Optical Services Modules Software Configuration Note, 12.2SX

February 2009

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

Text Part Number: OL-5347-21

CONTENTS

Preface

This preface describes who should read the *Optical Services Modules Software Configuration Note, 12.2SX*, how it is organized, and its document conventions.

Document Revision History

The Document Revision History table below records technical changes to this document. The table shows the document revision number for the change, the date of the change, and a brief summary of the change. Note that not all Cisco documents use a Document Revision History table.

Revision	Date	Change Summary
OL-5347-21	January, 2006	Added RFC 1483 Spanning-Tree Interoperability Enhancements to Chapter 8. Added Configuring Dot1q Transparency for EoMPLS to Chapter 11.
OL-5347-20	September, 2005	Added information about H-VPLS to Chapter 11.
OL-5347-19	August, 2005	Added notes to HDLC and AToM sections in Chapter 11 for SUP720-PFC3B and SUP720-PFC3BXL core-facing configurations.
OL-5347-19	May, 2005	Added Distribution of Remaining Bandwidth to Chapter 9.
OL-5347-18	April, 2005	• Added the following information to Chapter 3: – Configuring Multipoint Bridging – Configuring Strict Priority LLQ Support on POS Optical Service Modules • Added Configuring Strict Priority Low Latency Queuing (LLQ) Support on the OSM-2+4GE-WAN+ to Chapter 4. • Added Configuring Multipoint Bridging to Chapter 8. • Added the following information to Chapter 9: – Cisco IPv6 QoS on the OSMs – Configuring QoS: Match VLAN • Added the following information to Chapter 11: – HQoS for EoMPLS Virtual Circuits – Support bandwidth command in HQoS parent class at Supported Features – DE/CLP and EXP Mapping on FR/ATMoMPLS VC – HDLC Over MPLS – PPP Over MPLS
OL-5347-17	March, 2005	• Changed text for Bridging Control Protocol Usage Guidelines and Restrictions section in Chapter 3. • Added note to Bridging Control Protocol Usage Guidelines and Restrictions section in Chapter 3. • Added Document Revision History table.

Audience

This publication is for experienced network administrators who are responsible for configuring and maintaining the Optical Services Modules (OSMs) for the Cisco 7600 series router and Catalyst 6000 family switches.

Only trained and qualified service personnel (as defined in IEC 60950 and AS/NZS3260) should install, replace, or service the equipment described in this publication.

Organization

This publication is organized as follows:

Chapter	Title	Description
Chapter 1	Product Overview	Provides an overview of the OSM.
Chapter 2	Basic Configurations	Describes how to perform basic configurations on the OSMs.
Chapter 3	Configuring the OC-3c/STM-1, OC-12c/STM-4, and OC-48c/STM-16 SONET/SDH Optical Services Modules	Describes how to configure the POS/SDH modules.
Chapter 4	Configuring 4-Port Gigabit Ethernet WAN Optical Services Modules	Describes how to configure the 4-port Gigabit Ethernet WAN modules.
Chapter 5	Configuring the Channelized OC-12/T3 SONET/SDH Optical Services Modules	Describes how to configure the OC-12 and OC-48 channelized modules.
Chapter 6	Configuring the Channelized OC-12/T1 Optical Services Modules	Describes how to configure the channelized OSM-1CHOC12/T1-SI SONET/SDH Optical Services Modules (OSMs).
Chapter 7	Configuring the Channelized 12-port CT3/T1 Optical Services Modules	Describes how to configure the 12-port channelized and unchannelized DS3 Optical Services Modules (OSM-12CT3/T1).
Chapter 8	Configuring the OC-12 ATM Optical Services Modules	Describes how to configure the OC-12 ATM WAN modules.
Chapter 9	Configuring QoS on the Optical Services Modules	Describes how to configure quality of service (QoS) on the OSMs.
Chapter 10	Configuring Destination Sensitive Services on the Optical Services Modules	Describes how to configure Destination Sensitive Services (DSS) on the OSMs.
Chapter 11	Configuring Multiprotocol Label Switching on the Optical Services Modules	Describes how to configure MPLS and EoMPLS on the OSMs.

Related Documentation

The following publications are available for the OSMs:

- *Optical Services Module Installation and Verification Note*
- *Cisco 7600 Series Router Module Installation Guide*
- *Cisco 7600 Series Router Command Reference*
- *Cisco 7600 Series Router System Message Guide*
- *Catalyst 6000 Family and Cisco 7600 Series Router MSFC Release Notes*
- *Catalyst 6000 Family Quick Software Configuration*
- *Catalyst 6000 Family Module Installation Guide*
- *Catalyst 6000 Family Software Configuration Guide*
- *Catalyst 6000 Family Command Reference*
- *System Message Guide—Catalyst 6000 Family, 4000 Family, 2926G Series, and 2980G Switches*

- *Release Notes for Catalyst 6000 Family Software Release 6.x*
- *Regulatory Compliance and Safety Information for the Cisco 7600 Series Router*
- *Regulatory Compliance and Safety Information for the Catalyst 6000 Family Switches*
- Cisco IOS Configuration Guides and Command References—Use these publications to help you configure the Cisco IOS software that runs on the MSFC.
- For information about MIBs, refer to http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Conventions

Note Throughout this publication, the term *supervisor engine* is used to refer to Supervisor Engine 2.

This publication uses the following conventions:

Convention	Description
boldface font	Commands and keywords are in **boldface**.
italic font	Arguments for which you supply values are in *italics*.
[ ]	Elements in square brackets are optional.
{ x | y | z }	Alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`screen` font	Terminal sessions and information the system displays are in `screen` font.
`boldface screen` font	Information you must enter is in **`boldface screen`** font.
italic screen font	Arguments for which you supply values are in *italic screen* font.
——→	`This pointer highlights an important line of text in an example.`
^	The symbol ^ represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Nonprinting characters, such as passwords are in angle brackets.

Notes use the following conventions:

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

Cautions use the following conventions:

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

Warnings use the following conventions:

Warning **This warning symbol means *danger*. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. To see translations of the warnings that appear in this publication, refer to the *Regulatory Compliance and Safety Information* document that accompanied this device.**

Waarschuwing **Dit waarschuwingssymbool betekent gevaar. U verkeert in een situatie die lichamelijk letsel kan veroorzaken. Voordat u aan enige apparatuur gaat werken, dient u zich bewust te zijn van de bij elektrische schakelingen betrokken risico's en dient u op de hoogte te zijn van standaard maatregelen om ongelukken te voorkomen. Voor vertalingen van de waarschuwingen die in deze publicatie verschijnen, kunt u het document *Regulatory Compliance and Safety Information* (Informatie over naleving van veiligheids- en andere voorschriften) raadplegen dat bij dit toestel is ingesloten.**

Varoitus **Tämä varoitusmerkki merkitsee vaaraa. Olet tilanteessa, joka voi johtaa ruumiinvammaan. Ennen kuin työskentelet minkään laitteiston parissa, ota selvää sähkökytkentöihin liittyvistä vaaroista ja tavanomaisista onnettomuuksien ehkäisykeinoista. Tässä julkaisussa esiintyvien varoitusten käännökset löydät laitteen mukana olevasta *Regulatory Compliance and Safety Information* -kirjasesta (määräysten noudattaminen ja tietoa turvallisuudesta).**

Attention **Ce symbole d'avertissement indique un danger. Vous vous trouvez dans une situation pouvant causer des blessures ou des dommages corporels. Avant de travailler sur un équipement, soyez conscient des dangers posés par les circuits électriques et familiarisez-vous avec les procédures couramment utilisées pour éviter les accidents. Pour prendre connaissance des traductions d'avertissements figurant dans cette publication, consultez le document *Regulatory Compliance and Safety Information* (Conformité aux règlements et consignes de sécurité) qui accompagne cet appareil.**

Warnung **Dieses Warnsymbol bedeutet Gefahr. Sie befinden sich in einer Situation, die zu einer Körperverletzung führen könnte. Bevor Sie mit der Arbeit an irgendeinem Gerät beginnen, seien Sie sich der mit elektrischen Stromkreisen verbundenen Gefahren und der Standardpraktiken zur Vermeidung von Unfällen bewußt. Übersetzungen der in dieser Veröffentlichung enthaltenen Warnhinweise finden Sie im Dokument *Regulatory Compliance and Safety Information* (Informationen zu behördlichen Vorschriften und Sicherheit), das zusammen mit diesem Gerät geliefert wurde.**

Avvertenza **Questo simbolo di avvertenza indica un pericolo. La situazione potrebbe causare infortuni alle persone. Prima di lavorare su qualsiasi apparecchiatura, occorre conoscere i pericoli relativi ai circuiti elettrici ed essere al corrente delle pratiche standard per la prevenzione di incidenti. La traduzione delle avvertenze riportate in questa pubblicazione si trova nel documento *Regulatory Compliance and Safety Information* (Conformità alle norme e informazioni sulla sicurezza) che accompagna questo dispositivo.**

Advarsel **Dette varselsymbolet betyr fare. Du befinner deg i en situasjon som kan føre til personskade. Før du utfører arbeid på utstyr, må du vare oppmerksom på de faremomentene som elektriske kretser innebærer, samt gjøre deg kjent med vanlig praksis når det gjelder å unngå ulykker. Hvis du vil se oversettelser av de advarslene som finnes i denne publikasjonen, kan du se i dokumentet *Regulatory Compliance and Safety Information* (Overholdelse av forskrifter og sikkerhetsinformasjon) som ble levert med denne enheten.**

Aviso **Este símbolo de aviso indica perigo. Encontra-se numa situação que lhe poderá causar danos físicos. Antes de começar a trabalhar com qualquer equipamento, familiarize-se com os perigos relacionados com circuitos eléctricos, e com quaisquer práticas comuns que possam prevenir possíveis acidentes. Para ver as traduções dos avisos que constam desta publicação, consulte o documento *Regulatory Compliance and Safety Information* (Informação de Segurança e Disposições Reguladoras) que acompanha este dispositivo.**

¡Advertencia! **Este símbolo de aviso significa peligro. Existe riesgo para su integridad física. Antes de manipular cualquier equipo, considerar los riesgos que entraña la corriente eléctrica y familiarizarse con los procedimientos estándar de prevención de accidentes. Para ver una traducción de las advertencias que aparecen en esta publicación, consultar el documento titulado *Regulatory Compliance and Safety Information* (Información sobre seguridad y conformidad con las disposiciones reglamentarias) que se acompaña con este dispositivo.**

Varning! **Denna varningssymbol signalerar fara. Du befinner dig i en situation som kan leda till personskada. Innan du utför arbete på någon utrustning måste du vara medveten om farorna med elkretsar och känna till vanligt förfarande för att förebygga skador. Se förklaringar av de varningar som förkommer i denna publikation i dokumentet *Regulatory Compliance and Safety Information* (Efterrättelse av föreskrifter och säkerhetsinformation), vilket medföljer denna anordning.**

Obtaining Documentation

The following sections provide sources for obtaining documentation from Cisco Systems.

World Wide Web

You can access the most current Cisco documentation on the World Wide Web at the following sites:

- http://www.cisco.com
- http://www-china.cisco.com
- http://www-europe.cisco.com

Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or as an annual subscription.

Ordering Documentation

Cisco documentation is available in the following ways:

- Registered Cisco Direct Customers can order Cisco Product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco corporate headquarters (California, USA) at 408 526-7208 or, in North America, by calling 800 553-NETS(6387).

Obtaining Documentation and Submitting a Service Request

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

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

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

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain documentation, troubleshooting tips, and sample configurations from online tools by using the Cisco Technical Assistance Center (TAC) Web Site. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC Web Site.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com is a highly integrated Internet application and a powerful, easy-to-use tool that provides a broad range of features and services to help you to

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise
- Register for online skill assessment, training, and certification programs

You can self-register on Cisco.com to obtain customized information and service. To access Cisco.com, go to the following URL:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two types of support are available through the Cisco TAC: the Cisco TAC Web Site and the Cisco TAC Escalation Center.

Inquiries to Cisco TAC are categorized according to the urgency of the issue:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.
- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

Which Cisco TAC resource you choose is based on the priority of the problem and the conditions of service contracts, when applicable.

Cisco TAC Web Site

The Cisco TAC Web Site allows you to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC Web Site, go to the following URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco services contract have complete access to the technical support resources on the Cisco TAC Web Site. The Cisco TAC Web Site requires a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to the following URL to register:

http://www.cisco.com/register/

If you cannot resolve your technical issues by using the Cisco TAC Web Site, and you are a Cisco.com registered user, you can open a case online by using the TAC Case Open tool at the following URL:

http://www.cisco.com/tac/caseopen

If you have Internet access, it is recommended that you open P3 and P4 cases through the Cisco TAC Web Site.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses issues that are classified as priority level 1 or priority level 2; these classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer will automatically open a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to the following URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled; for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). In addition, please have available your service agreement number and your product serial number.

CHAPTER 1

Product Overview

The Optical Services Modules (OSMs) are supported in the Cisco 7600 series router and Catalyst 6500 series switches. The OSMs are supported with the following system configurations:

- Supervisor Engine 720, PFC3A, and MSFC3
- Supervisor Engine SUP720-3BXL and PFC3BXL
- Supervisor Engine 2, Policy Feature Card 2 (PFC2), and Multilayer Switch Feature Card 2 (MSFC2)
- Supervisor Engine 2, PFC2, MSFC2, and Switch Fabric Module (SFM) or SFM2

Refer to the *Release Notes for Catalyst 6500 and Cisco 7600 Series Router Software Release 6.x* and the *Release Notes for Catalyst 6500 and Cisco 7600 Series Router for Cisco IOS Release 12.1E* and the *Release Notes for Cisco IOS Release 12.2 SX on the Catalyst 6500 and Cisco 7600 Supervisor Engine and MSFC* publications for complete information about the chassis, modules, software features, protocols, and MIBs supported by the OSMs.

Contents

This chapter consists of these sections:

Overview

Table 1-1 describes the Cisco 7600 series router and Catalyst 6500 series chassis.

Table 1-1 Cisco 7600 Series and Catalyst 6500 Series Chassis

Chassis	Description
Cisco 7600 Series	Cisco 7603 series router—3 slots
	Cisco 7604 series router—4 slots
	Cisco 7606 series router—6 slots
	Cisco 7609 series router—9 vertical slots
	Cisco 7613 series router—13 slots
Catalyst 6500 Series	Catalyst 6504 switch—4 slots
	Catalyst 6506 switch—6 slots
	Catalyst 6509 switch—9 slots
	Catalyst 6509-NEB switch—9 vertical slots
	Catalyst 6513 switch—13 slots

For information on installing the for installing and connecting Optical Services Modules (OSMs) in Cisco 7600 series series routers and Catalyst 6500 series switches, see the *Optical Services Module Installation and Verification Note* at http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/hardware/78_11239.htm.

Optical Services Modules

Table 2 lists the standard OSMs and Table 3 list the enhanced OSMs that are covered in this publication. For additional information on these modules, see the *Cisco 7600 Series Router Module Installation Guide* at http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/hardware/osmodule/index.htm.

Table 2 Standard Optical Services Modules

Module	Description
OSM-2OC12-POS-MM, -SI, -SL	2-port OC-12 POS[1], plus 4 Gigabit Ethernet ports (requires GBICs[2]). The module has SC fiber connectors for use with MMF[3] and SMF[4].
OSM-4OC12-POS-MM, -SI, -SL	4-port OC-12 POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has SC fiber connectors for use with MMF and SMF.
OSM-4OC3-POS-SI	4-port OC-3 POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has MT-RJ connectors for use with MMF and SMF.
OSM-8OC3-POS-SI, -SL	8-port OC-3 POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has MT-RJ connectors for use with MMF and SMF.
OSM-16OC3-POS-SI, -SL	16-port OC-3 POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has MT-RJ connectors for use with MMF and SMF.

Table 2 Standard Optical Services Modules (continued)

Module	Description
OSM-1OC48-POS-SS, -SI, -SL	1-port OC-48 POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has SC fiber connectors for use with SMF.
OSM-2OC48/1DPT-SS, -SI, -SL	2-port OC-48 DPT[5]/POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has LC fiber connectors for use with SMF.
OSM-2OC12-ATM-MM, SI	2-port OC-12 ATM[6], plus 4 Gigabit Ethernet ports (require GBICs). The module has SC fiber connectors for use with MMF and SMF.
OSM-4GE-WAN-GBIC	4-port Gigabit Ethernet (requires GBICs).

1. POS = Packet over SONET.
2. GBIC = Gigabit Interface Converters; GBICs are available in three styles (SX, LX/LH, and ZX) and have an SC connector for use with either MMF or SMF.
3. MMF = multimode fiber.
4. SMF = single-mode fiber.
5. DPT = Dynamic Packet Transport.
6. ATM = Asynchronous Transfer Mode.

Table 3 Enhanced Optical Services Modules

Module	Description
OSM-2OC12-POS-MM+, -SI+	2-port OC-12 POS[1], plus 4 Gigabit Ethernet ports (requires GBICs[2]). The module has SC fiber connectors for use with MMF[3] and SMF[4].
OSM-4OC12-POS-SI+	4-port OC-12 POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has SC fiber connectors for use with SMF.
OSM-4OC3-POS-SI+	4-port OC-3 POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has MT-RJ connectors for use with SMF.
OSM-8OC3-POS-SI+, -SL+	8-port OC-3 POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has MT-RJ connectors for use with SMF.
OSM-16OC3-POS-SI+	16-port OC-3 POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has MT-RJ connectors for use with MF.
OSM-1OC48-POS-SS+, -SI+, -SL+	1-port OC-48 POS, plus 4 Gigabit Ethernet ports (requires GBICs). The module has SC fiber connectors for use with SMF.
OSM-1CHOC12/T3-SI[5]	1-port channelized OC-12, plus 4 Gigabit Ethernet ports (requires GBICs). The module has LC fiber connectors for use with SMF.
OSM-12CT3/DS0[5]	12-port channelized T3. The module has mini-SMB connectors for use with 75-Ohm copper coax cable.
OSM-1CHOC12/T1-SI[5]	1-port channelized OC-12, plus 4 Gigabit Ethernet ports (requires GBICs). The module has LC fiber connectors for use with SMF.
OSM-2OC12-ATM-MM+, SI+	2-port OC-12 ATM[6], plus 4 Gigabit Ethernet ports (require GBICs). The module has SC fiber connectors for use with MMF and SMF.
OSM-2+4GE-WAN+	2-port Layer 2 Gigabit Ethernet LAN and 4-port Layer 3 Gigabit Ethernet WAN (all ports require GBICs).

1. POS = Packet over SONET.

2. GBIC = Gigabit Interface Converters; GBICs are available in three styles (SX, LX/LH, and ZX) and have an SC connector for use with either MMF or SMF.
3. MMF = multimode fiber.
4. SMF = single-mode fiber.
5. The channelized OSMs are supported only on the Cisco 7600 series router platform.
6. ATM = Asynchronous Transfer Mode.

Hardware Features

Refer to the *OSM Installation and Verification Note* for a description of the hardware features supported on the OSMs:

http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/hardware/78_11239.htm

Software Features

The following software features are supported on the OSMs:

Note Features in the Cisco IOS 12.2SX releases that are also supported in the Cisco IOS 12.2 mainline, 12.2T and 12.2S releases are documented in the corresponding publications for those releases. When applicable, this section refers to those publications for platform-independent features supported in the Cisco IOS 12.2SX releases. The Cisco IOS 12.2S releases do not support software images for the Cisco 7600 series routers, and the Cisco IOS 12.2S publications do not list support for the Cisco 7600 series routers.

Layer 2 Software Features

The Gigabit Ethernet ports on the OSMs are configured from the supervisor engine of the Catalyst 6500 series switch or the Cisco 7600 series router.

For feature support and configuration information for the OSM Layer 2 Gigabit Ethernet ports, refer to these publications:

Catalyst 6500 Series Cisco IOS Software Configuration Guide, 12.2 SX and the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2 SX* at these URLs:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/index.htm

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm

Cisco 7600 Series Cisco IOS Software Configuration Guide, 12.2 SX and the *Cisco 7600 Series Cisco IOS Command Reference, 12.2 SX* at these URLs:

http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/swcg/index.htm

http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/cmdref/index.htm

Encapsulation Features

The following encapsulation features are supported on the OSM WAN ports:

- High-Level Data Link Control (HDLC) protocol
- Point-to-Point Protocol (PPP)
- PPP over SONET/SDH (RFC 2615)
- PPP in HDLC-like framing (RFC 1662)
- SONET 1+1 Automatic Protection Switching
- SDH 1+1 Multiplex Section Protection (MSP)

  Configure the serial interface encapsulation as described in the *Cisco IOS Interface Configuration Guide* under "Configuring Serial Interfaces" and in the *Cisco IOS Interface Command Reference* publication at these URLs:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/inter_c/index.htm

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/inter_r/index.htm

- Frame Relay

  Configure Frame Relay as described in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.1 under "Configuring Frame Relay" and in the *Cisco IOS Wide-Area Networking Command Reference*, Release 12.1 at these URLs:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_c/wcdfrely.htm

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_r/wrdfrely.htm

  Configure traffic shaping for Frame Relay as described in the *Cisco IOS Quality of Service Solutions Configuration Guide* under "Configuring Distributed Traffic Shaping" at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt4/qcfdts.htm

- Multilink Frame Relay (FRF.16)

  Configure FFR.16 as described at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s14/fs_mfr.htm.

  Note FRF.16 not supported on CHOC12-T3 OSM.

  - The following restrictions apply to FFR.16 with the Channelized OSMs:
  - There is a maximum 168 bundles with two T1/E1 links.
  - There is a maximum 12 links in the bundle.
  - For the OSM-12CT3/T1and OSM-2CHOC12/T1, all the links must be of T1 bandwidth or E1 bandwidth.
  - There is a maximum of 1024 channels (including the multilink frame relay [MFR] bundle).
  - Using Cisco Discovery Protocol (CDP) on MFR interfaces is not recommended because of excessive cpu usage if a large number of sub-interfaces are configured.

