for SPA-4XOC3-POS
card in subslot 4/1. Number of successful/failure upgrade(s):1/0.
Mar 29 11:39:11:%FPD_MGMT-5-CARD_POWER_CYCLE:SPA-4XOC3-POS card in
subslot 4/1 is being power cycled for the FPD image upgrade to take
effect.
```

Verifying a Successful Upgrade

After the upgrade process is complete, you can use the **show hw-module all fpd** command to verify that the FPD image has been successfully upgraded:

```
Router# show hw-module all fpd

==== ====================== ====== =============================================
                             H/W   Field Programmable   Current   Min. Required
Slot Card Type               Ver.  Device:"ID-Name"    Version      Version
==== ====================== ====== ================== =========== ==============
   4 7600-SIP-200            0.550 1-I/O FPGA             1.1         1.1
                                   2-EOS FPGA             1.211       1.211
                                   3-PEGASUS TX FPGA      1.129       1.129
                                   4-PEGASUS RX FPGA      1.3         1.3
                                   5-ROMMON               1.2         1.2
---- ---------------------- ------ ------------------ ----------- --------------
4/1 SPA-4XOC3-POS           0.209  1-I/O FPGA             3.4         3.4
==== ====================== ====== =============================================
```

PART 9

Glossary

GLOSSARY

B

blank filler plate An empty panel used to fill vacant subslots on a SIP. For proper operation, a SIP should be fully installed with either functional SPAs or blank filler plates.

D

double height Describes the dimension of a SPA that occupies two, vertically aligned SIP subslots.

F

FPD Field-programmable device. General term for any hardware component implemented on switch cards that supports separate software upgrades. SIPs and SPAs must have the right FPD version to function properly; an FPD incompatibility will disable all interfaces on the SPA or all SPAs within the SIP.

FPD image package An FPD image package is used to upgrade FPD images. Whenever a Cisco IOS image is released that supports SPAs, a companion SPA FPD image package is also released for that Cisco IOS software release.

O

OIR Online insertion and removal. Feature supported by SIPs and SPAs allowing removal of the cards while the switch and the cards are activated, without affecting the operation of other cards or the switch. Although this removal can be done while the SIP or SPA is activated, it is generally recommended that you gracefully deactivate the hardware using the appropriate commands for your platform prior to removal of the hardware.

S

SFP Small form-factor pluggable optical transceiver. A type of fiber-optic receptacle device that mounts flush with the front panel to provide network connectivity.

single height Describes the dimension of a SPA that occupies a single SIP subslot, or half of the SIP.

SIP SPA interface processor. A SIP is a platform-specific carrier card that inserts into a switch slot like a line card. A SIP can hold one or more SPAs in its subslots, depending on the SIP type. The SPA provides the network interface. The SIP provides the connection between the route processor (RP) and the SPA.

SPA Shared port adapter. A SPA is a modular, platform-independent port adapter that inserts into a subslot of a compatible SIP carrier card to provide network connectivity and increased interface port density. The SPA provides the interface between the network and the SIP.

subslot Secondary slot on a SIP where a SPA is installed. The primary slot is the chassis slot on the switch.

INDEX