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.

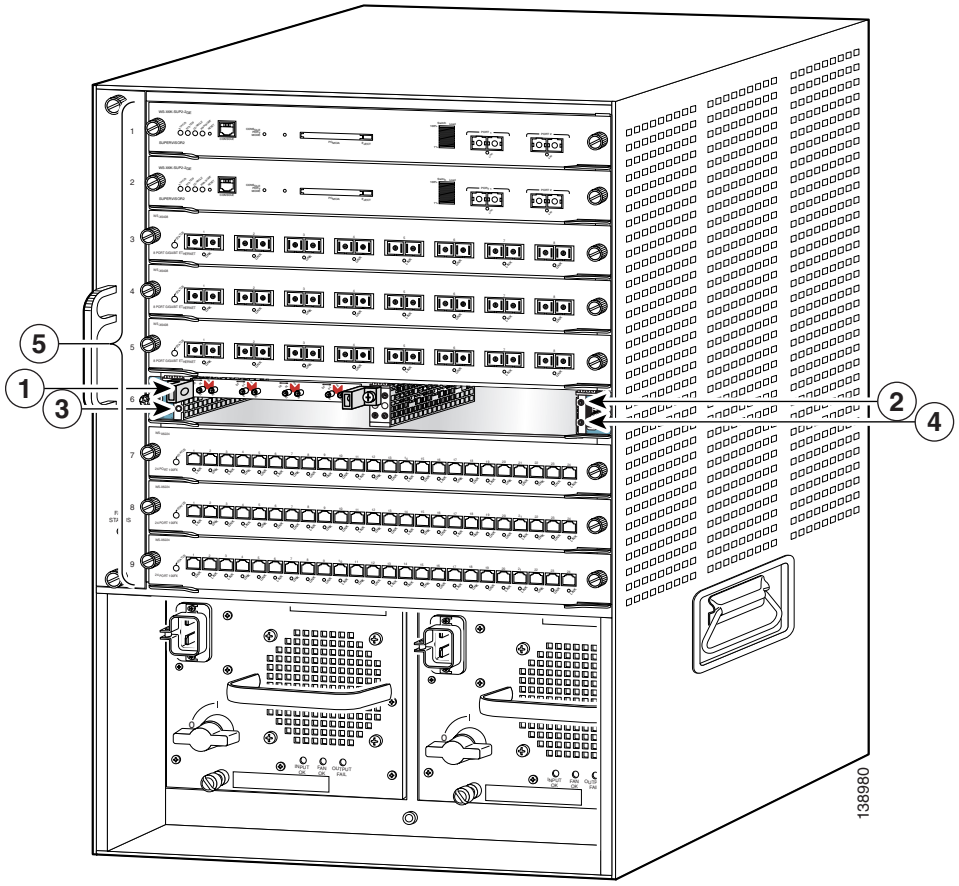
Specifying the Slot Location for a SIP or SSC

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

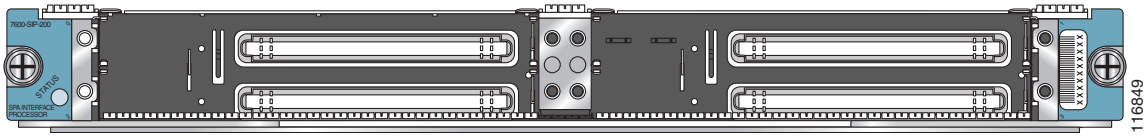
SIP subslots begin their numbering with 0 and have a horizontal or vertical orientation depending on the orientation of the SIP in the router chassis slot.

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.

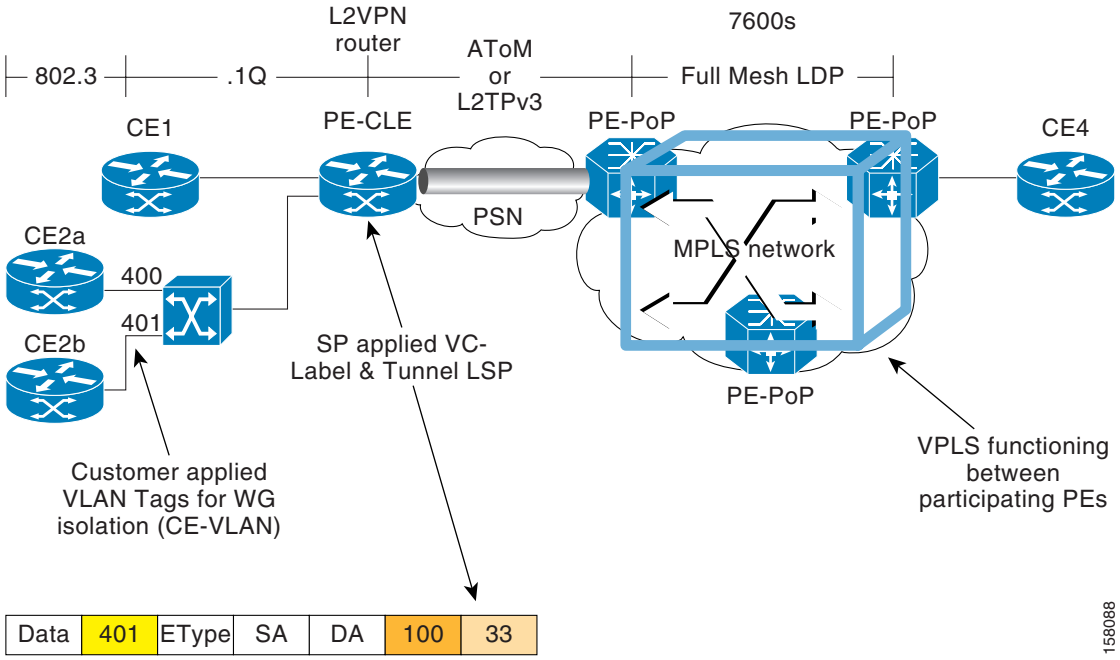
In Cisco IOS Release 12.2(33)SXH and later releases, if there are a number of customer sites connecting to the spoke, you can terminate multiple VCs per spoke into the same VFI or VPLS instance at the hub.

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 Gi2/0/0
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 Gi2/0/0
Building configuration...

Current configuration: 219 bytes
!
interface GigabitEthernet2/0/0
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 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.	• 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-transmit** 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/en/US/docs/ios/12_1/12_1e11/feature/guide/dtnbarad.html

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 "HQoS for EoMPLS Virtual Circuits" 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#wp1159857

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
```