Network Management Application Software

The following network management application software is supported on the OSMs:

- CiscoWorks2000

  Installation and administration information for CiscoWorks2000 is available at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cw2000/index.htm

- CiscoView

  Installation and administration information for CiscoView is available in the *Using CiscoView 5.1* publication at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cw2000/cw2000_d/2steditn/use_view/index.htm

- AtmDirector

  For information on using AtmDirector, refer to the *Using AtmDirector* publication at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cw2000/camp_mgr/cwsi_2x/cwsi_2_2/atmd_c/index.htm

- VlanDirector

  For information on using VlanDirector, refer to the *Using VlanDirector* publication at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cw2000/camp_mgr/cwsi_2x/cwsi_2_2/vd_c/index.htm

- Cisco command-line interface (CLI) support
- SNMP support

  Information on CLI and SNMP support is found in the *Cisco IOS Configuration Fundamentals Configuration Guide* and the *Cisco IOS Configuration Fundamentals Command Reference* publication at these URLs:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_c/index.htm

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/index.htm

Traffic Management Features

The OSMs support the following traffic management features:

- Common Open Policy Service (COPS)

  Configure COPS as described in the *COPS for RSVP* Feature Module at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t1/copsrsvp.htm

- Resource Reservation Protocol (RSVP)

  Configure RSVP as described in *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.1 at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt5/qcdrsvp.htm

- Differentiated Services Control Point (DSCP)
- IP precedence with ToS reclassification

- Classification and priority marking based on the following:
  - Ethertype
  - IP source address (SA)
  - IP destintation address (DA)
  - TCP port number
  - UDP port number
  - IP SA + TCP/UDP port number + IP DA + TCP/UDP port number
- Destination Sensitive Services (DSS)

Quality of Service

If your Catalyst 6500 series switch or Cisco 7600 series router is running Cisco IOS software on the MSFC2 and Catalyst software on the supervisor engine, QoS is configured using the Modular QoS Command Line Interface (MQC) and the Catalyst 6500 supervisor engine CLI commands. If you are running Cisco IOS software only, QoS is configured using existing Modular QoS Command Line Interface (MQC).

Refer to the Cisco IOS QoS solutions publications and Catalyst 6500 publications listed below for QoS configuration information.

The OSMs support the following QoS implementations:

- Differentiated services code point (DSCP) and IP precedence classification
- Class-based traffic shaping
- Class-based weighted fair queuing (CBWFQ)

  For a list of the modules that support CBWFQ, see Chapter 9, “Configuring QoS on the Optical Services Modules.”
- Low latency queuing (LLQ)
- Weighted Random Early Detection (WRED)

  For a list of the modules that support WRED, see Chapter 9, “Configuring QoS on the Optical Services Modules.”
- Hierarchical Shaping (supported on Frame Relay, ARPA, dot1q, HLDC, and PPP encapsulations.)

For QoS configuration information and examples, see Chapter 9, “Configuring QoS on the Optical Services Modules.”

For general information on how to configure QoS, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide* and the *Cisco IOS Quality of Service Solutions Command Reference* publication at these URLs:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/index.htm

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/index.htm

For general information on how to configure QoS features on Catalyst 6500 systems running Cisco IOS software on the MSFC2 and Catalyst software on the supervisor engine, refer to the *Catalyst 6500 Seires Software Configuration Guide* and *Catalyst 6500 Series Command Reference* publication at these URLs:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_6_2/confg_gd/index.htm

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_6_2/cmd_ref/index.htm

For general information on how to configure QoS features on the Cisco 7600 series router running Catalyst operating software on the Supervisor Engine 2 and Cisco IOS software on the MSFC2, refer to the *Cisco 7600 Optical Services Router Software Configuration Guide* and the *Cisco 7600 Optical Services Router Command Reference* publications at these URLs:

http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/rel_6_2/swcg/index.htm

http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/rel_6_2/cmdref/index.htm

For general information on how to configure QoS features on Catalyst 6500 systems running Cisco IOS software on the supervisor engine and the MSFC, refer to the *Catalyst 6500 Series Software Configuration Guide* and the *Catalyst 6500 Series Command Reference* publications at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/12_1e/index.htm

For general information on how to configure QoS features on the Cisco 7600 Supervisor Engine 2 running Cisco IOS software, refer to the *Cisco 7600 Series Router Software Configuration Guide* and the *Cisco 7600 Series Router Command Reference* publications at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/ios121_8/index.htm

Destination Sensitive Services

Destination Sensitive Services (DSS) allow traffic accounting and traffic shaping to known autonomous system numbers in order to better engineer and plan network circuit peering and transit agreements. DSS is supported on ingress and egress POS ports on the OC-3, OC-12, and OC-48 POS OSMs and on the GE-WAN ports on the four-port Gigabit Ethernet WAN (GBIC) OSMs.

DSS consists of these two components:

- Destination Sensitive Billing (DSB)

  DSB allows accounting based on destination traffic indexes and provides a means of classifying customer traffic according to the route that the traffic travels. Trans-Pacific, Trans-Atlantic, satellite, domestic, and other provider traffic can be identified and accounted for on a destination network basis when the customer traffic is on a unique software interface. DSB provides packet and byte counters, which represent counts for IP packets per destination network. DSB is implemented using route-maps to classify the traffic into one of seven possible indexes, which represent a traffic classification.

- Destination Sensitive Traffic Shaping (DSTS)

  DSTS performs inbound and outbound traffic shaping based on the destination traffic index configuration. DSTS is supported with ingress DSS only.

  See Chapter 10, “Configuring Destination Sensitive Services on the Optical Services Modules” for configuration information.

Multiprotocol Label Switching

MPLS is supported on all Catalyst 6500 and Cisco 7600 series modules.

For information about platform-specific limitations and restrictions, and supported features, see Chapter 11, “Configuring Multiprotocol Label Switching on the Optical Services Modules.”

For information on MPLS and how to configure it on the OSMs, refer to the Multiprotocol Label Switching on Cisco Routers Feature Module at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/mpls4t.htm.

For general information on MPLS, refer to *Multiprotocol Label Switching* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt3/index.htm

Ethernet over Multiprotocol Label Switching

Ethernet over Multiprotocol Label Switching (EoMPLS) is supported on all all Catalyst 6500 and Cisco 7600 series modules and the FlexWAN modules. You can configure EoMPLS using PFC3BXL-based systems or OSM-based systems.

EoMPLS allows you to connect two VLAN networks that are in different locations without using bridges, routers, or switches at the VLAN locations. You can enable the MPLS backbone network to accept Layer 2 VLAN traffic by configuring the label edge routers (LERs) at both ends of the MPLS backbone.

For information about EoMPLS and how to configure it, see Chapter 11, “Configuring Multiprotocol Label Switching on the Optical Services Modules.”

CHAPTER 1

Basic Configurations

This chapter describes basic configuration information for the Optical Services Modules (OSMs).

This chapter consists of these sections:

For detailed configuration and configuration information for platform-specific features supported on the different OSMs, see the individual chapters.

For information on configuring the Packet over SONET (POS) OSMs, see Chapter 3, "Configuring the OC-3c/STM-1, OC-12c/STM-4, and OC-48c/STM-16 SONET/SDH Optical Services Modules."

For information on configuring the 4-port Gigabit Ethernet WAN modules, see Chapter 4, "Configuring 4-Port Gigabit Ethernet WAN Optical Services Modules."

For information on configuring the channelized OC-12 and OC-48 modules, see Chapter 5, "Configuring the Channelized OC-12/T3 SONET/SDH Optical Services Modules."

For information on configuring the OC-12 ATM modules, see Chapter 8, "Configuring the OC-12 ATM Optical Services Modules."

For information on configuring quality of service and traffic shaping on the OSMs, see Chapter 9, "Configuring QoS on the Optical Services Modules."

For information on configuring Destination Sensitive Services, see Chapter 10, "Configuring Destination Sensitive Services on the Optical Services Modules."

For information on configuring MPLS and EoMPLS, see Chapter 11, "Configuring Multiprotocol Label Switching on the Optical Services Modules."

Configuring the OSMs

This section describes how to perform a basic configuration: enabling an interface (with the **no shutdown** command) and specifying IP routing.

You might also need to enter other configuration subcommands, depending on the requirements for your system configuration and the protocols you plan to route on the interface.

In the following procedure, press the **Return** key after each step unless otherwise noted. At any time you can exit the privileged level and return to the user level by entering **disable** at the prompt as follows:

Router# **disable**

Router>

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# interface pos 7/1`	Specifies the new interface to configure.
Step 3	`Router(config-if)# ip address 10.0.0.10 255.255.255.255`	Assigns an IP address and subnet mask to the interface (if IP routing is enabled on the system), as in this example.
Step 4	`Router(config-if)# no shutdown`	Changes the shutdown state to up and enables the interface. The **no shutdown** command passes an enable command to the interface and causes the OSM to configure itself based on the previous configuration commands sent.
Step 5	`Router# copy running-config startup-config`	Writes the new configuration to NVRAM.

Customizing the Configuration

You can change the default values of all configuration parameters to match your network environment. Use the interface subcommands in the following sections if you need to customize the OSM configuration.

Note The interface subcommands in this section function the same regardless of the platform in which your OSM is installed; however, all of these commands require that you first enter the **interface pos** command to select the interface that you want to configure.

Setting the MTU Size

The default maximum transmission unit (MTU) size is 4470 bytes. To set the MTU size, enter the **mtu** *bytes* command, where *bytes* is a value in the range of 64 through 9216.

```
Router(config)# interface pos 7/1
Router(config-if)# mtu 3000
```

To restore the default of 4470 bytes, enter the **no mtu** command.

Configuring Framing

Note The channelized OC-12 and OC-48 modules do not support SDH framing.

The default framing setting is SONET STS-3c. To configure for SDH STM-1, enter the **pos framing-sdh** command:

```
Router(config)# interface pos 7/1
Router(config-if)# pos framing-sdh
```

To change back to SONET STS-3c, enter the **no pos framing-sdh** command.

Setting the Source of the Transmit Clock

The clocking default specifies that the OSM uses the recovered receive (Rx) clock to provide transmit (Tx) clocking (called loop timing). To specify the OSM to generate the transmit clock internally, enter the **clock source internal** command:

```
Router(config)# interface pos 7/1
Router(config-if)# clock source internal
```

To restore loop timing, enter the **no clock source internal** command or the clock source line command.

Configuring Cyclic Redundancy Checks

The cyclic redundancy check (CRC) default is for a 16-bit CRC (CRC-CITT). The CRC is an error-checking technique that uses a calculated numeric value to detect errors in transmitted data. The OSM also supports a 32-bit CRC. The sender of a data frame calculates the frame check sequence (FCS). The sender appends the FCS value to outgoing messages. The receiver recalculates the FCS and compares it to the FCS from the sender. If a difference exists, the receiver assumes that a transmission error occurred and sends a request to the sender to resend the frame.

To configure an interface for a 32-bit CRC, enter the **crc 32** command:

```
Router(config)# interface pos 7/1
Router(config-if)# crc 32
```

To disable the 32-bit CRC and return the interface to the default 16-bit CRC, enter the **no crc 32** command.

Configuring SONET Payload Scrambling

The default is that SONET payload scrambling is disabled. SONET payload scrambling applies a self-synchronous scrambler (x^43+1) to the SPE of the WAN interface to ensure sufficient bit-transition density.

Note Both ends of the connection must use the same scrambling algorithm.

You enable SONET payload scrambling using the **pos scramble-atm** command. (This command has no keywords or arguments.)

To enable SONET payload scrambling, use the following command sequence:

```
Router(config)# interface pos 7/1
Router(config-if)# pos scramble-atm
Router(config-if)# no shutdown
Router(config-if)# end
```

To verify that SONET payload scrambling is enabled on an interface, enter the **show startup-config** command. If scrambling is enabled, the following line is displayed in the configuration:

```
pos scramble-atm
```

To disable SONET payload scrambling, enter the **no pos scramble-atm** command.

CHAPTER 1

Configuring the OC-3c/STM-1, OC-12c/STM-4, and OC-48c/STM-16 SONET/SDH Optical Services Modules

This chapter describes the OC-3c/STM-1, OC-12c/STM-4, and OC-48c/STM-16 Packet over Synchronous Optical Network (SONET) (POS)/synchronous digital hierarchy (SDH) Optical Services Modules (OSMs).

This chapter consists of these sections:

Supported Features

These sections list the standard Cisco IOS POS and SDH features supported on the OC-3c/STM-1, OC-12c/STM-4, and OC-48c/STM-16 POS/SDH OSMs:

SONET/SDH Compliance

This section lists the SONET/SDH Compliance features:

- Bellcore GR-253-CORE
- ITU-T G.707, G.783, G.957, G.958
- 1+1 SONET Automatic Protection Switching (APS) as per G.783 Annex A
- 1+1 SDH Multiplex Section Protection (MSP) as per G.783 Annex A
- APS Reflector Mode

SONET/SDH Error, Alarm, and Performance Monitoring

This section lists supported SONET/SDH error, alarms, and performance monitoring:

- Signal failure bit error rate (SF-ber)
- Signal degrade bit error rate (SD-ber)
- Signal label payload construction (C2)
- Path trace byte (J1)
- Section:
  - Loss of signal (LOS)
  - Loss of frame (LOF)
  - Error counts for B1
  - Threshold crossing alarms (TCA) for B1
- Line:
  - Line alarm indication signal (LAIS)
  - Line remote defect indication (LRDI)
  - Line remote error indication (LREI)
  - Error counts for B2
  - Threshold crossing alarms (TCA) for B2
- Path:
  - Path alarm indication signal (PAIS)
  - Path remote defect indication (PRDI)
  - Path remote error indication (PREI)
  - Error counts for B3
  - Threshold crossing alarms (TCA) for B3
  - Loss of pointer (LOP)
  - New pointer events (NEWPTR)
  - Positive stuffing event (PSE)
  - Negative stuffing event (NSE)

SONET/SDH Synchronization

This section lists supported SONET/SDH synchronization:

- Local (internal) timing (for inter-router connections over dark fiber or WDM equipment)
- Loop (line) timing (for connecting to SONET/SDH equipment)
- +/- 20 ppm clock accuracy over full operating temperature

WAN Protocols

This section lists the supported WAN protocols:

- IETF RFC 1661, Point-to-Point Protocol (PPP)
- IETF RFC 1662, PPP in HDLC framing
- IETF RFC 2615, PPP over SONET/SDH with $1+x^{43}$ self-synchronous payload scrambling
- Cisco Protect Group Protocol over UDP/IP (Port 172) for APS and MSP
- Multiprotocol Label Switching (MPLS)

Note The 2-port OC-48c/STM-16 POS/DPT OSMs does support MPLS but does not support EoMPLS.

- Ethernet over Multiprotocol Label Switching (EoMPLS)
- Frame Relay

  Configure the POS interfaces for Frame Relay as described in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.1 under "Configuring Frame Relay" and in the *Cisco IOS Wide-Area Networking Command Reference*, Release 12.1 at these URLs:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_c/wcdfrely.htm

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_r/wrdfrely.htm

  Configure traffic shaping for Frame Relay as described in the *Cisco IOS Quality of Service Solutions Configuration Guide* under "Configuring Distributed Traffic Shaping" at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt4/qcfdts.htm

  See the "Configuring Frame Relay and Frame Relay Traffic Shaping" section on page 3-17 for information about platform-specific configurations, commands, and limitations.

Note The 2-port OC-48c/STM-16 POS/DPT OSMs do not support Frame Relay.

Dynamic Packet Transport Protocol

The 2-port OC-48c/STM-16 POS/DPT OSMs (OSM-2OC48/1DPT) support these Dynamic Packet Transport (DPT) protocol features:

- DPT Spatial Reuse Protocol (SRP) MAC
- DPT SRP fairness algorithm (SRP-fa)
- DPT SRP intelligent protection switching (IPS)
- SRR (single ring recovery)

Bridging Control Protocol

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

The following OSMs support BCP as defined in RFC 3518:

- OC-3 POS:
  - OSM-4OC3-POS-SI
  - OSM-4OC3-POS-SI+
  - OSM-8OC3-POS-SI, -SL
  - OSM-8OC3-POS-SI+, -SL+
  - OSM-16OC3-POS-SI, -SL
  - OSM-16OC3-POS-SI+
- OC-12 POS:
  - OSM-2OC12-POS-MM, -SI, -SL
  - OSM-2OC12-POS-MM+, -SI+
  - OSM-4OC12-POS-MM, -SI, -SL
  - OSM-4OC12-POS-SI+
- OC-48 POS:
  - OSM-1OC48-POS-SS, -SI, -SL
  - OSM-1OC48-POS-SS+, -SI+, -SL+
  - OSM-2OC48-POS/DPT-SS, -SI, -SL

Note For interoperability purposes, keep in mind that OSM POS interfaces with BCP configured can forward both Layer 2 and Layer 3 traffic at the same time, while POS interfaces on other Cisco platforms support only Layer 2 forwarding when BCP is enabled.

Figure 3-1 shows a topology where BCP is used to allow transparent forwarding of VLAN traffic over a SONET network.

Figure 1-1 BCP Topology in a SONET Network

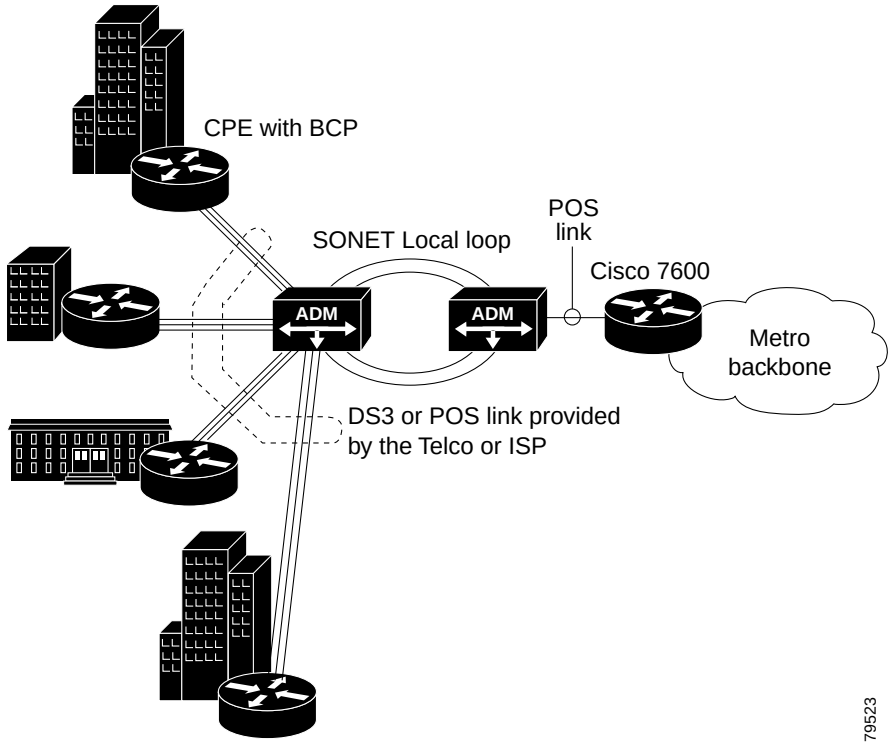
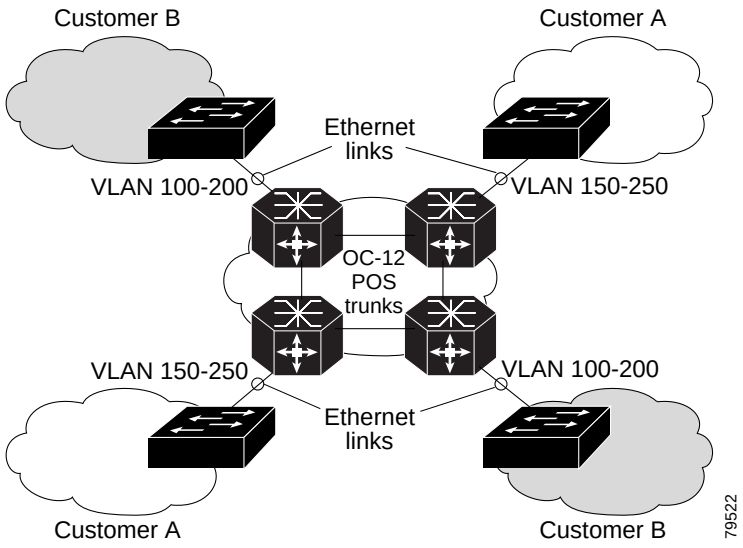


Figure 3-2 shows a topology where VLAN IDs are used to create VPNs for different customers and BCP is used to forward the VPN traffic over a SONET network.

Figure 1-2 BCP Topology in a VPN Network



For information on configuring BCP, see the "Configuring Bridging Control Protocol" section on page 3-22.

Quality of Service Support with BCP

Quality of Service (QoS) is supported on BCP links using the three experimental bits in a label to determine the priority of packets. To support QoS between LERs, you set the experimental bits in both the VC and tunnel labels. The experimental bits need to be set in the VC label because the tunnel label is popped at the penultimate router.

Routing and Scalability Protocols

This section lists the supported routing and scalability protocols:

- Distributed Cisco Express Forwarding (dCEF)
- WCCP v2
- With the Policy Feature Card 2 (PFC2) only, GRE encapsulated tunneling (supported in software)

Note Generic routing encapsulation (GRE) tunnel IP source and destination VRF membership is not supported with the **tunnel vrf** command.

Network Management

This section lists the supported network management features:

- Local (diagnostic) loopback
- Network loopback
- NetFlow Data Export
- IP over the Data Communications Channel (DCC)

Note The 2-port OC-48c/STM-16 POS/SDH OSMs do not support DCC.

- RFC 1595 performance statistics for timed intervals (current, 15 minute, multiple 15 minute, and 1-day intervals):
  - Regenerator section
  - Multiplex section
  - Path errored seconds
  - Severely errored seconds
  - Severely errored framed seconds

Quality of Service Protocols

This section lists the supported QoS features:

- 2,048 QoS queues per module (32 service classes and 64 DSCP queues/class)
- Class-based traffic shaping
- Differentiated Services Control Point (DSCP) classification

- IP precedence classification
- Class-based weighted fair queuing (CBWFQ)
- Low latency queuing (LLQ)
- Hierarchical traffic shaping for Frame Relay, HDLC, and PPP encapsulations.

Note The OC-48 POS/DPT modules do not support LLQ, CBWFQ, or DSCP classification. Class-based traffic shaping is supported for ingress traffic only.

Security Protocols

This section lists the supported security features:

- Standard and extended access control lists (ACL)
- Named, dynamic, reflexive, and time-based ACLs
- IPv4 NAT (supported in software)

Multiprotocol Label Switching

MPLS is supported on all Catalyst 6500 and Cisco 7600 series modules.

For information about platform-specific limitations and restrictions, and supported features, see Chapter 11, "Configuring Multiprotocol Label Switching on the Optical Services Modules."

For information on MPLS and how to configure it on the OSMs, refer to the Multiprotocol Label Switching on Cisco Routers Feature Module at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/mpls4t.htm.

For general information on MPLS, refer to *Multiprotocol Label Switching* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt3/index.htm

Understanding Packet-Over-SONET

Packet-over-SONET is a high-speed method of transporting IP traffic between two points. This technology combines the Point-to-Point Protocol (PPP) with SONET and Synchronous Digital Hierarchy (SDH) interfaces.

SONET is an octet-synchronous multiplex scheme defined by the American National Standards Institute (ANSI) standard (T1.1051988) for optical digital transmission at hierarchical rates from 51.840 Mbps to 2.5 Gbps (Synchronous Transport Signal, STS-1 to STS-48) and greater. SDH is an equivalent international standard for optical digital transmission at hierarchical rates from 155.520 Mbps (STM-1) to 2.5 gigabits per second (Gbps) (STM-16) and greater. SONET electrical specifications have been defined for single-mode fiber, multimode fiber, and CATV 75-ohm coaxial cable. The OC-3c/STM-1, OC-12c/STM-4, and OC-48c/STM-16 POS/SDH OSMs allow transmission over single-mode and multimode optical fiber at Optical Carrier 3, 12, and 48 (OC-3, OC-12, and OC-48) rates.

SONET/SDH transmission rates are integral multiples of 51.840 Mbps. The following transmission multiples are currently specified and commonly used:

- OC-3c/STM-1c—155.520 Mbps

- OC-12c/STM-4c—622.080 Mbps
- OC-48c/STM-16c—2488.320 Mbps

The POS specification (RFC 1619) describes the use of PPP encapsulation over SONET/SDH links. Because SONET/SDH is, by definition, a point-to-point circuit, PPP is well-suited for use over these links. PPP treats SONET/SDH transport as octet-oriented full-duplex synchronous links. PPP presents an octet interface to the physical layer. The octet stream is mapped into the SONET/SDH Synchronous Payload Envelope (SPE), with the octet boundaries aligned with the SPE octet boundaries. The PPP frames are located by row within the SPE payload. Because frames are variable in length, the frames are allowed to cross SPE boundaries.

The basic rate for POS is OC-3/STM-1, which is 155.520 Mbps. The available information bandwidth is 149.760 Mbps, which is the OC-3c/STM-1 SPE with section, line, and path overhead removed.

SONET Distance Limitations

The specification for optical fiber transmission defines two types of fiber: single-mode and multimode. Within the single-mode category, three transmission types are defined: short reach, intermediate reach, and long reach. Within the multimode category, only short reach is available.

For information on cable distance limitations and power budget, see http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/hardware/osmodule/02prep.htm.

Configuring the Interfaces

This section describes how to configure the OC-3c/STM-1, OC-12c/STM-4, and OC-48c/STM-16 OSMs:

Initial Configuration of the POS/SDH OSMs

If you installed a new POS/SDH OSM or want to change the configuration of an existing interface, you must enter configuration mode by using the **configure** command in the privileged EXEC mode. Table 3-1 shows the default configuration of an enabled module. For more information, see the "Customizing the POS/SDH OSM Configuration" section on page 3-10.

Table 1-1 POS/SDH Module Configuration Default Values

Parameter	Configuration Command	Default Value
Keepalive	[**no**] **keepalive**	keepalive
Encapsulation	**encapsulation** [**hdlc** | **ppp** | **frame-relay**]	hdlc
Cisco Discovery Protocol (cdp)	[**no**] **cdp enable**	cdp enable
Maximum transmission unit (mtu)	[**no**] **mtu** *bytes*	4470 bytes
Framing	**pos framing** [**sdh** | **sonet**]	SONET OC-3c; OC-12c; OC-48c
Bandwidth	[**no**] **bandwidth** *kilobits*	155000; 622000; 2500000
SONET overhead	**pos flag** [**c2** *value* | **j0** *value* | **s1s0** *value* | **s1 ignore**]	c2 set to 0xcf; j0 set to 0xcc; s1s0 set to 0; s1 set to ignore the received s1 byte setting.
Loop internal	[**no**] **loop** [**internal** | **line**]	no loopback
POS SPE scrambling	[**no**] **pos scramble-atm**	no POS SPE scramble
Cyclic Redundancy Check	**crc** [**16** | **32**]	32
Clock source	**clock source** [**internal** | **line**]	line

Configuring the Interface

After you verify that the new POS/SDH OSM is installed correctly, use the **configure** command in the privileged EXEC mode to configure the new interface. Be prepared with the information you will need, such as the interface IP address.

The following procedure is for creating a basic configuration, which includes enabling an interface and specifying IP routing.

A Catalyst 6500 series switch and Cisco 7600 series router identifies an interface address by its module slot number and port number in the format *slot/port*. For example, the slot/port address of an interface on a 1-port OC-48c/STM-16 POS/SDH OSM installed in slot 4 is *4/1*. Even though the card contains only one port, you must use the *slot/port* notation.

Before using the **configure** command, you must enter the privileged level mode of the EXEC command interpreter by using the **enable** command. The system will prompt you for a password if one is set.

To configure the POS/SDH OSMs (press the **Return** key after each configuration step unless otherwise noted), perform this task:

	Command	Purpose
Step 1	`Router# show version`	Confirms that the system recognizes the module by entering the **show version** command.
Step 2	`Router# show interface`	Checks the status of each port by entering the **show interface** command.
Step 3	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal will be the source of the configuration subcommands.
Step 4	`Router(config)# ip routing`	Enables IP routing by entering the **ip routing** command.
Step 5	`Router(config)# interface pos` *slot/port*	Specifies the new interface to configure by entering the **interface** command, followed by *type* and *slot/port*.
Step 6	`Router(config-if)# ip address` *ip-address mask* [**secondary**]	Assigns an IP address and subnet mask to the interface.
Step 7	`Router(config-if)# encapsulation` *encapsulation-type*	Verifies that HDLC encapsulation is correct for this interface; *encapsulation-type* is one of the keywords, **hdlc** or **ppp** or **frame-relay.**
Step 8	`Router(config-if)# clock source` {**line** | **internal**}	Verifies that the default value for the clock source is correct. The default value is *line*. Use it when clocking is derived from the network. The **clock source internal** command is typically used when two Cisco 7600 series routers or Catalyst 6500 series switches are connected back-to-back or are connected over dark fiber where no clocking is available. In either case, each device should have its clock source set to internal.
Step 9	`Router(config-if)# no shutdown`	Changes the interface state to up and enables the interface.
Step 10	`Router(config-if)# keepalive`	Turns on or off keepalive messages as desired. Keepalive messages are useful for encapsulated protocols such as HDLC. The keepalive default is on.
Step 11	`Router# copy running-config startup-config`	Writes the new configuration to memory.

Customizing the POS/SDH OSM Configuration

This section documents new platform-specific commands. Other commands used in OSM configuration are documented in the Cisco IOS Release 12.1 command reference publications.

You can change the default values of all POS/SDH OSM configuration parameters to match your network environment. Perform the tasks in the following sections if you need to customize the POS/SDH OSM configuration:

- Selecting a POS/SDH OSM Interface, page 3-11
- Configuring Framing, page 3-11
- Specifying SONET Overhead, page 3-11
- Configuring POS SPE Scrambling, page 3-11

Selecting a POS/SDH OSM Interface

An OC-3c/STM-1, OC-12c/STM-4, or OC-48c/STM-16 interface is referred to as **pos**, for packet-over-SONET, in the configuration commands. To select a specific POS interface, use the **interface pos** *slot/port* command in the configuration mode:

```
Router(config)# interface pos slot/port
```

Configuring Framing

The **pos framing** command allows you to set framing to SONET OC or SDH STM. The default is SONET.

```
Router(config-if)# pos framing [sdh|sonet]
```

Specifying SONET Overhead

The **pos flag** command allows you to specify values for the specific elements of the frame header.

```
Router(config-if)# pos flag [c2 value] [j0 value] [s1s0 value]
```

where

- **c2** is a path signal identifier, and *value* is one of the following:
  - 0xCF = PPP or HDLC (default)
  - 0x13 = ATM
- **j0** is the section trace byte, and *value* is 0x1 for interoperability with some SDH devices in Japan. The default value is 0xCC.
- **s1s0** is part of the payload pointer byte, and *value* is one of the following:
  - 0 = OC-3c (default)
  - 2 = AU-4

Configuring POS SPE Scrambling

The POS scrambling command allows you to scramble the POS SPE (synchronous payload envelope) payload. The default is no POS SPE scramble.

```
Router(config-if)#[no] pos scramble-atm
```

Using show Commands to Check System Status

Each OSM maintains information about its configuration, traffic, and errors. You can access this information by using the **show** commands.

Descriptions and examples of module and system status **show** commands follow:

- Use the **show interfaces** command and the **show interfaces pos** *slot/port* command to display information about the system interfaces. The following example illustrates the **show interface pos** *slot/port* command for port 1 of a module installed in slot 5:

```
Router# show interfaces pos 5/1
POS5/1 is administratively down, line protocol is down
  Hardware is Packet over SONET
  MTU 4470 bytes, BW 155000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation HDLC, crc 32, loopback not set, keepalive set (10 sec)
  Scramble disabled
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  queuing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
              0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 applique, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
...
(output truncated)
```

- Use the **show version** command to display the configuration of the system hardware (the number of each module type installed), the Cisco IOS software version, the names and sources of configuration files, and the boot images. The following example illustrates the **show version** command for a Catalyst 6509 switch:

```
Router# show version
WS-C6509 Software, Version NmpSW: 6.1(2)
Copyright (c) 1995-2001 by Cisco Systems
NMP S/W compiled on Jan 25 2001, 12:28:23
System Bootstrap Version: 6.1(2)
Hardware Version: 2.0 Model: WS-C6509 Serial #: SCA042101NG
Mod     Port    Model               Serial #        Versions
---     ----    ------------------- ----------- -----------------------------------
---
1       2       WS-X6K-SUP2-2GE     SAD044102J9     Hw : 1.1
                                                    Fw : 6.1(2)
                                                    Fw1: 6.1(3)
                                                    Sw : 6.1(2)
                                                    Sw1: 6.1(2)
                WS-F6K-PFC2         SAD04470KPP     Hw : 1.0
3       8       WS-X6408-GBIC       SAD03090264     Hw : 1.4
                                                    Fw : 4.2(0.24)VAI78
                                                    Sw : 6.1(2)
4       8       WS-X6408A-GBIC      SAD043500LE     Hw : 1.3
                                                    Fw : 5.4(2)
                                                    Sw : 6.1(2)
5       4       OSM-4OC12-POS-MM    SAD050202EJ     Hw : 0.101
                                                    Fw : 12.1(6.5)E1
                                                    Sw : 12.1(6.5)E1
```

```
6       24      WS-X6224-100FX-MT   SAD03040765     Hw : 1.2
                                                    Fw : 4.2(0.24)VAI78
                                                    Sw : 6.1(2)
9       48      WS-X6248            SAD03200773     Hw : 1.1
                                                    Fw : 4.2(0.24)VAI78
                                                    Sw : 6.1(2)
15      1       WS-F6K        SAD044803FK     Hw : 1.1
                                                    Fw : 12.1(3a)E4
                                                    Sw : 12.1(3a)E4
         DRAM                      FLASH                   NVRAM
Module   Total     Used    Free    Total   Used    Free    Total  Used   Free
------   -------   ------- ------- ------- ------- ------- -----  -----  -----
1      130944K     57316K  73628K  16384K  6647K   9737K   512K   302K   210K
Uptime is 2 days, 19 hours, 50 minutes
Console> (enable))
```

- Use the **show protocols** command to display the global (system-wide) and interface-specific status of any configured Level 3 protocol.
- Use the **show running-config** command to display the currently running configuration in RAM:

```
Router# show running-config
Building configuration...
Current configuration:
!
version 11.2
no service udp-small-servers
no service tcp-small-servers
!
hostname Maxwell
!
enable secret 5 $1$ZBC0$tJO8EeP3VI769LAw.3edJ1
enable password xxxx
!
ip host ray 172.27.136.253
ip host crusty 171.69.209.28
ip domain-name cisco.com
ip name-server 171.69.209.10
clock timezone EST -5
clock summer-time EDT recurring
!
interface POS0/0
 no ip address
 shutdown
 crc 32
!
interface POS0/1
 no ip address
 shutdown
 crc 32
!
(output truncated)
```

Configuring Automatic Protection Switching

Automatic protection switching (APS) allows switchover of packet-over-SONET (POS) circuits and is often required when connecting SONET equipment to telecommunications equipment. When APS is configured, a protect POS interface is brought into the SONET network from the intervening SONET equipment and the protect POS interface becomes the working POS interface on the circuit.

The protect interface is configured with the IP address of the router that has the working interface. The APS Protect Group Protocol provides communication between the process controlling the working interface and the process controlling the protect interface. When you use the APS Protect Group Protocol, POS interfaces can be switched in the event of a router failure, degradation or loss of channel signal, or manual intervention.

Two SONET connections are required to support APS. In a telecommuncations environment, the SONET circuits must be provisioned as APS. You must also provision the operation, mode, and revert options. If the SONET connections are homed on two separate routers (the normal configuration), an out-of-band (OOB) communications channel between the two routers needs to be set up for APS communication.

When configuring APS, we recommend that you configure the working interface first, along with the IP address of the interface being used as the APS OOB communications path.

Note To prevent the protected interface from becoming the active circuit and disabling the working circuit when it is discovered, configure the working interface before configuring the protected interface.

For more information on APS and configuration information for additional APS features, refer to the *Cisco IOS Interface Configuration Guide*, Release 12.1 at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/inter_c/index.htm

Configuring the Working Interface

To configure the working interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface pos` *slot/port*	Specifies the POS interface to be configured as the working interface and enters interface configuration mode.
Step 2	`Router(config-controller)# aps working` *circuit-number*	Configures this interface as a working interface.
Step 3	`Router(config)# end`	Exits configuration mode.

Step 4	`Router# show controllers pos`
	`Router# show interface pos`
	`Router# show aps`
	`Router# show aps controller`
		Displays information about the POS controllers and interface so that you can verify that the interface is configured correctly.

Note If a router has two or more protect interfaces, the **aps group** command for each interface must precede the corresponding **aps protect** command.

Configuring the Protect Interface

To configure the protect interface, perform this task beginning in global configuration mode:

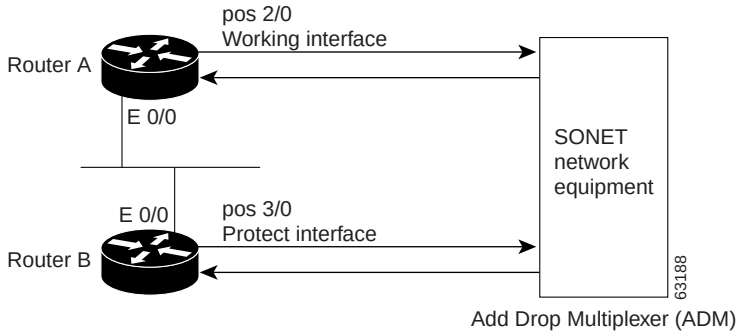
	Command	Purpose
Step 1	`Router(config)# interface pos slot/port`	Specifies the POS interface to be configured as the protect interface and enters interface configuration mode.
Step 2	`Router(config-if)# aps protect circuit-number ip-address`	Configures this interface as a protect interface. Specifies the IP address of the router that contains the working interface.
Step 3	`Router(config-if)# end`	Exits configuration mode.

Step 4	`Router# show controllers pos`
	`Router# show interface pos`
	`Router# how aps`
		Displays information about the POS controllers and interface so that you can verify that the interface is configured correctly.

Configuring Basic APS

The following example shows the configuration of APS on router A and router B (see Figure 3-3). In this example, router A is configured with the working interface, and router B is configured with the protect interface. If the working interface on router A becomes unavailable, the connection will automatically switch over to the protect interface on router B. The working and protect interfaces are configured at the controller level.

Figure 1-3 Basic APS Configuration



Step 1 On router A, which contains the working interface, use the following configuration:

```
Router# configure terminal
Router(config)# interface loopback 1
Router(config-if)# ip address 7.7.7.7 255.255.255.0
Router(config)# exit
Router(config)# interface pos 2/0
Router(config-if)# aps working 1
router(config-if)# pos ais-shut
Router(config-if)# end
Router#
```

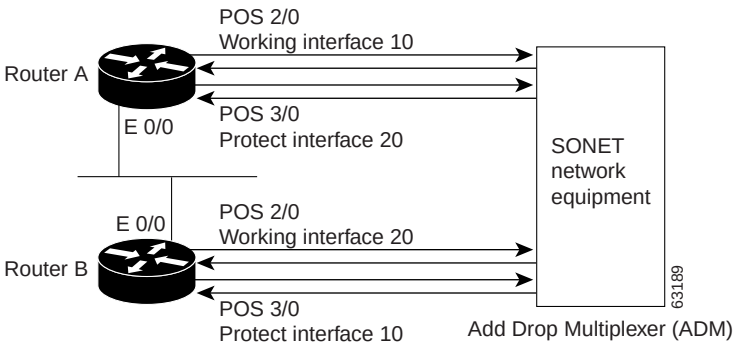
Step 2 On router B, which contains the protect interface, use the following configuration:

```
Router# configure terminal
Router(config)# interface loopback 2
Router(config-if)# ip address 7.7.7.6 255.255.255.0
Router(config)# exit
Router(config-if)# interface pos 3/0
Router(config-if)# aps protect 1 7.7.7.7
router(config-if)# pos ais-shut
Router(config-if)# end
Router#
```

Multiple APS Interface Configuration

To configure more than one protect/working interface, use the **aps group** command. The following example in Figure 3-4 shows the configuration of grouping more than one working/protect interface. In this example, router A is configured with a working interface and a protect interface, and router B is configured with a working interface and a protect interface. If the working interface 2/0 on router A becomes unavailable, the connection will switch over to the protect interface 3/0 on router B because they are both in APS group 10. Similarly, if the working interface 2/0 on router B becomes unavailable, the connection will switch over to the protect interface 3/0 on router A because they are both in APS group 20.

Figure 1-4 Multiple Working and Protect Interfaces Configuration



Note Configure the working interface before configuring the protect interface to avoid the protect interface from becoming the active circuit and disabling the working circuit when it is discovered.

Step 1 On router A, which contains the working interface for group 10 and the protect interface for group 20, use the following configuration:

```
router# configure terminal
router(config)# interface ethernet 0/0
router(config-if)# ip address 7.7.7.6 255.255.255.0
router(config-if)# exit
router(config)# interface POS 2/0
router(config-if)# aps group 10
router(config-if)# aps working 1
router(config-if)# exit
router(config)# interface POS 3/0
router(config-if)# aps group 20
router(config-if)# aps protect 1 7.7.7.7
```

```
router(config-if)# end
router#
```

Step 2 On router B, which contains the protect interface for group 10 and the working interface for group 20, use the following configuration:

```
router# configure terminal
router(config)# interface ethernet 0/0
router(config-if)# ip address 7.7.7.7 255.255.255.0
router(config-if)# exit
router(config)# interface POS 2/0
router(config-if)# aps group 20
router(config-if)# aps working 1
router(config-if)# exit
router(config)# interface POS 3/0
router(config-if)# aps group 10
router(config-if)# aps protect 1 7.7.7.6
router(config-if)# end
router#
```

Configuring Frame Relay and Frame Relay Traffic Shaping

This section describes Frame Relay configurations, platform-specific commands, and limitations:

Configure the interfaces for Frame Relay as described in the *Cisco IOS Wide-Area Networking Configuration Guide,* Release 12.1 under “Configuring Frame Relay” and in the *Cisco IOS Wide-Area Networking Command Reference*, Release 12.1 at these URLs:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_c/wcdfrely.htm

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_r/wrdfrely.htm

Configure traffic shaping for Frame Relay as described in the *Cisco IOS Quality of Service Solutions Configuration Guide* under “Configuring Distributed Traffic Shaping” at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt4/qcfdts.htm.

Frame Relay Limitations and Restrictions

The following limitations and restrictions apply to Frame Relay:

- Frame Relay is not supported on SVCs.
- IP addresses cannot be assigned to main interfaces configured for Frame Relay.
- Frame Relay is supported only on point-to-point connections.
- Frame Relay switching functionality is not supported. The Frame-Relay switching configuration is available only to configure the **frame-relay intf-type dce** option.
- Frame Relay Fragmentation and Compression is not supported.
- Only FIFO queuing is supported.
- DLCI is configurable on subinterfaces only and cannot be configured on the main interface.
- Only class-based traffic shaping is supported. The following commands are not supported:
  - Router(config-pmap-c)# **shape** [**average** | **peak**] *mean-rate* [[*burst-size*] [*excess-burst-size*]]
  - Router(config-pmap-c)# **priority** {*kbps* | **percent** *percent*} [*bytes*]
  - Router(config-pmap-c)# **fair-queue** *number-of-queues*
  - Router(config-map-class)# **frame-relay adaptive-shaping** [**becn** | **foresight**]
  - Router(config-map-class)# **frame-relay cir** {**in** | **out**} *bps*
  - Router(config-map-class)# **frame-relay** {**bc** | **be**} {**in** | **out**} *bits*
  - Router(config-map-class)# **frame-relay traffic-rate average** [**peak**]
  - Router(config-map-class)# **frame-relay priority-group** *list-number*
  - Router(config-map-class)# **frame-relay fragment** *fragment_size*
  - Router(config-if)# **frame-relay payload-compress packet-by-packet**
  - Router(config-if)# **frame-relay de-group** *group-number dlci*
  - Router# **show traffic-shape queue**

Frame Relay Traffic Shaping Configuration Example

To configure frame relay traffic shaping, perform this task:

	Command	Purpose
Step 1	`Router(config-pmap)# class-map [match-all | match-any]`	Creates a class map to be used for matching packets to a class you define and specifies the criteria to match on. Match criteria for classes can be based on IP DSCP or IP precedence.
Step 2	`Router(config-pmap)# match`	Identifies a match criterion.
Step 3	`Router(config)# policy-map policy_map`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
Step 4	`Router(config-pmap)# class class-name`	Defines the classes you want the service policy to contain.
Step 5	`Router(config-pmap-c)# shape average mean-rate [burst-size]`	Shapes traffic to the indicated bit rate.

	Command	Purpose
Step 6	`Router(config)# map-class frame-relay` *map-class-name*	Specifies a map class to define quality of service (QoS) values.
Step 7	`Router(config-map-class)# no frame-relay adaptive-shaping`	Disables backward notification.
Step 8	`Router(config-map-class)# service-policy input` *policy-map*	Attaches the specified policy map to the input interface.
Step 9	`Router(config-map-class)# service-policy output` *policy-map*	Attaches the specified policy map to the output interface.
Step 10	`Router(config)# interface` *interface*	Specifies the interface to which the policy map will be applied.
Step 11	`Router(config-subif)# ip address` *ip_address mask*	Assigns an IP address to the subinterface.
Step 12	`Router(config-subif)# no cdp enable`	Disables CDP.
Step 13	`Router(config-subif)# frame-relay interface-dlci` *dlci*	Assigns a data link connection identifier (DLCI) to a specified Frame Relay subinterface.
Step 14	`Router(config-fr-dlci)# class` *class-name*	Specifies the name a predefined map-class which was defined with the **map-class frame-relay** command.

We recommend that you explicitly disable CDP on the subinterfaces. Should CDP be required on the subinterfaces, the input-queue depth may need to be adjusted. To accommodate the number of incoming CDP packets, configure the input-queue depth on the main interface to be slightly larger than the number of subinterfaces on which you have enabled CDP. The default input-queue depth is 75 and can be adjusted with the **hold-queue** interface command:

```
Router(config-if)# hold-queue 300 in
```

The following example shows a configuration that shapes the traffic for DLCI 18 to be 8 Mbps on both input and output traffic flows:

```
Router(config)# class-map match-all fr-classmap
Router(config-cmap)# match any
Router(config-cmap)# exit
Router(config)# policy-map fr-map
Router(config-pmap)# class fr-classmap
Router(config-pmap-c)# shape average 8000000 32000 32000
Router(config-pmap-c)# exit
Router(config)# map-class frame-relay fr-shaping
Router(config-map-class)# no frame-relay adaptive-shaping
Router(config-map-class)# service-policy input fr-pmap
Router(config-map-class)# service-policy output fr-pmap
Router(config-map-class)# exit
Router(config)# interface POS7/15.1 point-to-point
Router(config-subif)# ip address 72.0.0.1 255.255.0.0
Router(config-subif)# no cdp enable
Router(config-subif)# frame-relay interface-dlci 18
Router(config-fr-dlci)# class fr-shaping
Router(config-fr-dlci)# exit
```

Configuring Dynamic Packet Transport Protocol

Dynamic Packet Transport (DPT) is a packet ring technology that allows you to scale and distribute your Internet and IP services across a reliable optical packet ring infrastructure.

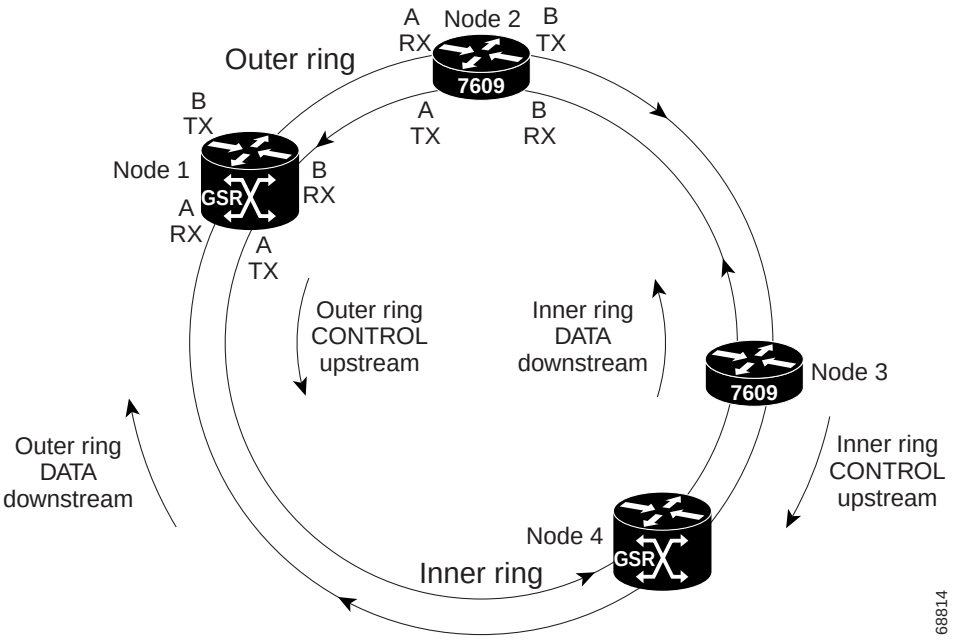
For general overview information for DPT, refer to the *Dynamic Packet Transport Feature Guide* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/srpapsgs.htm

The 2-port OC-48c/STM-16c OSM can be used as a 2-port POS/SDH uplink module or as a single-port DPT module. When the 2-port OC-48c/STM-16c OSM is used as a DPT module, one of the OC-48 interfaces functions as the Side-A interface and the other as the Side-B interface.

Figure 3-5 shows a DPT ring created with two 1-port OC-48c/STM-16c SRP modules installed in the Cisco 12000 series router and one 2-port OC-48c/STM-16c OSM installed in the Cisco 7600 series routers.

Figure 1-5 SRP/DPT Ring Example



To configure DPT on the 2-port OC-48c/STM-16 OSM, perform this task from configuration mode:

	Command	Purpose
Step 1	`Router(config)# hw-module slot 4 srp`	Converts the module to SRP/DPT mode.
Step 2	`Router(config)# interface srp 4/1`	Selects the SRP interface to be configured.
Step 3	`Router(config-if)# ip address 10.1.2.1 255.255.255.0`	Configures the IP address.
Step 4	`Router(config-if)# no cdp enable`	Disables CDP.
Step 5	`Router(config-if)# no shutdown`	Brings up the interface.
Step 6	`Router(config-if)# exit`	Exits interface configuration mode.

	Command	Purpose
Step 7	`Router(config)# exit`	Exits configuration mode.
Step 8	`Router# show interfaces srp 4 /1`	Displays interface configuration.

This example shows how to configure the 2-port OC-48c/STM-16c OSM for SRP/DPT mode.

```
Router(config)# hw-module slot 4 srp
```

Note Wait for the module in slot 4 to be configured to SRP/DPT mode and automatically reloaded. Continue with the configuration.

```
Router(config)# interface srp 4/1
Router(config-if)# ip address 10.1.2.1 255.255.255.0
Router(config-if)# no cdp enable
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# exit
Router# show interfaces srp 4/1
SRP4/1 is up, line protocol is up
  Hardware is SRP, address is 00d0.01d7.4c0a (bia 00d0.01d7.4c0a)
  Internet address is 10.1.2.1/24
  MTU 4470 bytes, BW 2488000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 41/255
  Encapsulation SRP2,
  Side A: loopback not set
  Side B: loopback not set
     3 nodes on the ring   MAC passthrough not set
     Side A: not wrapped   IPS local: IDLE       IPS remote: IDLE
     Side B: not wrapped   IPS local: IDLE       IPS remote: IDLE
  Scramble enabled
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  queuing strategy: fifo
  Output queue :0/40 (size/max)
  Side A: 5 minutes output rate 0 bits/sec, 0 packets/sec
          5 minutes input rate 0 bits/sec, 0 packets/sec
  Side B: 5 minutes output rate 0 bits/sec, 0 packets/sec
          5 minutes input rate 0 bits/sec, 0 packets/sec
  L2 Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes
  L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes mcast
  L3 out Switched: ucast: 0 pkt, 0 bytes
     360563 packets input, 286645033 bytes, 0 no buffer
     Received 0 broadcasts, 43 runts, 0 giants, 0 throttles
     50 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 7 abort
     847443 packets output, 34168034 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 output buffer failures, 0 output buffers swapped out
     Side A received errors:
        33 input errors, 0 CRC, 0 ignored,
        29 framer runts, 0 framer giants, 4 framer aborts,
        0 mac runts, 0 mac giants, 0 mac aborts
     Side B received errors:
        17 input errors, 0 CRC, 0 ignored,
        14 framer runts, 0 framer giants, 3 framer aborts,
        0 mac runts, 0 mac giants, 0 mac aborts
Router#
```

Configuring Bridging Control Protocol

When BCP is used to forward Ethernet frames over SONET no Layer 3 routing information needs to be exchanged, and the POS links function like Ethernet trunks carrying VLAN traffic over the existing reliable high-speed SONET network. BCP is not supported on DPT OSM.

Usage Guidelines and Restrictions

When configuring BCP, observe the following guidelines and restrictions.

- Each PXF complex supports only one instance of a VLAN. As a result, although more than one interface might be supported per PXF complex, the same VLAN cannot be configured on more than one interface per PXF complex. Depending on the particular POS OSM, each interface may share a PXF complex with other interfaces. For example, on a 4-port OC-12 POS OSM, port 1 and 2 share one PXF complex and port 3 and 4 share another PXF complex. If VLAN 400 is configured on port 1, that same VLAN cannot be configured on port 2. But VLAN 400 is allowed on either port 3 or port 4.

  Additionally, if you configure a given VLAN for BCP, then you cannot configure the same VLAN for any other bridging feature on an interface attached to the same PXF complex. This includes Frame Relay bridging as well as VPLS (Virtual Private LAN Service).
- In order for a POS interface to support bridging, the POS interface minimum MTU size should be 24 bytes larger than the VLAN interfaces and Ethernet interfaces MTU size. This accounts for 6 bytes of RFC 3518 header and 18 bytes of 802.1Q header.

  For example, if the MTU size on an ingress Ethernet port is 3000 bytes, the POS port MTU size should be at least 3024 bytes.

Quality of Service Support

OSMs use DSCP-based queuing and shaping, but because BCP does Layer 2 traffic forwarding, there is no DSCP value to look at. Instead, the 3-bit CoS field in the 802.1Q header is mapped to a 6-bit DSCP value.

When BCP is enabled, the CoS value in the 802.1Q header is mapped to the DSCP value in the IP header according to this default CoS to DSCP mapping:

CoS	0	1	2	3	4	5	6	7
DSCP	0	8	16	24	32	40	48	56

For information about QoS on the Layer 3 OSM ports, see Chapter 9, “Configuring QoS on the Optical Services Modules.” For information on PFC2 QoS support, refer to the QoS chapter of the *Cisco 7600 Series Router Software Configuration Guide* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/swcg/qos.htm

To configure BCP, peform this task:

	Command	**Purpose**
Step 1	`Router(config)# interface pos mod/port`	Selects the interface.
Step 2	`Router(config-if)# encapsulation ppp`	Configures the interface for PPP encapsulation.

	Command	Purpose
Step 3	`Router(config-if)# bridge-enable`[1]	Enables BCP on the interface.
Step 4	`Router(config-if)# switchport trunk {allowed | pruning vlan {add | all | except | remove}}`	Configures the trunk characteristics.
Step 5	`Router(config-if)# end`	Exits configuration mode.
Step 6	`Router# show interface pos` *mod/num*	Displays interface configuration.

1. Enter the **bridge-enable** command while the port is in shutdown state. If you enter the **bridge-enable** command while the port is in up state, enter the **shutdown** command followed by the **no shutdown** command in order to bring up BCP on the POS port.

In this example, BCP forwarding of all VLANs except for VLAN 400 is configured on a POS interface:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface pos 3/2
Router(config)# encapsulation ppp
Router(config-if)# bridge-enable
Router(config-if)# switchport trunk allowed vlan all
Router(config-if)# switchport trunk allowed vlan remove vlan 400
Router(config-if)# shutdown
Router(config-if)# no shutdown
Router(config-if)# end
Router# show running-config interface pos 3/2
!
interface POS3/2
 ip address 2.2.2.2 255.255.255.0
 encapsulation ppp
  bridge-enable
 switchport
 switchport trunk allowed vlan 1-399,401-1005
 switchport mode trunk
 no cdp enable
end

Router# show interface pos 3/2 switchport
Name:Po3/2
Switchport:Enabled
Administrative Mode:trunk
Operational Mode:trunk
Administrative Trunking Encapsulation:dot1q
Operational Trunking Encapsulation:dot1q
Negotiation of Trunking:Off
Access Mode VLAN:1 (default)
Trunking Native Mode VLAN:1 (default)
Administrative private-vlan host-association:none
Administrative private-vlan mapping:none
Operational private-vlan:none
Trunking VLANs Enabled:1-399,401-1005
Pruning VLANs Enabled:2-1001

Router# show interface pos 3/2 trunk

Port      Mode         Encapsulation  Status        Native vlan
Po3/2     on           802.1q         trunking      1

Port      Vlans allowed on trunk
Po3/2     1-399,401-1005

Port      Vlans allowed and active in management domain
Po3/2     1,31-32,34,91-92,100,500,1002-1005
```

```
Port      Vlans in spanning tree forwarding state and not pruned
Po3/2     31-32,34,91-92,100,1002-1005
```

OC-3c/STM-1 POS Module Configuration Example

The following is an example of configuration file commands for a Cisco 7600 series router (first router) with an OC-3c/STM-1 POS module in slot 3 connected back-to-back with a Cisco 7500 series router (second router) with a POS Interface Processor (POSIP) module in slot 3.

The configuration commands for the first router are as follows:

```
interface pos 3/1
ip address 10.1.2.3 255.0.0.0
clock source internal
no shutdown
no keepalive
no cdp enable
no ip mroute-cache
crc 32
```

The configuration commands for the second router are as follows:

```
interface pos 3/0/0
ip address 10.1.2.4 255.0.0.0
clock source internal
no shutdown
no keepalive
no cdp enable
crc 32
```

Configuring Multipoint Bridging

Multipoint bridging enables point-to-multipoint bridging for Frame Relay data-link connection identifiers (DLCIs). This feature allows the use of multiple DLCIs per VLAN for bridging on the following OSMs:

- 8-Port OC-3 POS
- 16-Port OC-3 POS
- 2-Port OC-12 POS
- 4-Port OC-12 POS:
- 1-Port OC-48 POS
- 2-Port OC-48 POS/DPT

Multipoint bridging allows service providers to add support for Ethernet-based Layer 2 services to the proven technology of their existing Frame Relay legacy networks. Customers can then use their current VLAN-based networks over the 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.

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

Note 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.

In Cisco IOS Release 12.2(18)SXE, multipoint bridging supports the following modes of operation:

- Raw (default)—Default bridging access mode, in which the bridged connection acts on and transmit bridge protocol data unit (BPDU) packets.
- Access—Access-only bridging access mode, in which the bridged connection does not act on or transmit BPDU packets.
- 802.1Q—Performs IEEE 802.1Q tagging to preserve the class of service (CoS) information from the Ethernet frames across the ATM network.
- 802.1Q Tunnel—IEEE 802.1Q tunneling mode, so that service providers can use one VLAN to support customers with multiple VLANs, as well as accept untagged frames.

Restrictions and Usage Guidelines

The following restrictions apply to the Multipoint Bridging feature:

- Supported only on Enhanced OSMs; non-enhanced OSMs are not supported.
- Multipoint bridging on Frame Relay interfaces supports only IETF encapsulation. Cisco encapsulation is not supported on when doing multipoint bridging.
- VLAN ID 1 is not available as a bridge domain for doing multipoint bridging.
- Multipoint bridging supports an absolute maximum of 60 VCs per each VLAN, and an absolute maximum number of VLANs per peer is 4096. We recommend configuring at most 30 VCs per VLAN, with at most 1024 VLANs per VC.

Prerequisites

The following prerequisites apply to Multipoint Bridging:

- VLANs must be manually added to the VLAN database, using the **vlan** command, to be able to use those VLANs in multipoint bridging.

Configuring Multipoint Bridging for Frame-Relay Interfaces

This section describes how to configure multipoint bridging on Frame-Relay interfaces. You can configure multipoint bridging on individual DLCI circuits. You can optionally add 802.1Q tagging or 802.1Q tunneling. To perform this configuration, use the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan** {*vlan-id* | *vlan-range*}
4. **interface** *iftype slot/port*
5. **no ip address**
6. **encapsulation frame-relay ietf**

Note The **encapulsulation frame-relay ietf** command does not work with Cisco encapsulation.

7. **mls qos trust** {**cos** | **dscp**}
8. **interface** *iftype slot/port.subinterface* {**multipoint** | **point-to-point**}
9. **mls qos trust** {**cos** | **dscp**}
10. **frame-relay interface-dlci** *dlci* [**ietf**]
11. **bridge-domain** *vlan-id* [**access** | **dot1q** | **dot1q-tunnel**] [**split-horizon**]
12. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**vlan** {*vlan-id* | *vlan-range*}
	
	Example:
	`Router(config)# vlan 2,5,10-12,20,25,4000`
	`Router(config-vlan)#`
		Adds the specified VLAN IDs to the VLAN database and enters VLAN configuration mode.
		• *vlan-id*—Specifies a single VLAN ID. The valid range is from 1 to 4094 (but VLAN 1 is not supported for multipoint bridging).
		• *vlan-range*—Specifies multiple VLAN IDs, as either a list or a range. The *vlan-range* can contain a list of the VLAN IDs, separated either by a comma (,), dash (-), or both.
		Note You must manually enter a VLAN ID into the VLAN database before you can use that VLAN for multipoint bridging.

	Command or Action	Purpose

Step 4	**interface** *iftype slot/port*
	
	Example:
	`Router(config)# interface pos 4/1`
	`Router(config-if)#`
		Enters configuration mode for the specified interface.

Step 5	**no ip address**
	
	Example:
	`Router(config-if)# no ip address`
	`Router(config-if)#`
		Removes the IP address, if any, that is configured on the interface.

Step 6	**encapsulation frame-relay ietf**
	
	Example:
	`Router(config-if)# encapsulation frame-relay ietf`
	`Router(config-if)#`
		Enables Frame Relay encapsulation on the interface, using IETF encapsulation. You must specify the **ietf** keyword either here or in Step 10 for each individual DLCI.
		
		Note Multipoint bridging does not support Cisco encapsulation using the **cisco** keyword.

	Note The **ietf** keyword is not available for a Frame Relay DLCI on a multipoint interface. If you want to configure a bridge-domain on a DLCI attached to a multipoint interface you must first enable Frame Relay encapsulation on the interface using IETF encapsulation.	

Step 7	**mls qos trust** {**cos** | **dscp**}
	
	Example:
	`Router(config-if)# mls qos trust cos`
	`Router(config-if)#`
		(Optional) Specifies the trusted state of the interface. The default state is untrusted, but can be changed with one of the following options:
		• **cos**—(Optional) Specifies that the CoS bits in incoming frames are trusted and derives the internal DSCP value from the CoS bits.
		• **dscp**—(Optional) Specifies that the ToS bits in the incoming packets contain a DSCP value.

Step 8	**interface** *iftype slot/port.subinterface* {**multipoint** | **point-to-point**}
	
	Example:
	`Router(config-if)# interface pos 4/1.21`
	`Router(config-subif)#`
		Enters configuration mode for the specified subinterface.

Step 9	**mls qos trust** {**cos** | **dscp**}
	
	Example:
	`Router(config-subif)# mls qos trust cos`
	`Router(config-subif)#`
		(Optional) Specifies the trusted state of the subinterface. The default state is untrusted, but can be changed with one of the following options:
		• **cos**—(Optional) Specifies that the CoS bits in incoming frames are trusted and derives the internal DSCP value from the CoS bits.
		• **dscp**—(Optional) Specifies that the ToS bits in the incoming packets contain a DSCP value.

	Command or Action	Purpose

Step 10	**frame-relay interface-dlci** *dlci*
	
	Example:
	`Router(config-subif)# frame-relay interface-dlci 28`
	`Router(config-fr-dlci)#`
		Creates the specified DLCI on the subinterface and enters DLCI configuration mode.
		• *dlci*—DLCI number to be used on the specified subinterface.
		Note This command includes other options that are not supported when using multipoint bridging.

Step 11	**bridge-domain** *vlan-id* [**access** | **dot1q** | **dot1q-tunnel**] [**split-horizon**]
	
	Example:
	`Router(config-fr-dlci)# bridge-domain 100 dot1q-tunnel`
	`Router(config-fr-dlci)#`
		Enables RFC 1490 bridging to map a bridged VLAN to a PVC. The following options are supported:
		Note This command has additional options that are not supported in a multipoint bridging configuration.
		• *vlan-id*—Number of VLAN to be used in this bridging configuration. The valid range is from 2 to 4094 (but the VLAN ID must have been previously added to the VLAN database in Step 3).
		Note This is the default configuration; frames are not tagged with the dot1q header but STPs and BPDUs are transmitted.
		• **access**—Enables bridging access mode, so that the bridged connection does not act on or transmit BPDUs.
		• **dot1q**—(Optional) Terminates dot1q traffic. Also enables IEEE 802.1Q tagging to preserve the class of service (CoS) information from the Ethernet frames across the ATM network. Without this option, the COS values are not preserved.
		• **dot1q-tunnel**—(Optional) Enables IEEE 802.1Q tunneling mode, so that service providers can use one VLAN to support customers with multiple VLANs, as well as accept untagged frames.
		Note The **access**, **dot1q**, and **dot1q-tunnel** options are mutually exclusive. If you do not specify any of these options, the connection operates in “raw” bridging access mode, which is similar to **access**, except that the connection does act on and transmit BPDU packets.
		Note **split-horizon**—(Optional) Enables RFC 1490 split horizon mode to globally prevent bridging between PVCs in the same VLAN.

Step 12	**end**
	
	Example:
	`Router(config-fr-dlci)# end`
	`Router#`
		Exits DLCI configuration mode and returns to privileged EXEC mode.

The following is an example of a Multipoint Bridging configuration on a Frame-Relay interface:

```
frame-relay switching
...
!
interface POS3/8
 no ip address
 encapsulation frame-relay ietf
 logging event link-status
 mls qos trust dscp
 clock source internal
 frame-relay intf-type dce
!
interface POS3/8.10 multipoint
 mls qos trust dscp
 frame-relay interface-dlci 120
  bridge-domain 100 dot1q-tunnel
 frame-relay interface-dlci 130
  bridge-domain 100 dot1q-tunnel
```

Configuring Strict Priority LLQ Support on POS Optical Service Modules

Starting with Cisco IOS Release 12.2(18)SXE, the Low Latency Queuing feature is changed for the Packet over SONET (POS) Optical Services Modules. With this change, priority queue policing is supported on these OSMs. Using Hierarchical Queuing Framework (HQF), the **police** command is combined with strict priority in a class on the OSM.

Note This command is supported on the OC-3 and OC-12 modules. It is not supported on the OC-48 modules. The **priority percent** *%* and **priority** *kbps* commands from previous releases are no longer supported on the OC-3 and OC-12 modules. However, these commands are still supported on the POS OC-48 OSM.

If a second priority police class is included in the policy, police must be configured first.

To configure strict priority LLQ support, perform the following tasks, starting in global configuration mode:

	Command or Action	Purpose
Step 1	`Router(config)# policy-map` *policy-name* **Example:** `Router(config)# policy-map policy11`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-pmap)# class` *class-name* **Example:** `Router(config)# class class204`	Specifies the name of a predefined class included in the service policy.

	Command or Action	Purpose

Step 3	`Router(config-pmap-c)# priority`
	
	`Example:Router(config)# priority`
		Configures the strict priority class.

Step 4	`Router(config-pmap-c)# police rate`
	
	Example:
	`Router(config-pmap-c) # police 1000000#`
		Sets the policing rate (in bps).

Examples

The following example shows a typical configuration and verification for the supported POS OSMs:

```
!
 Policy Map child-pos
   Class prec1
     priority
    police cir 1000000 bc 31250 be 31250 conform-action transmit exceed-action drop
   Class prec2
     bandwidth remaining 50 (%)
   Class prec3
     bandwidth remaining 30 (%)
   Class class-default
     bandwidth remaining 20 (%)
!
   Class class-default
     bandwidth 2200 (kbps)
     shape average 3000000 12000 12000
     service-policy child-pos
!
interface POS3/2
no ip address
encapsulation frame-relay
mls qos trust dscp
clock source internal
end
!
interface POS3/2.16 point-to-point
ip address 25.0.0.1 255.255.255.0
mls qos trust dscp
no cdp enable
frame-relay interface-dlci 16
service-policy output parent-pos
end
```

The following show command verifies the configuration:

```
Router #show policy interface pos3/2.16

POS3/2.16

 Service-policy output:parent-pos

   Class-map:class-default (match-any)
     0 packets, 0 bytes
     5 minute offered rate 0 bps, drop rate 0 bps
     Match:any
     Queueing
     queue limit 550 (packets)
```

```
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts queued/bytes queued) 0/0
        bandwidth 2200 kbps
        shape (average) cir 3000000, bc 12000, be 12000
        target shape rate 3000000
        (shape parameter is rounded to 2944000 bps due to granularity)
          lower bound cir 0,  adapt to fecn 0

        Service-policy :child-pos

          Class-map:prec1 (match-all)
            0 packets, 0 bytes
            5 minute offered rate 0 bps, drop rate 0 bps
            Match:ip precedence 1
            Priority:b/w exceed drops:0
            police:
                cir 1000000 bps, bc 31250 bytes
              (Police cir is rounded to 983040 bps due to granularity)

          Class-map:prec2 (match-all)
            0 packets, 0 bytes
            5 minute offered rate 0 bps, drop rate 0 bps
            Match:ip precedence 2
            Queueing
            queue limit 150 (packets)
            (queue depth/total drops/no-buffer drops) 0/0/0
            (pkts queued/bytes queued) 0/0
            bandwidth remaining 50% (600 kbps)
            (bandwidth parameter is rounded to 504 kbps due to granularity)

          Class-map:prec3 (match-all)
            0 packets, 0 bytes
            5 minute offered rate 0 bps, drop rate 0 bps
            Match:ip precedence 3
            Queueing
            queue limit 90 (packets)
            (queue depth/total drops/no-buffer drops) 0/0/0
            (pkts queued/bytes queued) 0/0
            bandwidth remaining 30% (360 kbps)
            (bandwidth parameter is rounded to 300 kbps due to granularity)

          Class-map:class-default (match-any)
            0 packets, 0 bytes
            5 minute offered rate 0 bps, drop rate 0 bps
            Match:any
            Queueing
            queue limit 60 (packets)
            (queue depth/total drops/no-buffer drops) 0/0/0
            (pkts queued/bytes queued) 0/0
            bandwidth remaining 20% (240 kbps)
            (bandwidth parameter is rounded to 197 kbps due to granularity)
Router#
```

CHAPTER 1

Configuring 4-Port Gigabit Ethernet WAN Optical Services Modules

This chapter provides an overview of the features supported on the 4-port Gigabit Ethernet WAN Optical Services Modules (OSM-2+4GE-WAN+ and OSM-4GE-WAN-GBIC) supported on Cisco Catalyst 6500 series switches and Cisco 7600 series routers.

This chapter consists of these sections:

Supported Features

The following Layer 3 features are supported on the Gigabit Ethernet WAN optical services modules (OSMs):

- Forwarding of distributed IP services
- Multiprotocol Label Switching (MPLS)
- Ethernet over Multiprotocol Label Switching (EoMPLS)
- Frame Relay over MPLS
- ATM cell relay over MPLS VC-Mode
- ATM AAL5 over MPLS
- IOS Modular QoS Command Line Interface (MQC) QoS
- Flow control
- 802.1Q VLAN trunking
- Advanced 802.1Q-to-802.1Q (QinQ) Service Mapping
- Hot Standby Routing Protocol (HSRP)
- Jumbo frames
- Support for up to 32,000 MAC addresses per port

- Support for up to 32,000 simultaneous ACL entries
- Support for up to 32,000 simultaneous QoS entries
- SNMP I and II
- Four RMON groups per port: statistics, history, alarms, and events
- Online insertion and removal (OIR)
- Inter-Switch Link (ISL)

Note The OSM-2+4GE-WAN+ module supports ISL on the Layer 2 Gigabit Ethernet LAN ports but does not support ISL on the Layer 3 Gigabit Ethernet WAN ports.

The Layer 2 Gigabit Ethernet ports on the OSMs are configured from the supervisor engine of the Cisco Catalyst 6500 series switch or the Cisco 7600 series router. For feature support and configuration information for the OSM Layer 2 Gigabit Ethernet ports, refer to the links in the "Layer 2 Software Features" section on page 1-5.

Saving your Configuration Before Upgrading from an OSM-4GE-WAN-GBIC to an OSM-2+4GE-WAN+

When you upgrade from OSM-4GE-WAN-GBIC to an OSM-2+4GE-WAN+, the existing configuration will not be saved and applied to the new OSM-2+4GE-WAN+.

To save your configuration when upgrading from an OSM-4GE-WAN-GBIC to an OSM-2+4GE-WAN+, perform this task:

Step 1 Enter the **write memory** command before removing the OSM-4GE-WAN-GBIC.

Step 2 Install the new OSM-2+4GE-WAN+.

Step 3 Enter the **copy startup-config running-config** command.

Step 4 Enter the **write memory** command.

Warning **The orientation of the GBIC in OSM-4GE-WAN-GBIC ports is reversed (upside down) from those of the LAN ports for OSM-2+4GE-WAN+.**

Gigabit Ethernet WAN Port Configuration

The four Gigabit Ethernet WAN ports on the 4-port Gigabit Ethernet WAN OSMs are controlled by Cisco IOS software and support all standard Cisco IOS features. For configuration information for standard Cisco IOS features and routing protocols supported on the GE-WAN ports, refer to the appropriate Cisco IOS configuration guide and command reference publications at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Basic Interface Configuration

After you verify that the 4-port Gigabit Ethernet WAN OSM is installed correctly, use the **configure** command to configure the Gigabit Ethernet WAN interfaces.

The following procedure is for creating a basic configuration—enabling an interface and specifying IP routing. You might also need to enter other configuration subcommands, depending on the requirements for your system configuration.

Note Subinterfaces on the 4-port Gigabit Etherent WAN module cannot share HSRP group numbers. As a result, only 16 HSRP groups per Gigabit Ethernet WAN port are supported.

Note The MTU size you specify on a main Gigabit Ethernet WAN interface will also apply to all subinterfaces you configure on the main interface. It is not possible to specify an MTU size on a subinterface that is different from the MTU size specified for the main interface.

To configure the Gigabit Ethernet WAN interfaces, perform this task:

Step 1 Confirm that the system recognizes the module by entering the **show version** command:

```
Router# show version
```

Step 2 Check the status of each port by entering the **show interface** command:

```
Router# show interface
```

Step 3 Enter configuration mode and specify that the console terminal will be the source of the configuration subcommands:

```
Router# configure terminal
```

Step 4 Enable IP routing by entering the **ip routing** command:

```
Router(config)# ip routing
```

Step 5 At the prompt, specify the new interface to configure by entering the **interface** command, followed by the *type* (**ge-wan**) and *slot/port* number. The example that follows is for a Gigabit Ethernet WAN OSM in slot 3:

```
Router(config)# interface ge-wan 3/0
```

Step 6 Assign an IP address and subnet mask to the interface with the **ip address** configuration subcommand, as in the following example:

```
Router(config-if)# ip address 10.1.2.3 255.255.255.255
```

By default, a GE-WAN interface is configured for automatic negotiation of link parameters, such as duplex, speed, and flow control. To disable flow control and to force the interface for 1000/full-duplex mode, turn off automatic negotiation with the command:

```
Router(config-if)# no negotiation auto
```

Note Changing the negotiation mode of an active interface flaps the interface by bringing it down and then back up, so as to implement the new negotiation mode. For this reason, we recommend changing the negotiation mode only when the interface is shutdown.

Tip Use the **negotiation auto** command to restore the default of automatic negotiation of link parameters.

Step 7 Change the shutdown state to up and enable the interface:

```
Router(config-if)# no shutdown
```

The **no shutdown** command passes an **enable** command to the Gigabit Ethernet module. It also causes the module to configure itself based on the most recent configuration commands received by the module.

Step 8 Write the new configuration to memory:

```
Router# copy running-config startup-config
```

When the configuration is stored, an OK message appears.

Configuring Strict Priority Low Latency Queuing (LLQ) Support on the OSM-2+4GE-WAN+

Starting with Cisco IOS Release 12.2(18)SXE, the Low Latency Queuing feature is changed for the OSM-2+4GE-WAN+ Optical Services Module. With this change, priority queue policing is supported on the module. Using Hiearchical Queuing Framework (HQF), the **police** command is combined with strict priority in a class on the OSM.

Note The **priority percent** *%* and **priority** *kbps* commands from previous releases are no longer supported.

If a second priority police class is included in the policy, police must be configured first.

To configure strict priority LLQ support, perform the following tasks, starting in global configuration mode:

	Command or Action	Purpose
Step 1	`Router(config)# policy-map` *policy-name* **Example:** `Router(config)# policy-map policy11`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-pmap)# class` *class-name* **Example:** `Router(config)# class class204`	Specifies the name of a predefined class included in the service policy.

	Command or Action	Purpose

Step 3	`Router(config-pmap-c)# priority`
	
	`Example:Router(config)# priority`
		Configures the strict priority class.

Step 4	`Router(config-pmap-c)# police rate`
	
	Example:
	`Router(config-pmap-c) # police 1000000#`
		Sets the policing rate (in bps).

Examples

The following example shows a typical configuration and verification for the OSM-2+4GE-WAN+ OSM.

```
!
 Policy Map child
   Class dscp-ef
     priority
    police cir 1000000 bc 31250 be 31250 conform-action transmit exceed-action drop
   Class dscp-af21
     bandwidth remaining 35 (%)
   Class dscp-af31
     bandwidth remaining 30 (%)
   Class class-default
     bandwidth remaining 25 (%)
!
 Policy Map parent
   Class vlan-2
     bandwidth 5000 (kbps)
     shape average 6000000 24000 24000
     service-policy child
!
interface ge-wan7/1
no ip address
negotiation auto
mls qos trust dscp
service-policy output parent
end
!
interface ge-wan7/1.2
encapsulation dot1Q 2
ip address 10.10.10.1 255.255.255.0
mls qos trust dscp
no cdp enable
end
```

The following show command verifies the configuration:

```
Router#show policy interface ge-wan7/1

 GE-WAN7/1

  Service-policy output: parent

    Class-map: vlan-2 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
```

```
  Match: vlan  2
  Queueing
  queue limit 1250 (packets)
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts queued/bytes queued) 0/0
  bandwidth 5000 kbps
  shape (average) cir 6000000, bc 24000, be 24000
  target shape rate 6000000
(shape parameter is rounded to 5952000 bps due to granularity)

  Service-policy : child

    Class-map: dscp-ef (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp ef
  Priority: b/w exceed drops: 0
      police:
          cir 1000000 bps, bc 31250 bytes
        (Police cir is rounded to 983040 bps due to granularity)

    Class-map: dscp-af21 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp af21
      Queueing
      queue limit 350 (packets)
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts queued/bytes queued) 0/0
      bandwidth remaining 35% (1400 kbps)
      (bandwidth parameter is rounded to 1392 kbps due to granularity)

    Class-map: dscp-af31 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp af31
      Queueing
    queue limit 300 (packets)
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts queued/bytes queued) 0/0
      bandwidth remaining 30% (1200 kbps)
      (bandwidth parameter is rounded to 1196 kbps due to granularity)

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      Queueing
      queue limit 250 (packets)
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts queued/bytes queued) 0/0
      bandwidth remaining 25% (1000 kbps)

Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
```

```
      queue limit 248750 (packets)
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts queued/bytes queued) 0/0
Router#
```

Quality of Services

The Gigabit Ethernet WAN modules support the following QoS implementations:

- Differentiated Services Code Point (DSCP) classification
- IP-precedence classification
- Class-based traffic shaping
- Class-based weighted fair queuing (CBWFQ)—Supported on the OSM-2+4GE-WAN+ only
- Low latency queuing (LLQ)—Supported on the OSM-2+4GE-WAN+ only
- Weighted Random Early Detection (WRED)—Supported on the OSM-2+4GE-WAN+ only
- Hierarchical traffic shaping for dot1q encapsulations—Supported for egress traffic on subinterfaces on the OSM-2+4GE-WAN+ only
- EoMPLS Support with CBWFQ, LLQ and WRED - CBWFQ, LLQ or WRED are applied to the EoMPLS uplink interface. Supported on the OSM-2+4GE-WAN+ only

For QoS configuration information and examples for the WAN OSM ports, see the "Configuring QoS on the OSMs" section on page 9-2.

See Chapter 10, "Configuring Destination Sensitive Services on the Optical Services Modules" for configuration information.

Advanced QinQ Service Mapping

The IEEE 802.1Q VLAN specification provides for a trunking option that tags packets with two VLAN tags:

- An inner tag that specifies the customer tag
- An outer tag that specifies the service provider tag—to allow multiple VLANs to be trunked together across an intermediate network.

This type of double-tagged tunnel is referred to as IEEE 802.1Q-in-802.1Q (Q-in-Q) tunneling.

Standard QinQ tunneling, however, is limited. Although double-tagged VLANs can identify different customers, they cannot easily distinguish different service flows for the same customer. You can use separate VLANs for each service flow, but IEEE 802.1Q VLANs are limited to a maximum of 1,024 VLANs. Extended VLANs have a maximum of 4,096 per router, but even this larger number could be exhausted if many customers are using multiple services.

The Advanced QinQ Service Mapping feature solves these problems by enabling the Gigabit Ethernet WAN (GE-WAN) interfaces on the OSM-2+4GE-WAN+ Optical Services Module (OSM) to act as a QinQ access gateway. The access gateway enhances QinQ tunneling by using the combination of inner and outer VLAN tags as a unique identifier for a particular customer's service flows. This allows the interface to perform the following:

- Translates packets that are tagged with an inner CE VLAN tag and an outer PE VLAN tag to a specifying outgoing trunk VLAN on the basis of the unique combination of CE and PE VLAN tags. Two types of packet translation are supported:

  - QinQ Translation (also known as double-tag to single-tag translation)—The CE and PE tags from the original incoming packet are replaced with a single trunk VLAN tag when the outgoing packet is transmitted.
  - QinQ Transparent Tunneling (also known as double-tag to double-tag translation)—The outer PE tag from the original incoming packet is replaced with an outer trunk VLAN tag when the outgoing packet is transmitted. The inner CE VLAN tag is left unchanged in the outgoing packet.
- Supports traffic shaping on the basis of the unique combination of CE and PE VLAN tags.
- Sets the IEEE 802.1P prioritization bits (P bits) in the outgoing trunk VLAN tag by copying the P bits either from the original packet's outer PE VLAN tag or from the original packet's inner CE VLAN tag.

In Cisco IOS Release 12.2(18)SXE and later releases, you can also combine multiple GE-WAN interfaces into a virtual QinQ link bundle (also known as a port-channel). This simplifies configuration and allows the system to automatically load balance the traffic moving across the physical interfaces.

See the following sections for more details on the QinQ translation process and on using QinQ link bundles.

QinQ Translation—Double Tag to Single Tag Translation

In a double-tag-to-single-tag translation, the Advanced QinQ Service Mapping feature replaces both the inner customer edge (CE) VLAN tag and the outer provider edge (PE) VLAN tag with a single trunk VLAN tag. The following shows the format of both the incoming original packet and the outgoing translated packet.

Original Incoming Packet							
DA	**SA**	**ETYPE= 0x8100**	**PE VLAN Tag**	**ETYPE=[1] 0x8100**	**CE VLAN[1] Tag**	**Data**	**FCS**
Outgoing Translated Packet							
DA	**SA**	**ETYPE= 0x8100**	**Trunk VLAN Tag**	**Data**	**FCS**		

1. The CE VLAN tag might not be present if the customer did not tag this packet with a VLAN ID before transmitting it to the service provider. The PE VLAN tag should always be present.

When the interface receives a packet, the following occurs:

- Examines the inner CE VLAN tag and outer PE VLAN tag, and uses that unique combination to perform the quality of service processing, rate shaping, and switching that is specified by the attached service policy map.

  If the packet includes a PE VLAN tag, but no mapping has been configured for this particular CE VLAN tag, or if the incoming packet does not contain any inner CE VLAN tag, the interface drops the packet (unless a subinterface has been configured for out-of-range packets).
- Removes the inner and outer VLAN tags and replaces them with the trunk VLAN tag that has been configured on the VLAN's subinterface.

- Sets the 802.1P bits (P bits) on the trunk VLAN tag in one of the following ways, depending on the service policy map being used:
  - Copies the P bits that were in the outer PE VLAN tag to the trunk VLAN tag (default).
  - Copies the P bits that were in the inner CE VLAN tag to the trunk VLAN tag (if the **set cos cos-inner** command was used in the service policy map).
  - Zeroes out the P bits if the interface or subinterface has been marked as untrusted.
- Forwards the translated single-tagged packet to the appropriate destination or service.

QinQ Transparent Tunneling—Double Tag to Double Tag Translation

When you configure the Advanced QinQ Service Mapping feature for double-tag-to-double-tag conversion, the Gigabit Ethernet WAN interface replaces the outer PE VLAN tag with the trunk VLAN tag. The inner CE VLAN tag remains unchanged. The following shows the format of both the incoming original packet and the outgoing translated packet:

Original Incoming Packet							
DA	**SA**	**ETYPE= 0x8100**	**PE VLAN Tag**	**ETYPE=[1] 0x8100**	**CE VLAN[1] Tag**	**Data**	**FCS**
Outgoing Translated Packet							
DA	**SA**	**ETYPE= 0x8100**	**Trunk VLAN Tag**	**ETYPE= 0x8100**	**CE VLAN Tag**	**Data**	**FCS**

1. The CE VLAN tag might not be present if the customer did not tag this packet with a VLAN ID before transmitting it to the service provider, in which case this becomes a single-tag to single-tag translation.

When the interface receives a packet, the following occurs:

- Examines the inner CE VLAN tag and outer PE VLAN tag, and uses that unique combination to perform the quality of service processing, rate shaping, and switching that is specified by the attached service policy map.

  If the packet includes a PE VLAN tag, but no mapping has been configured for this particular CE VLAN tag, or if the incoming packet does not contain any inner CE VLAN tag, the interface drops the packet (unless a subinterface has been configured for out-of-range packets).
- Removes the outer PE VLAN tag and replaces it with the trunk VLAN tag that is configured on the VLAN's subinterface. The inner CE VLAN tag is left unchanged.
- Sets the 802.1P bits (P bits) on the trunk VLAN tag in one of the following ways, depending on the service policy map being used:
  - Copies the P bits that were in the outer PE VLAN tag to the trunk VLAN tag (default).
  - Copies the P bits that were in the inner CE VLAN tag to the trunk VLAN tag (if the **set cos cos-inner** command was used in the service policy map).
  - Zeroes out the P bits if the interface or subinterface has been marked as untrusted.
- Forwards the translated double-tagged packet to the appropriate destination or service.

Out-of-Range and Unspecified In-Range Packets

Each PE VLAN supports a maximum of 32 CE VLANs, which must be in a contiguous block that starts on a number divisible by 32 (for example: 0, 32, 64, and so on). When you specify the first CE VLAN ID for a PE VLAN (using the **bridge-domain** command), the Cisco IOS software automatically associates the corresponding block of 32 IDs with that PE VLAN. Any other CE VLANs are considered out-of-range for that particular PE VLAN.

For example, specifying a CE VLAN ID of 131 automatically associates the CE VLAN IDs from 128 to 159 with that particular PE VLAN. Any CE VLANs that are outside of that block (from 1 to 127 and from 160 to 4094) are considered out-of-range. In addition, if a packet arrives without a CE VLAN tag, it is also considered to be out-of-range.

The default behavior is to drop all out-of-range packets that are received on an interface that has been configured for QinQ translation. You can change this behavior by configuring a subinterface to match out-of-range packets.

The QinQ access gateway interface also drops any packets with a CE VLAN that is in-range (within the block of 32 VLAN IDs) but not explicitly mapped on a subinterface. This behavior cannot be changed. For example, if you specify a CE VLAN of 32 and no other CE VLANs for a particular PE VLAN, the interface drops packets for that PE VLAN that have CE VLANs from 33 and 63.

Per VLAN Load Balancing for Advanced QinQ Service Mapping

In Cisco IOS Release 12.2(18)SXE and later releases, you can combine multiple GE-WAN interfaces into a QinQ link bundle, which is a virtual interface that you configure in the same way as the physical GE-WAN interfaces. Using QinQ link bundles has the following advantages:

- Simplifies configuration because you do not have to configure the individual GE-WAN physical interfaces. Instead, you configure only the one virtual interface with the required QinQ parameters, and those parameters are used for all of the physical interfaces in the bundle.
- Increases bandwidth by allowing you to aggregate individual physical interfaces into a single logical interface.
- Increases availability because if one link in the bundle goes down, the traffic is reallocated among the remaining interfaces until the link is reestablished.
- Enables load-balancing of PE VLANs among the physical interfaces. When the PE VLANs are created, they are automatically distributed among the physical interfaces in the bundle in a round-robin fashion. Adding or removing a physical interface to the QinQ link bundle automatically reallocates the PE VLANs among the physical interfaces, with a minimal interruption of the traffic flows along those VLANs.

  Note The load-balancing algorithm is based only on the number of PE VLANs, where all of the packets for a particular PE VLAN are sent through the same physical interface. The load-balancing does not take into account the bandwidth or the number of the individual CE VLANs that are being transported in each PE tunnel. The assignment of a particular PE VLAN is determined when the PE VLAN is first created, and this assignment does not change unless interfaces are added or removed from the QinQ link bundle.

- Allows you to logically group physical interfaces according to your management needs, such as application or location. You can obtain aggregate interface statistics by displaying the interface statistics for the bundle's virtual interface, as well as displaying the statistics for each of the individual physical interfaces in the bundle.

- Simplifies network management by allowing you to perform OIR and other maintenance operations on interfaces and cards in the QinQ link bundle without stopping the traffic flows. Instead, the traffic is automatically redistributed among the remaining physical interfaces. When the card and its interfaces are brought back up, the traffic is again redistributed among all of the slots in the bundles.
- Allows you to move OSM-2+4GE-WAN+ modules between slots without having to re-enter the complete interface configuration. Instead, you only have to remove the old interfaces from the QinQ link bundle and then add the new interfaces to the bundle. The bundle's configuration is then automatically applied to the card in its new location.
- Requires a minimal learning curve to learn, because QinQ link bundles are created using the same **port-channel** and **channel-group** commands that are used on LAN interfaces to create Ether Channels. The same monitoring and maintenance procedures that are used for Ether Channels can be used for QinQ link bundles.

Configuring Advanced QinQ Service Mapping

This section describes the following configuration tasks that are needed to enable and configure the Advanced QinQ Service Mapping feature:

Enabling IEEE 802.1Q-in-802.1Q Translation on a Gigabit Ethernet WAN Interface

To use the Advanced QinQ Service Mapping feature, you must first enable IEEE 802.1Q-in-802.1Q translation on the Gigabit Ethernet WAN interface that is connected to the provider edge router through the Metro Ethernet network. You can also optionally configure the interface as trusted, if you want to preserve the IEEE 802.1P bits (P bits) that are in the IEEE 802.1Q header of incoming packets.

To enable IEEE 802.1Q-in-802.1Q translation on a Gigabit Ethernet WAN interface, and optionally configure the interface as trusted, use the following procedure.

Prerequisites

- This feature requires a Cisco Catalyst 6500 series switch or Cisco 7600 series router with a Cisco Supervisor Engine 2 or Supervisor Engine 720 module that is running Cisco IOS Release 12.2(18)SXD or later.
- This feature is supported only on the Gigabit Ethernet WAN (GE-WAN) interfaces on the OSM-2+4GE-WAN+ Gigabit Ethernet Enhanced Optical Services Module (OSM). This feature is not supported on other port adapter modules or on LAN Gigabit Ethernet (GE) interfaces.
- The Cisco IOS software image must support the OSM-2+4GE-WAN+ Gigabit Ethernet Enhanced OSM card.

- You must remove all IP, MPLS, and other Layer 3 configuration on the main interface before enabling IEEE 802.1Q-in-802.1Q translation.

Note When a GE-WAN interface is configured for QinQ operation, the Cisco IOS command-line interface (CLI) blocks any IP configuration, but it is still possible to configure other Layer 3 features. All such configuration must be removed from the interface before QinQ can operate successfully.

Restrictions

- This configuration is supported only on the Gigabit Ethernet WAN interfaces on the OSM-2+4GE-WAN+ enhanced Optical Services Module (OSM).
- Only the main interface can be configured as an QinQ access gateway. Subinterfaces are then configured to specify the specific VLAN mappings.
- A Gigabit Ethernet WAN interface that is configured as a QinQ access gateway cannot also be configured with any IP, MPLS, or other Layer 3 configurations. Adding such configuration to the interface can interfere with the QinQ operation.
- Multiprotocol Label Switching (MPLS) Experimental (EXP) bit mappings and hierarchical QoS are not supported on the Gigabit Ethernet WAN interface being used for QinQ translation.
- Each provider edge (PE) VLAN (or outer VLAN) supports a a maximum of 32 consecutive customer edge (CE) VLANs (or inner VLANs). This range of CE VLANs must start on a boundary that is divisible by 32 (for example, 32 to 63, 64 to 95, and so on, up to 4000 to 4031, 4032 to 4063, and 4064 to 4094). The invalid or reserved VLANs are excluded from this rule. For example, the first range is 2 to 31 because VLAN 0 is not valid and VLAN 1 is, by default, reserved for a native VLAN. Each PE VLAN also supports one default function that is applied to VLANs that fall outside of this range of 32 VLANs.
- A PE VLAN can be configured on only one Gigabit Ethernet WAN interface in the router.
- A PE VLAN cannot have the same ID as the native VLAN that is also being used on any interface in the router. The default native VLAN for interfaces is VLAN ID 1, and we recommend using this default to simplify the use of QinQ tagging on the router.
- A PE VLAN cannot have the same ID as an MPLS-related VLAN ID being used on the same interface or on its paired interface. GE-WAN interfaces 1 and 2 constitute one pair, and GE-WAN interfaces 3 and 4 constitute another pair.

  For example, if interface 1 assigns VLAN ID 200 to an MPLS-based feature (such as MPLS VPN, AToM, or VPLS), you cannot use VLAN 200 as a PE VLAN on either interface 1 or 2. However, you still can use VLAN 200 on interface 3 or 4, because those interfaces are a separate interface pair.
- VLAN 4095 is reserved and cannot be used as a CE VLAN. Packets that contain a CE VLAN ID of 4095 are automatically dropped by subinterfaces that are configured for QinQ translation. VLAN 4095, however, can continue to be used as a native (non-QinQ) VLAN.
- VLAN IDs from 1006 to 4094 can be used for either PE VLANs or internal VLANs. Since internal VLANs are automatically allocated for certain features such as Layer 3 LAN ports, WAN interfaces, and WAN subinterfaces, you must coordinate your use of PE VLANs with the system's use of internal VLANs. In particular, be sure to reserve some of the VLAN IDs between 1006 and 4094 for use as internal VLANS, because internal VLANs cannot use IDs between 1 and 1005. If you run out of VLANs for use as internal VLANs, you might not be able to install new cards or use certain software features.

The router, by default, allocates internal VLANs starting with 1006 and ascending sequentially. We recommend that you change this behavior with the **vlan internal allocation policy descending** global configuration command, so that the router allocates internal VLANs starting with 4094 and descending sequentially.

Note If you change the allocation method, you must reboot the router before the changes take effect. This is because a number of internal VLANs are automatically allocated at router startup.

Tip To display the number of internal VLANs that are currently in use, use the **show vlan internal usage** command.

- The **match vlan** command is not supported in this feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan internal allocation policy descending**
4. **interface ge-wan** *slot/port*
5. **no ip address**
6. **mode dot1q-in-dot1q access-gateway**
7. **description** *string*
8. **no shutdown**
9. **end**

Tip You do not need to configure the **mls qos trust** command to preserve the CoS bits in the VLAN translation, because this command has no effect on a GE-WAN interface that has been configured with the **mode dot1q-in-dot1q access-gateway** command. When an interface or port-channel group has been configured for QinQ translation, it always trusts the VLAN Class of Service (CoS) bits.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**vlan internal allocation policy descending**
	
	Example:
	`Router(config)# vlan internal allocation policy descending`
	`Router(config)#`
		(Optional) Allocates internal VLANs starting with 4094 and descending sequentially. We recommend this configuration to avoid conflicts with the PE VLAN ID assignment.
		Note If you change the allocation method, you must reboot the router before the changes take effect. This is because a number of internal VLANs are automatically allocated at router startup.

Step 4	**interface ge-wan** *slot/port*
	
	Example:
	`Router(config)# interface ge-wan 5/1`
	`Router(config-if)#`
		Enters interface configuration mode for the specified Gigabit Ethernet WAN interface on the OSM-2+4GE-WAN+ Gigabit Ethernet WAN port.

Step 5	**no ip address**
	
	Example:
	`Router(config-if)# no ip address`
	`Router(config-if)#`
		(Optional) Removes the IP address that might be configured on the interface. This step is required if the interface has been configured previously with an IP address.

Step 6	**mode dot1q-in-dot1q access-gateway**
	
	Example:
	`Router(config-if)# mode dot1q-in-dot1q access-gateway`
	`Router(config-if)#`
		Enables IEEE 802.1Q-in-802.1Q translation on the interface, enabling the Advanced QinQ Service Mapping feature.

Step 7	**description** *string*
	
	Example:
	`Router(config-if)# description Connected to ISP ABC Port SJ-2`
	`Router(config-if)#`
		(Optional) Provides a description of this interface. The *string* parameter can be any arbitrary text that describes the interface, its neighbor, its purpose, or any other information that might be useful for maintaining and troubleshooting problems with this interface and configuration.

Step 8	**no shutdown**
	
	Example:
	`Router(config-if)# no shutdown`
	`Router(config-if)#`
		Activates the interface and enables it to pass traffic.

	Command or Action	Purpose
	Note Repeat Step 4 through Step 8 for each Gigabit Ethernet WAN interface to be configured.	

Step 9	`end`
	
	Example:
	`Router(config-if)# end`
	`Router#`
		Exits interface configuration mode and returns to privileged EXEC mode.

Examples

The following example shows a sample configuration for a Gigabit Ethernet WAN interface:

```
!
interface GE-WAN3/4
 description connected to SJ QinQ Tunnel
 no ip address
 logging event link-status
 negotiation auto
 mode dot1q-in-dot1q access-gateway
```

Enabling IEEE 802.1Q-in-802.1Q Translation on a QinQ Link Bundle

To use the Advanced QinQ Service Mapping feature on a QinQ link bundle, you must create a virtual port-channel interface and enable IEEE 802.1Q-in-802.1Q translation on that interface. You then must assign Gigabit Ethernet WAN interfaces to the port-channel group. To perform these tasks, use the following procedure.

Prerequisites

- The QinQ link bundle feature requires a Cisco Catalyst 6500 series switch or Cisco 7600 series router with a Cisco Supervisor Engine 2 or Supervisor Engine 720 module that is running Cisco IOS Release 12.2(18)SXE or later.
- When using the QinQ link bundle feature, the port-channel group must include only Gigabit Ethernet WAN (GE-WAN) interfaces on the OSM-2+4GE-WAN+ Gigabit Ethernet Enhanced Optical Services Module (OSM). This feature is not supported on other port adapter modules or on LAN Gigabit Ethernet (GE) interfaces.
- The Cisco IOS software image must support the OSM-2+4GE-WAN+ Gigabit Ethernet Enhanced OSM card.

Restrictions

- All restrictions listed for the Gigabit Ethernet WAN interfaces also apply to the use of QinQ link bundling. See the "Restrictions" section on page 4-12 for a list of those restrictions.
- Channel groups that are being used for QinQ link bundling can contain only GE-WAN interfaces on the OSM-2+4GE-WAN+ Optical Services Module (OSM) card.
- Port-channel interfaces that are being used for QinQ link bundling must not be configured for a Maximum Transmission Unit (MTU) value greater than 9170 bytes, which is the maximum MTU that is supported on the OSM-2+4GE-WAN+ OSM card.

- Only the **mode on** option is supported when using the **channel-group** command with GE-WAN interfaces on the OSM-2+4GE-WAN+ Optical Services Module for advanced QinQ translation. The other mode options are not supported on a QinQ link bundle.
- You cannot use the **channel-group** command on GE-WAN interfaces if Multiprotocol Label Switching (MPLS) is configured. You must remove all **mpls** configuration commands from the interface before using the **channel-group** command.
- You cannot attach a service policy to the main port-channel interface or to the individual member interfaces of the port-channel group. Instead, you must attach the service policy to the appropriate port-channel subinterfaces. Also, input service policies are not supported on port-channels being used for QinQ link bundling.
- Service policies for QinQ port-channel interfaces support only the **shaping** and **set cos cos-inner** commands. You cannot use other commands, such as the **bandwidth** command, on QinQ port-channel interfaces.
- Port-channel interface counters can be displayed with the **show interface port-channel** {*number* | *number.subif*} command. However, the **show interface port-channel counters** and **show counters interface port-channel** commands are not supported for channel groups that are using GE-WAN interfaces for QinQ link bundling.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan internal allocation policy descending**
4. **interface port-channel** *number*
5. **no ip address**
6. **mode dot1q-in-dot1q access-gateway**
7. **description** *string*
8. **no shutdown**
9. **interface ge-wan** *slot/port*
10. **no ip address**
11. **channel-group** *number* **mode on**
12. **no shutdown**
13. **end**

Tip You do not need to configure the **mls qos trust** command to preserve the CoS bits in the VLAN translation, because this command has no effect on a GE-WAN interface or port-channel group that has been configured with the **mode dot1q-in-dot1q access-gateway** command. When an interface or port-channel group has been configured for QinQ translation, it always trusts the VLAN Class of Service (CoS) bits.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**vlan internal allocation policy descending**
	
	Example:
	`Router(config)# vlan internal allocation policy descending`
	`Router(config)#`
		(Optional) Allocates internal VLANs starting with 4094 and descending sequentially. We recommend this configuration to avoid conflicts with the assignment of IDs for the PE VLANs.
		Note If you change the allocation method, you must reboot the router before the changes take effect. This is because a number of internal VLANs are automatically allocated at router startup.

Step 4	**interface port-channel** *number*
	
	Example:
	`Router(config)# interface port-channel 5`
	`Router(config-if)#`
		Creates a virtual port-channel interface and enters interface configuration mode. The valid range for *number* is from 1 to 256.

Step 5	**no ip address**
	
	Example:
	`Router(config-if)# no ip address`
	`Router(config-if)#`
		(Optional) Removes the IP address that might be configured on the interface. This step is required if the interface has been configured previously with an IP address.

Step 6	**mode dot1q-in-dot1q access-gateway**
	
	Example:
	`Router(config-if)# mode dot1q-in-dot1q access-gateway`
	`Router(config-if)#`
		Enables IEEE 802.1Q-in-802.1Q translation on the interface, enabling the Advanced QinQ Service Mapping feature.
		Note This command cannot be used on a port-channel that already contains a channel group member that is not a GE-WAN interface on a OSM-2+4GE-WAN+ card.

Step 7	**description** *string*
	
	Example:
	`Router(config-if)# description QinQ Link Bundle connected to LA-10/1`
	`Router(config-if)#`
		(Optional) Provides a description of this interface. The *string* parameter can be any arbitrary text that describes the interface, its neighbor, its purpose, or any other information that might be useful for maintaining and troubleshooting problems with this interface and configuration.

	Command or Action	Purpose

Step 8	**no shutdown**
	
	Example:
	`Router(config-if)# no shutdown`
	`Router(config-if)#`
		Activates the interface and enables it to pass traffic.

Step 9	**interface ge-wan** *slot/port*
	
	Example:
	`Router(config)# interface ge-wan 5/1`
	`Router(config-if)#`
		Enters interface configuration mode for either the specified Gigabit Ethernet WAN interface on the OSM-2+4GE-WAN+ Gigabit Ethernet WAN port.

Step 10	**no ip address**
	
	Example:
	`Router(config-if)# no ip address`
	`Router(config-if)#`
		(Optional) Removes the IP address that might be configured on the interface. This step is required if the interface has been configured previously with an IP address.

Step 11	**channel-group** *number* **mode on**
	
	`Example:`
	`Router(config-if)# channel-group 5 mode on`
	`Router(config-if)#`
		Adds this physical interface to the specified channel group. The *number* should be the same as that specified for the port-channel interface in Step 4.
		Note The **mode on** option is the only one allowed for port-channels that are being configured on GE-WAN interfaces for QinQ link bundling.

Step 12	**no shutdown**
	
	Example:
	`Router(config-if)# no shutdown`
	`Router(config-if)#`
		Activates the interface and enables it to pass traffic.

	Note Repeat Step 9 through Step 12 for each Gigabit Ethernet WAN interface to be added to the port-channel group.	

Step 13	**end**
	
	Example:
	`Router(config-if)# end`
	`Router#`
		Exits interface configuration mode and returns to privileged EXEC mode.

Note If after removing the last inner VLAN in a bridge domain, you want to perform a load rebalancing, issue the **shutdown** and **no shutdown** commands on the port-channel.

Examples

The following example shows a sample configuration for a port-channel interface that has two GE-WAN physical interfaces as part of its channel group:

```
!
interface Port-channel3
 no ip address
 logging event link-status
 speed nonegotiate
```

```
 mode dot1q-in-dot1q access-gateway
!
interface GE-WAN2/1
 no ip address
 logging event link-status
 negotiation auto
 channel-group 3 mode on
!
interface GE-WAN2/3
 no ip address
 logging event link-status
 negotiation auto
 channel-group 3 mode on
```

The following sample configuration shows the error message that appears if you attempt to enable QinQ translation on a port-channel interface that contains one or more invalid interfaces:

```
Router# configure terminal
Router(config)# interface port-channel 30
7600-2(config-if)# mode dot1q-in-dot1q access-gateway

% 'mode dot1q-in-dot1q access-gateway' is not supported on Port-channel30
% Port-channel30 contains 2 Layer 2 Gigabit Ethernet interface(s)

Router(config-if)#
```

To display the status of the port-channel interface, as well as the members of its channel group, use the **show interface** command. For example, this command would show the following output for the configuration listed above.

```
Router# show interface Port-channel 3

 Port-channel1 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 0007.8508.474a (bia 000d.edb5.7d7b)
  MTU 1500 bytes, BW 2000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Full-duplex, Auto-speed
  input flow-control is off, output flow-control is unsupported
  Members in this channel: GE2/1 Pseudo GE2/3 Pseudo
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out

Router#
```

To display the inner, outer, and trunk VLANs that are used in a QinQ translation, use the **show cwan qinq** command. The following examples show the resulting output for the port-channel interface using the **show cwan qinq** command alone and with each of the following optional keywords:

- **configured**—Displays statistics for all configured bridge domains.
- **detail**—Displays the details of the inner VLAN configurations for each bridge domain.
- **list**—Displays the currently configured assignments.

Caution The **show cwan qinq [configured | detail | list]** command applies to port-channel interfaces only. Using this command with physical interfaces may provide incorrect results.

```
Router#show cwan qinq

Bridge-domain   Interface       Egress-if       Inner-start Total Active
3               Po1             GE3/1           0           1     1
  Sub-Interface   Trunk-vlan   Inner-vlan   Service         State
  Po1.2           2            4            dot1q           up/up

Router#show cwan qinq configured

Port-channel1 has total 2 bridge-domain vlan(s)
  Po1 - GE-WAN3/1 has 1 bridge-domain vlan(s) egress configured
        13
  Po1 - GE-WAN3/2 has 1 bridge-domain vlan(s) egress configured
        3

Router#show cwan qinq detail

Port-channel1 has total 2 bridge-domain vlan(s)
  Po1 - GE-WAN3/1 has 1 bridge-domain vlan(s) detail
    Bridge-domain Inner  Configured Active
    ------------- ------ ---------- ------
    13            active 1          1
  Po1 - GE-WAN3/2 has 1 bridge-domain vlan(s) detail
    Bridge-domain Inner  Configured Active
    ------------- ------ ---------- ------
    3             active 1          1

Router#show cwan qinq list

Port-channel1 has total 2 bridge-domain vlan(s)
  Po1 - GE-WAN3/1 has 1 bridge-domain vlan(s) egress active
        13
 Po1 - GE-WAN3/2 has 1 bridge-domain vlan(s) egress active
        3
```

The related **show cwan qinq load-balance** commands also apply to port-channel interfaces only.

```
Router#show cwan qinq load-balance

Port-channel1 has total 2 bridge-domain vlan(s)
  Po1 - GE-WAN3/1 has 1 bridge-domain vlan(s)
  Po1 - GE-WAN3/2 has 1 bridge-domain vlan(s)
```

```
Router#show cwan qinq load-balance detail
Port-channel1 has total 2 bridge-domain vlan(s)
Po1 - GE-WAN3/1 has 1 bridge-domain vlan(s) detail
Bridge-domain Inner Configured Active
------------- ------ ---------- ------
13            active 1          1
Po1 - GE-WAN3/2 has 1 bridge-domain vlan(s) detail
Bridge-domain Inner Configured Active
------------- ------ ---------- ------
3             active 1          1
```

The following related **show** commands can be applied to both port-channel and physical interfaces:

```
Router#show cwan qinq bridge-domain

GE-WAN3/1, group 1, total_rate_active 1
13
GE-WAN3/2, group 1, total_rate_active 1
3
Port-channel1, group 1, total_rate_active 2


Router#show cwan qinq interface

Interface        Status    Egress op PE   CE   TRNK Input packets/       Output packets/
                                                     Input bytes          Output bytes
---------------- --------- ------ -- ---- ---- ---- -------------------- ----------------
Po1.2            up/up     GE3/2  1  3    4    2    0                    0
                                                     0                    0
Po1.12           up/up     GE3/1  1  13   14   12   0                    0
                                                     0                    0
```

Note For additional information regarding these related commands, see the *Cisco 7600 Router Cisco IOS Command Reference—Release 12.2SX*.

Configuring the Service Provider Edge Router

This section describes the procedure to configure the Gigabit Ethernet interface on the service provider edge router that is connected to the Gigabit Ethernet WAN interface that is acting as the IEEE 802.1Q-in-802.1Q (QinQ) access gateway.

Prerequisites

- The service provider edge router must be using a Gigabit Ethernet interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan** *vlan-id*
4. **interface GigabigEthernet** *slot/port*

5. **no ip address**
6. **mls qos trust** [**cos** | **dscp** | **ip-precedence**]
7. **switchport**
8. **switchport trunk encapsulation dot1q**
9. **switch trunk allowed vlan** {*vlan-list* | *vlan-range*}
10. **switchport mode trunk**
11. **description** *string*
12. **no shutdown**
13. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**vlan** *vlan-id*
	
	Example:
	`Router(config)# vlan 22`
	`Router(config)#`
		Add the VLAN ID to be used as the provider edge (PE) VLAN to the router's VLAN database (if not already entered). The valid range for *vlan-id* is either 1 to 1023, or from 1 to 4094, depending on the Cisco IOS software image being used on the router or switch.

Step 4	**interface GigabitEthernet** *slot/port*
	
	Example:
	`Router(config)# interface GigabitEthernet3/1`
	`Router(config-if)#`
		Enters interface configuration mode for the specified Gigabit Ethernet interface.

Step 5	**no ip address**
	
	Example:
	`Router(config-if)# no ip address`
	`Router(config-if)#`
		Removes the IP address that might be configured on the interface.

	Command or Action	Purpose

Step 6	**mls qos trust** [**cos** | **dscp** | **ip-precedence**]
	
	Example:
	`Router(config-if)# mls qos trust dscp`
	`Router(config-if)#`
		(Optional) Specifies which quality of service (QoS) bits in incoming frames can be trusted.
		• **cos**—(Optional) Specifies that the CoS bits in incoming frames are trusted and derives the internal DSCP value from the CoS bits.
		• **dscp**—(Optional, default) Specifies that the ToS bits in the incoming packets contain a DSCP value.
		• **ip-precedence**—(Optional) Specifies that the IP precedence bits (found in the ToS bits) of incoming packets are trusted, and derives the internal DSCP value from the IP precedence bits.
		Note To configure the interface as untrusted, use the **no mls qos trust** command. The interface then zeroes out the P bits of all incoming packets.

Step 7	**switchport**
	
	Example:
	`Router(config-if)# switchport`
	`Router(config-if)#`
		Configures the interface for Layer 2 switching.

Step 8	**switchport trunk encapsulation dot1q**
	
	Example:
	`Router(config-if)# switchport trunk encapsulation dot1q`
	`Router(config-if)#`
		Configures the trunk link to use IEEE 802.1Q encapsulation.

Step 9	**switch trunk allowed vlan** {*vlan-list* | *vlan-range*}
	
	Example:
	`Router(config-if)# switch trunk allowed vlan 3001-4000`
	`Router(config-if)#`
		(Optional) Configures the list of provider edge (PE) VLANs allowed on the trunk. All VLANs are allowed by default. You can either specify a list of individual VLAN IDs separated by commas, or you can specify a range of VLAN IDs separated by a hyphen.

Step 10	**switchport mode trunk**
	
	Example:
	`Router(config-if)# switchport mode trunk`
	`Router(config-if)#`
		Puts the interface into permanent trunking mode.

Step 11	**description** *string*
	
	Example:
	`Router(config-if)# description Connected to Metro interface SJ-3`
	`Router(config-if)#`
		(Optional) Provides a description of this interface. The *string* parameter can be any arbitrary text that describes the interface, its neighbor, its purpose, or any other information that might be useful for maintaining and troubleshooting problems with this interface and configuration.

	Command or Action	Purpose

Step 12	`no shutdown`
	
	Example:
	`Router(config-if)# no shutdown`
	`Router(config-if)#`
		Activates the interface and enables it to pass traffic.

	Note Repeat Step 4 through Step 12 for each interface to be configured.	

Step 13	`end`
	
	Example:
	`Router(config-if)# end`
	`Router#`
		Exits interface configuration mode and returns to privileged EXEC mode.

Examples

The following example shows a sample configuration for a Gigabit Ethernet interface that is connected to the Gigabit Ethernet WAN port that is providing IEEE 802.1Q-in-802.1Q translation. VLAN ID 3001 is being used as the PE VLAN.

```
vlan 3001

...

!
interface GigabitEthernet3/1
 description connected to Metro SJ-3 (QinQ tunnel)
 no ip address
 logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 3001-4000
 switchport mode trunk
```

Configuring QinQ Translation—Double Tag to Single Tag Translation

When you configure the Advanced QinQ Service Mapping feature for QinQ translation, also known as double-tag-to-single-tag translation, the outgoing interface replaces both the inner customer edge (CE) VLAN tag and the outer provider edge (PE) VLAN tag with a Trunk VLAN tag. Use the following procedure to configure a subinterface for double-tag-to-single-tag translation.

Note Cisco IOS Release 12.2(18)SXD used the **bridge-vlan** command to configure the QinQ translation, but Cisco IOS Release 12.2(18)SXE and later releases have changed this to **bridge-domain**. Earlier configurations that use **bridge-vlan** are automatically configured to **bridge-domain** when the configuration is loaded.

Prerequisites

- You must have previously enabled IEEE 802.1Q-in-802.1Q VLAN translation on either a Gigabit Ethernet WAN interface, or on a port-channel interface. See either the "Enabling IEEE 802.1Q-in-802.1Q Translation on a Gigabit Ethernet WAN Interface" section on page 4-11 and the "Enabling IEEE 802.1Q-in-802.1Q Translation on a QinQ Link Bundle" section on page 4-15.

Restrictions

- You can configure a maximum of 32 inner CE VLANs for each outer PE VLAN. The inner CE VLANs must be in a contiguous block that starts on a 32-block boundary (32, 64, and so on), excluding invalid or reserved VLANs.
- You cannot specify an out-of-range configuration for a PE VLAN until you have first configured at least one specific inner CE VLAN ID for that particular PE VLAN. This is required so that the system can determine which VLAN IDs should be considered in-range and out-of-range.
- VLAN 4095 is reserved and cannot be used as a CE VLAN. Packets that contain a CE VLAN ID of 4095 are automatically dropped by subinterfaces that are configured for QinQ translation. VLAN 4095, however, can continue to be used as a native (non-QinQ) VLAN.
- A PE VLAN cannot have the same ID as a native (non-QinQ) VLAN that is also being used on the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** {**ge-wan** *slot/port.subinterface* | **port-channel** *number.subinterface*}
4. **encapsulation dot1q** *trunk-vlan-id*
5. **bridge-domain** *vlan-id* **dot1q** *inner-vlan-id*
   or
   bridge-domain *vlan-id* **dot1q-tunnel out-range**
6. **mls qos trust** [**cos** | **dscp** | **ip-precedence**]
7. **service policy input** *policy-name*
8. **service policy output** *policy-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface** {**ge-wan** *slot/port.subinterface* | **port-channel** *number.subinterface*}
	
	Example:
	`Router(config)# interface ge-wan 5/1.64`
	`Router(config-subif)#`
		Enters subinterface mode for the specified subinterface.

Step 4	**encapsulation dot1q** *trunk-vlan-id*
	
	Example:
	`Router(config-subif)# encapsulation dot1q 2`
	`Router(config-subif)#`
		Configures the subinterface to use the specified IEEE 802.1Q trunk VLAN on outgoing packets:
		• *trunk-vlan-id*—Specifies the trunk VLAN ID to be used for this traffic. The valid range is any VLAN from 1 to 4094, except for the numbers already allocated and the numbers in the range from 1002 to 1005, which are reserved.

Command or Action	Purpose

Step 5 **bridge-domain** *vlan-id* **dot1q** *inner-vlan-id*
or
bridge-domain *vlan-id* **dot1q-tunnel out-range**
 	
Example:
`Router(config-subif)# bridge-domain 2 dot1q 64`
`Router(config-subif)#`
 	
`or`
 	
`Router(config-subif)# bridge-domain 2 dot1q-tunnel out-range`
`Router(config-subif)#`
	Creates a table map for the specified outer (provider) VLAN ID to the specified inner (customer) VLAN ID, specifying that these VLAN tags should be replaced by the trunk VLAN tag when the packet is output.
	• *vlan-id*—VLAN ID for the provider edge (PE), or outer, VLAN. The valid range is 1 to 4094, except for the native VLAN (which defaults to 1) and the numbers in the range from 1002 to 1005, which are reserved. This value must match the VLAN that is actually configured on the provider edge router.
	• **dot1q** *inner-vlan-id*—VLAN ID for the customer edge (CE), or inner, VLAN that is to be mapped to this PE VLAN. The valid range is 1 to 4094, except for the numbers in the range from 1002 to 1005, which are reserved.
	• **dot1q-tunnel out-range**—Creates a table map for all inner (customer) VLAN IDs that are outside of the previously mapped block of 32 VLANs for this particular provider VLAN. If you do not specify an **out-range** mapping for a PE VLAN, the interface drops all packets for that PE VLAN that either do not have a CE VLAN tag, or that have a CE VLAN outside of the mapped block.
	Note You must configure at least one subinterface with a specific CE VLAN ID for a PE VLAN, before you can use the **dot1q-tunnel out-range** option.

Note When you specify the first *inner-vlan-id* for a PE VLAN, the interface automatically associates the correct block of 32 VLANs with that PE VLAN, and those CE VLANs cannot be used for any other purpose. For example, specifying a CE VLAN of 98 associates the VLANs from 96 to 127 with that PE VLAN. Any other CE VLANs received on that PE VLAN are considered out of range.	

Step 6 **mls qos trust** [**cos** | **dscp** | **ip-precedence**]
 	
Example:
`Router(config-subif)# mls qos trust dscp`
`Router(config-subif)#`
	(Optional) Specifies which quality of service (QoS) bits in incoming frames can be trusted.
	• **cos**—(Optional) Specifies that the CoS bits in incoming frames are trusted and derives the internal DSCP value from the CoS bits.
	• **dscp**—(Optional, default) Specifies that the ToS bits in the incoming packets contain a DSCP value.
	• **ip-precedence**—(Optional) Specifies that the IP precedence bits (found in the ToS bits) of incoming packets are trusted, and derives the internal DSCP value from the IP precedence bits.
	Note To configure the interface as untrusted, use the **no mls qos trust** command. The Layer 2 interface then zeroes out the P bits of all incoming packets before any QoS processing is done.

	Command or Action	Purpose

Step 7	**service policy input** *policy-name*
	
	Example:
	`Router(config-subif)# service policy input policy-in1`
	`Router(config-subif)#`
		(Supported only on physical GE-WAN interfaces, not port-channel interfaces) Specifies a policy map that should be used on incoming packets when they are received on the Gigabit Ethernet WAN interface.

Step 8	**service policy output** *policy-name*
	
	Example:
	`Router(config-subif)# service policy output cos-xlat1`
	`Router(config-subif)#`
		Specifies a policy map that should be used on outgoing packets before they leave the Gigabit Ethernet WAN interface.
		
		Note Policy maps that use **set cos cos-inner** command must be applied as the output policy on the subinterface.

	Note Repeat Step 3 through Step 8 for each subinterface/VLAN mapping to be configured.	

Step 9	**end**
	
	Example:
	`Router(config)# end`
	`Router#`
		Exits global configuration mode and returns to privileged EXEC mode.

Examples

The following shows a typical configuration that creates two double-tag-to-single-tag mappings on a subinterface. The first subinterface configuration creates a specific PE/CE mapping, and the second subinterface configuration creates an out-of-range configuration:

```
interface GE-WAN 3/3
 no ip address
 mode dot1q-indot1q access-gateway
...
!
interface GE-WAN3/3.42
 encapsulation dot1Q 2
 bridge-domain 133 dot1q 42
 mls qos trust dscp
end
...
!
interface GE-WAN3/3.5032
 encapsulation dot1Q 31
 bridge-domain 133 dot1q-tunnel out-range
 mls qos trust dscp
end
```

These QinQ mappings operate as follows:

- The first subinterface matches incoming packets that are tagged with a PE VLAN ID of 133 and a CE VLAN ID of 42, and translates those packets into an outgoing packet with a single trunk VLAN ID of 2. This configuration also automatically associates the block of CE VLANs from 32 to 63 with PE VLAN 133. Any packets with a CE VLAN ID in that range that also have a PE VLAN ID of 133, and are not explicitly mapped by another subinterface, are dropped. Any other CE VLANs that are received on PE VLAN 133 are considered out of range.

- The second subinterface matches incoming packets that are tagged with a PE VLAN ID of 133, and that either do not have a CE VLAN, or that have a CE VLAN ID that is out of range (that is ranging from 1 to 31 or from 64 to 4094). These packets are translated into an outgoing packet with a trunk VLAN ID of 31 as the outer tag and an unchanged CE VLAN inner tag (if present).

This configuration performs the following mapping on packets that have a PE VLAN ID of 133:

Table 1-1 Example Double-Tag-to-Single-Tag Mappings

PE VLAN ID	CE VLAN ID	Action
133	1 to 31	Mapped to trunk VLAN 31, CE VLAN 1 to 31 (out of range)
133	32 to 41	Dropped (because not explicitly mapped)
133	42	Mapped to trunk VLAN 2 (explicitly mapped by GE-WAN3/3.42)
133	43 to 63	Dropped (because not explicitly mapped)
133	64 to 4094	Mapped to trunk VLAN 31, CE VLAN 64 to 4094 (out of range)
133	(none)	Mapped to trunk VLAN 31 (out of range)

Configuring QinQ Transparent Tunneling—Double Tag to Double Tag Translation

When you configure the Advanced QinQ Service Mapping feature for QinQ transparent tunneling, as known as double-tag-to-double-tag translation, the Gigabit Ethernet WAN interface replaces the outer (provider edge or PE) VLAN tag with the trunk VLAN tag. The inner CE VLAN tag (if present) remains unchanged. Use the following procedure to configure a subinterface for double-tag-to-double-tag translation.

Note Cisco IOS Release 12.2(18)SXD used the **bridge-vlan** command to configure the QinQ translation, but Cisco IOS Release 12.2(18)SXE and later releases have changed this to **bridge-domain**. Earlier configurations that use **bridge-vlan** are automatically configured to **bridge-domain** when the configuration is loaded.

Prerequisites

- You must have previously enabled IEEE 802.1Q-in-802.1Q VLAN translation on either a Gigabit Ethernet WAN interface, or on a port-channel interface. See either the "Enabling IEEE 802.1Q-in-802.1Q Translation on a Gigabit Ethernet WAN Interface" section on page 4-11 and the "Enabling IEEE 802.1Q-in-802.1Q Translation on a QinQ Link Bundle" section on page 4-15.

Restrictions

- You can configure a maximum of 32 inner CE VLANs for each outer PE VLAN. The inner VLANs must be in a contiguous block that starts on a 32-block boundary (0, 32, 64, and so on).
- VLAN 4095 is reserved and cannot be used as a CE VLAN. Packets that contain a CE VLAN ID of 4095 are automatically dropped by subinterfaces that are configured for QinQ translation. VLAN 4095, however, can continue to be used as a native (non-QinQ) VLAN.
- You cannot specify an out-of-range configuration for a PE VLAN until you have first configured at least one specific inner CE VLAN ID for that particular PE VLAN. This is required so that the system can determine which VLAN IDs are considered in-range or out-of-range.

- A PE VLAN cannot have the same ID as a native (non-QinQ) VLAN that is also being used on the router.
- Input service policies (the **service-policy input** command) are not supported on port-channels being used for QinQ link bundling.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** {**ge-wan** *slot/port.subinterface* | **port-channel** *number.subinterface*}
4. **encapsulation dot1q** *trunk-vlan-id*
5. **bridge-domain** *vlan-id* **dot1q-tunnel** {*inner-vlan-id* | **out-range**}
6. **mls qos trust** [**cos** | **dscp** | **ip-precedence**]
7. **service policy input** *policy-name*
8. **service policy output** *policy-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface** {**ge-wan** *slot/port.subinterface* | **port-channel** *number.subinterface*}
	
	Example:
	`Router(config)# interface ge-wan 5/1.64`
	`Router(config-subif)#`
		Enters subinterface mode for the specified subinterface.

Step 4	**encapsulation dot1q** *trunk-vlan-id*
	
	Example:
	`Router(config-subif)# encapsulation dot1q 2`
	`Router(config-subif)#`
		Configures the subinterface to use the specified IEEE 802.1Q trunk VLAN on outgoing packets:
		• *trunk-vlan-id*—Specifies the trunk VLAN ID to be used for this traffic. The valid range is any VLAN from 1 to 4094, except for the numbers already allocated and the numbers in the range from 1002 to 1005, which are reserved.

	Command or Action	Purpose

Step 5	**bridge-domain** *vlan-id* **dot1q-tunnel** {*inner-vlan-id* | **out-range**}
	
	Example:
	`Router(config-subif)# bridge-domain 2 dot1q 64`
	`Router(config-subif)#`
	
	`or`
	
	`Router(config-subif)# bridge-domain 2 dot1q out-range`
	`Router(config-subif)#`
		Creates a table map for the specified outer (provider) VLAN ID to the specified inner (customer) VLAN ID, specifying that the outer VLAN tag should be replaced by the trunk VLAN tag when the packet is output (leaving the inner tag unchanged):
		• *vlan-id*—VLAN ID for the provider edge (PE), or outer, VLAN. The valid range is 1 to 4094, except for the native VLAN (which defaults to 1) and the numbers in the range from 1002 to 1005, which are reserved. This value must match the VLAN that is actually configured on the provider edge router.
		• *inner-vlan-id*—VLAN ID for the customer edge (CE), or inner, VLAN that is to be mapped to this PE VLAN. The valid range is 1 to 4094, except for the numbers in the range from 1002 to 1005, which are reserved.
		• **out-range**—Matches all inner VLAN IDs that are outside of the previously mapped block of 32 VLANs for this particular provider VLAN. If you do not specify an **out-range** mapping for a PE VLAN, the interface drops all packets for that PE VLAN with a CE VLAN outside of the mapped block.
		Note You must configure at least one subinterface for a specific CE VLAN ID for a PE VLAN, before you can use the **out-range** command.

	Note When you specify the first *inner-vlan-id* for a PE VLAN, the interface automatically associates the correct block of 32 VLANs with that PE VLAN, and those CE VLANs cannot be used for any other purpose. For example, specifying a CE VLAN of 98 associates the VLANs from 96 to 127 with that PE VLAN. Any other CE VLANs received on that PE VLAN are considered out of range.	

Step 6	**mls qos trust** [**cos** | **dscp** | **ip-precedence**]
	
	Example:
	`Router(config-subif)# mls qos trust dscp`
	`Router(config-subif)#`
		(Optional) Specifies which quality of service (QoS) bits in incoming frames can be trusted.
		• **cos**—(Optional) Specifies that the CoS bits in incoming frames are trusted and derives the internal DSCP value from the CoS bits.
		• **dscp**—(Optional, default) Specifies that the ToS bits in the incoming packets contain a DSCP value.
		• **ip-precedence**—(Optional) Specifies that the IP precedence bits (found in the ToS bits) of incoming packets are trusted, and derives the internal DSCP value from the IP precedence bits.
		Note To configure the interface as untrusted, use the **no mls qos trust** command. The Layer 2 interface then zeroes out the P bits of all incoming packets before any QoS processing is done.

	Command or Action	Purpose

Step 7	**service policy input** *policy-name*
	
	Example:
	`Router(config-subif)# service policy input policy-in1`
	`Router(config-subif)#`
		(Supported only on physical GE-WAN interfaces, not port-channel interfaces) Specifies a policy map that should be used on incoming packets when they are received on the Gigabit Ethernet WAN interface.

Step 8	**service policy output** *policy-name*
	
	Example:
	`Router(config-subif)# service policy output cos-xlat1`
	`Router(config-subif)#`
		Specifies a policy map that should be used on outgoing packets before they leave the Gigabit Ethernet WAN interface.

	Note Repeat Step 3 through Step 8 for each subinterface/VLAN mapping to be configured.	

Step 9	**end**
	
	Example:
	`Router(config)# end`
	`Router#`
		Exits global configuration mode and returns to privileged EXEC mode.

Examples

The following shows a typical configuration that creates two double-tag-to-double-tag mappings on a subinterface. The first subinterface configuration creates a specific PE/CE mapping, and the second subinterface configuration creates an out-of-range configuration:

```
!
interface GE-WAN1/1.98
 encapsulation dot1Q 12
 bridge-domain 65 dot1q-tunnel 98
 mls qos trust dscp
end
...
!
interface GE-WAN1/1.5096
 encapsulation dot1Q 31
 bridge-domain 65 dot1q-tunnel out-range
 mls qos trust dscp
end
```

These QinQ mappings operate as follows:

- The first subinterface matches incoming packets that are tagged with a PE VLAN ID of 65 and a CE VLAN ID of 98, and translates those packets into an outgoing packet with a trunk VLAN ID of 12 and a CE VLAN ID of 98. This configuration also automatically associates the block of CE VLANs from 96 to 127 with PE VLAN 65. Any packets with a CE VLAN ID in that range that also have a PE VLAN ID of 65, and are not explicitly mapped by another subinterface, are dropped. Any other CE VLANs that are received on PE VLAN 65 are considered out of range.
- The second subinterface matches incoming packets that are tagged with a PE VLAN ID of 65, and that either do not have a CE VLAN tag, or that have a CE VLAN ID that is out of range (that is ranging from 1 to 95 or from 128 to 4094). These packets are translated to an outgoing packet that has a trunk VLAN ID of 31 and an unchanged CE VLAN tag (if present).

This configuration performs the following mapping on packets that have a PE VLAN ID of 65:

Table 1-2 Example Double-Tag-to-Double-Tag Mappings

PE VLAN ID	CE VLAN ID	Action
65	1 to 95	Mapped to trunk VLAN 31, CE VLAN 1 to 31 (out of range)
65	96 to 97	Dropped (because not explicitly mapped)
65	98	Mapped to trunk VLAN 12, CE VLAN 98 (explicitly mapped by GE-WAN3/3.42)
65	99 to 127	Dropped (because not explicitly mapped)
65	128 to 4094	Mapped to trunk VLAN 31, CE VLAN 128 to 4094 (out of range)
65	(none)	Mapped to trunk VLAN 31 (out of range)

Configuring a Policy Map to Use the Inner COS Bits

By default, the IEEE 802.1Q-to-IEEE 802.1Q translation sets the IEEE 802.1P bits (P bits) in the IEEE 802.1Q header of the outgoing packet's trunk VLAN tag by copying the P bits from the outer PE VLAN tag. To change this behavior, create a policy map with a class map that contains the **set cos cos-inner** command. The system then copies the P bits from the inner CE VLAN tag to the trunk VLAN tag that is put on the outgoing packet.

Prerequisites

- After creating the policy map, you must apply it to the appropriate VLAN's subinterface by using the **service-policy output** command in subinterface configuration mode. See the following sections for more details:
  - Configuring QinQ Translation—Double Tag to Single Tag Translation, page 4-24
  - Configuring QinQ Transparent Tunneling—Double Tag to Double Tag Translation, page 4-29

Restrictions

- The **set cos cos-inner** command is supported only for subinterfaces that are configured with an inner CE VLAN. The **set cos cos-inner** command is not supported on subinterfaces that use the **out-range** option with the **bridge-domain** command.
- You cannot use these policy maps on a main Gigabit Ethernet WAN interface or on a main port-channel interface.
- For the **set cos cos-inner** command to have any effect, you must configure an interface or subinterface to be a trusted interface, using the **mls qos trust** command. Otherwise, if the interface or subinterface is untrusted, the interface zeroes out the 802.1P bits of incoming packets before the bits can be copied to the outgoing packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*

4. **class** {*class-name* | **class-default**}
5. **set cos cos-inner**
6. **shape** {**average** | **peak**} *mean-rate* [*bc* [*be*]]
7. (other configuration commands as desired)
8. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**policy-map** *policy-map-name*
	
	Example:
	`Router(config)# policy-map pmap1`
	`Router(config-pmap)#`
		Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters policy-map configuration mode.
		• *policy-map-name*—Name of the policy map. The name can be a maximum of 40 alphanumeric characters.

Step 4	**class** {*class-name* | **class-default**}
	
	Example:
	`Router(config-pmap)# class class-default`
	`Router(config-pmap-c)#`
		Creates or modifies a policy class, and enters policy map class configuration mode.
		• *class-name*—Name of the class to be configured or modified.
		• **class-default**—Specifies the default class that should be used when no other class has been specified.

Step 5	**set cos cos-inner**
	
	Example:
	`Router(config-pmap-c)# set cos cos-inner`
	`Router(config-pmap-c)#`
		(Optional) Sets the IEEE 802.1 prioritization bits (P bits) of the trunk VLAN tag of an IEEE 802.1Q-in-802.1Q translated outgoing packet with the priority value from the incoming packet's inner (customer edge) VLAN tag. The default value is the **no** form of this command, which uses the P bits from the incoming packet's outer (provider edge) VLAN tag.

	Command or Action	Purpose

Step 6	**shape** {**average** | **peak**} *mean-rate* [*bc* [*be*]]
	
	Example:
	`Router(config-pmap-c)# shape average 4000000 16000 16000`
	`Router(config-pmap-c)#`
		(Optional) Specifies the traffic shaping rates to be used with this policy:
		• **average**—(Optional) Maximum number of bits sent out in each interval is equal to the committed burst size (Bc).
		• **peak**—(Optional) Specifies that the maximum number of bits sent out in each interval is equal to the burst size (Bc) plus the excess burst size (Be).
		• **mean-rate**—(Optional) Also called committed information rate (CIR). Indicates the bit rate used to shape the traffic, in bits per second.
		• **bc**—(Optional) The number of bits in a measurement interval burst size (Bc).
		• **be**—(Optional) The number of bits permitted to go over the excess burst size (Be).

Step 7	**end**
	
	Example:
	`Router(config-pmap-c)# end`
	`Router#`
		Exits policy-map class configuration mode and returns to privileged EXEC mode.

Examples

The following example shows a typical policy map configuration using the **set cos cos-inner** command:

```
!
policy-map pmap1
  class class-default
    shape average 4000000
    set cos cos-inner
```

Disabling IEEE 802.1Q-in-802.1Q Mapping and Translation

To disable the mapping and translation of IEEE 802.1Q-in-802.1Q double-tagged packets on the Gigabit Ethernet interface or on one of its subinterfaces, use one of the following procedures:

- Disabling All IEEE 802.1Q-to-802.1Q Translation on An Interface, page 4-35
- Disabling IEEE 802.1Q-to-802.1Q Translation on One Subinterface, page 4-37

Disabling All IEEE 802.1Q-to-802.1Q Translation on An Interface

To disable all IEEE 802.1Q-to-802.1Q translation on a Gigabit Ethernet WAN interface or a port-channel interface, use the following procedure. This procedure also removes all subinterfaces and their configurations from the interface, which then allows the associated VLANs to be used for other purposes or on other cards.

Tip Be sure to save the configuration before you begin this procedure if you want to move the configuration to another interface.

Note Removing the interface card from the router does not remove the interface configuration, because the Cisco IOS software assumes you will be performing an online insertion and removal (OIR) operation. You must disable IEEE 802.1Q-in-802.1Q translation from all interfaces on a card before removing the card from the chassis, before the VLANs that are configured on the card can become available for use by other interfaces.

Prerequisites

If you have previously attached a service policy that contains a **set cos cos-inner** command to the interface, you must first remove that service policy before you can use the **no mode dot1q-in-dot1q access-gateway** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** {**ge-wan** *slot/port* | **port-channel** *number*}
4. **shutdown**
5. **no mode dot1q-in-dot1q access-gateway**
6. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface** {**ge-wan** *slot/port* | **port-channel** *number*}
	
	Example:
	`Router(config)# interface ge-wan 5/1`
	`Router(config-if)#`
		Enters interface configuration mode for the specified Gigabit Ethernet WAN interface or port-channel interface.

Step 4	**shutdown**
	
	Example:
	`Router(config-if)# shutdown`
	`Router(config-if)#`
		(Optional) Disables the interface and prevents it from passing traffic.

	Command or Action	Purpose

Step 5	`no mode dot1q-in-dot1q access-gateway`
	
	Example:
	`Router(config-if)# no mode dot1q-in-dot1q access-gateway`
	`Router(config-if)#`
		Disables IEEE 802.1Q-in-802.1Q translation on the interface. This disables the Advanced QinQ Service Mapping feature, and removes all subinterface configuration from the interface.
		
		Note Be sure to save the configuration before giving this command if you plan to move the configuration to another interface.

Step 6	`end`
	
	Example:
	`Router(config-if)# end`
	`Router#`
		Exits global configuration mode and returns to privileged EXEC mode.

Disabling IEEE 802.1Q-to-802.1Q Translation on One Subinterface

Use the following procedure to disable IEEE 802.1Q-to-802.1Q translation on an individual subinterface. You can either completely delete the subinterface, or you can remove just the **bridge-domain** configuration on the subinterface, depending on whether you want to use the subinterface to continue passing other traffic. Both methods release the CE and PE VLANs being used on the subinterface.

Prerequisites

If you have previously attached a service policy that contains a **set cos cos-inner** command to the interface, you must first remove that service policy before you can use the **no bridge-domain** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no interface** {**ge-wan** *slot/port.subinterface* | **port-channel** *number.subinterface*}

or

4. **interface** {**ge-wan** *slot/port.subinterface* | **port-channel** *number.subinterface*}
5. **no bridge-domain** *vlan-id* **dot1q** {*inner-vlan-id* | **out-range**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**no interface** {**ge-wan** *slot/port.subinterface* | **port-channel** *number.subinterface*}
	
	`Example:`
	`Router(config)# no interface ge-wan 5/1.64`
	`Router(config-subif)#`
		Completely removes the subinterface and its configuration. All traffic passing through this interface stops.
		Note After entering this command, proceed to Step 6.

or		

Step 4	**interface** {**ge-wan** *slot/port.subinterface* | **port-channel** *number.subinterface*}
	
	Example:
	`Router(config)# interface ge-wan 5/1.64`
	`Router(config-subif)#`
		Enters subinterface mode for the specified subinterface.

Step 5	**no bridge-domain** *vlan-id* **dot1q** {*inner-vlan-id* | **out-range**}
	
	Example:
	`Router(config-subif)# no bridge-domain 2 dot1q 64`
	`Router(config-subif)#`
		Removes the table mapping for this subinterface, disabling the IEEE 802.1Q-to-IEEE802.1Q translation for this particular combination of VLANs. Traffic continues to pass, depending on the remaining configuration of the subinterface.

Step 6	**end**
	
	Example:
	`Router(config-subif)# end`
	`Router#`
		Exits subinterface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Advanced QinQ Service Mapping

This section provides the following sample configurations:

- QinQ Translation Configuration Example—Two-Tag to One-Tag Translation, page 4-39
- QinQ Transparent Tunneling Configuration Example, page 4-42
- QinQ Translation Using Port-Channel Interfaces Example, page 4-45

QinQ Translation Configuration Example—Two-Tag to One-Tag Translation

The following excerpt from a configuration file shows the configuration for a simple QinQ translation, in which incoming packets are received with inner customer edge (CE) and outer provider edge (PE) VLAN tags. The packets are then output, using the configured policy map, with a single trunk VLAN tag.

This configuration configures Gigabit Ethernet WAN interface 4/1 as the QinQ access gateway, and shows two PE-to-CE mappings:

- The first set of subinterfaces is configured for a PE VLAN ID of 2 and CE VLAN IDs in the range of 32 to 46. These subinterfaces are all configured as trusted (**mls qos trust dscp**) and use policy maps that use the **set cos cos-inner** command, so that the 802.1P bits in the customer's original CE VLAN tag are copied to the outgoing trunk VLAN tag.

  Subinterface 47 is configured to match any packets that arrive with a PE VLAN ID of 2 and an out-of-range CE VLAN ID (between 47 and 63). Note that the **set cos cos-inner** command has no effect on out-of-range packets, even when using a policy map that includes this command.

- The second set of subinterfaces is configured for a trunk VLAN ID of 100 and a PE VLAN ID of 45. These subinterfaces accept incoming CE VLAN IDs in the range of 1237 to 1240. This configuration does not include an **out-of-range** subinterface, so any packets that arrive with a PE VLAN ID of 45 and an out-of-range CE VLAN ID (from 1216 to 1236 and from 1241 to 1247) are dropped. All subinterfaces use a policy map that does not include the **set cos cos-inner** command, which means that the trunk VLAN tag uses the 802.1P bits in the original PE VLAN tag.

```
!
vlan internal allocation policy descending
!
vlan 1-1240
!
policy-map pmap1
  class class-default
    shape average 4000000
   set cos cos-inner
policy-map pmap2
  class class-default
    shape average 8000000 32000 32000
   set cos cos-inner
policy-map pmap3
  class class-default
    shape average 20000000 80000 80000
   set cos cos-inner
policy-map pmap4
  class class-default
    shape average 2000000 16000 16000
!
!
interface GigabitEthernet4/1
 description connected to SP GE1/3
 no ip address
 logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface GigabitEthernet4/2
 no ip address
 shutdown
!
!--This is the QinQ Access Gateway interface
interface GE-WAN4/1
 description connected to PE-4 GigabitEthernet0/3
```

```
 no ip address
 logging event link-status
 negotiation auto
 mode dot1q-in-dot1q access-gateway
!--This command configures the interface as trusted, which
!--is required to be able to use the original packet's 802.1P CoS bits.
 mls qos trust dscp

!--First set of PE/CE mappings
!
interface GE-WAN4/1.32
 encapsulation dot1Q 32
!--note that this bridge-domain command automatically configures the
!--CE VLAN range for this PE VLAN to be from 32 to 63
 bridge-domain 2 dot1q 32
 mls qos trust dscp
 service-policy output pmap3
!
interface GE-WAN4/1.33
 encapsulation dot1Q 33
 bridge-domain 2 dot1q 33
 mls qos trust dscp
 service-policy output pmap2
!
interface GE-WAN4/1.34
 encapsulation dot1Q 34
 bridge-domain 2 dot1q 34
 mls qos trust dscp
 service-policy output pmap1
!
interface GE-WAN4/1.35
 encapsulation dot1Q 35
 bridge-domain 2 dot1q 35
 mls qos trust dscp
 service-policy output pmap2
!
interface GE-WAN4/1.36
 encapsulation dot1Q 36
 bridge-domain 2 dot1q 36
 mls qos trust dscp
 service-policy output pmap3
!
interface GE-WAN4/1.37
 encapsulation dot1Q 37
 bridge-domain 2 dot1q 37
 mls qos trust dscp
 service-policy output pmap1
!
interface GE-WAN4/1.38
 encapsulation dot1Q 38
 bridge-domain 2 dot1q 38
 mls qos trust dscp
 service-policy output pmap1
!
interface GE-WAN4/1.39
 encapsulation dot1Q 39
 bridge-domain 2 dot1q 39
 mls qos trust dscp
 service-policy output pmap2
!
interface GE-WAN4/1.40
 encapsulation dot1Q 40
 bridge-domain 2 dot1q 40
 mls qos trust dscp
```

```
 service-policy output pmap3
!
interface GE-WAN4/1.41
 encapsulation dot1Q 41
 bridge-domain 2 dot1q 41
 mls qos trust dscp
 service-policy output pmap2
!
interface GE-WAN4/1.42
 encapsulation dot1Q 42
 bridge-domain 2 dot1q 42
 mls qos trust dscp
 service-policy output pmap1
!
interface GE-WAN4/1.43
 encapsulation dot1Q 43
 bridge-domain 2 dot1q 43
 mls qos trust dscp
 service-policy output pmap2
!
interface GE-WAN4/1.44
 encapsulation dot1Q 44
 bridge-domain 2 dot1q 44
 mls qos trust dscp
 service-policy output pmap3
!
interface GE-WAN4/1.45
 encapsulation dot1Q 45
 bridge-domain 2 dot1q 45
 mls qos trust dscp
 service-policy output pmap3
!
interface GE-WAN4/1.46
 encapsulation dot1Q 46
 bridge-domain 2 dot1q 46
 mls qos trust dscp
 service-policy output pmap1
!
interface GE-WAN4/1.47
 description out-of-range configuration for CE VLANs 47 to 63
 encapsulation dot1Q 47
 bridge-domain 2 dot1q-tunnel out-range
 mls qos trust dscp
!-- Although this policy map includes the set cos cos-inner command,
!-- this command is not used for out-of-range packets
 service-policy output pmap4

!--Second set of PE/CE mappings
!
interface GE-WAN4/1.1237
 encapsulation dot1Q 1237
!--note that this bridge-domain command automatically configures the
!--CE VLAN range for this PE VLAN to be from 1216 to 1247
 bridge-domain 45 dot1q 1237
 no mls qos trust
 service-policy output pmap4
!
interface GE-WAN4/1.1238
 encapsulation dot1Q 1238
 bridge-domain 45 dot1q 1238
 no mls qos trust
 service-policy output pmap4
!
interface GE-WAN4/1.1239
```

```
 encapsulation dot1Q 1239
 bridge-domain 45 dot1q 1239
 no mls qos trust
 service-policy output pmap4
!
interface GE-WAN4/1.1240
 encapsulation dot1Q 1240
 bridge-domain 45 dot1q 1240
 no mls qos trust
 service-policy output pmap4

...
```

QinQ Transparent Tunneling Configuration Example

The following excerpt from a configuration file shows a typical configuration for a simple QinQ transparent tunneling configuration, in which incoming packets are received with inner customer edge (CE) and outer provider edge (PE) VLAN tags. The packets are then output, using the configured policy map, with a new trunk VLAN tag and the original inner CE VLAN tag. This configuration is called two-tag to one-tag translation.

This configuration configures Gigabit Ethernet WAN interface 4/1 as the QinQ access gateway, and creates a PE/CE mapping with the following characteristics:

- PE VLAN ID of 152.
- CE VLAN IDs in the range from 2048 to 2079.
- Subinterface GE-WAN 4/1.15233 matches any packets that contain CE VLAN IDs that are outside of this range (either from 1 to 2047 or from 2080 to 4094).
- The interface and all subinterfaces, except for the out-of-range subinterface, are configured as trusted (**mls qos trust dscp**), which allows them to copy the 802.1P bits in the packet's original PE VLAN tag to the outgoing trunk VLAN tag. (The original CE VLAN tag is unchanged and includes its original 802.1P bits.)

```
!
vlan internal allocation policy descending
!
vlan 1-4094

...

!--This is an IP LAN interface
interface GigabitEthernet4/1
 description QinQ tunnel to Catalyst 3550 Gigabit Ethernet 0/6
 no ip address
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 340
 switchport mode trunk
!
!
interface GigabitEthernet4/2
 no ip address
 shutdown
!
!--This is the QinQ Access Gateway interface
interface GE-WAN4/1
 description connected to GSR Gigabit Ethernet 4/1
 no ip address
 logging event link-status
```

```
 no negotiation auto
 mode dot1q-in-dot1q access-gateway
 mls qos trust dscp
!
interface GE-WAN4/1.15201
 encapsulation dot1Q 180
!--note that this bridge-domain command automatically configures the
!--CE VLAN range for this PE VLAN to be from 2048 to 2079
 bridge-domain 152 dot1q-tunnel 2048
 mls qos trust dscp
!
interface GE-WAN4/1.15203
 encapsulation dot1Q 182
 bridge-domain 152 dot1q-tunnel 2049
 mls qos trust dscp
!
interface GE-WAN4/1.15204
 encapsulation dot1Q 183
 bridge-domain 152 dot1q-tunnel 2050
 mls qos trust dscp
!
interface GE-WAN4/1.15205
 encapsulation dot1Q 184
 bridge-domain 152 dot1q-tunnel 2051
 mls qos trust dscp
!
interface GE-WAN4/1.15206
 encapsulation dot1Q 185
 bridge-domain 152 dot1q-tunnel 2052
 mls qos trust dscp
!
interface GE-WAN4/1.15207
 encapsulation dot1Q 186
 bridge-domain 152 dot1q-tunnel 2053
 mls qos trust dscp
!
interface GE-WAN4/1.15208
 encapsulation dot1Q 187
 bridge-domain 152 dot1q-tunnel 2054
 mls qos trust dscp
!
interface GE-WAN4/1.15209
 encapsulation dot1Q 188
 bridge-domain 152 dot1q-tunnel 2055
 mls qos trust dscp
!
interface GE-WAN4/1.15210
 encapsulation dot1Q 189
 bridge-domain 152 dot1q-tunnel 2056
 mls qos trust dscp
!
interface GE-WAN4/1.15211
 encapsulation dot1Q 190
 bridge-domain 152 dot1q-tunnel 2057
 mls qos trust dscp
!
interface GE-WAN4/1.15212
 encapsulation dot1Q 191
 bridge-domain 152 dot1q-tunnel 2058
 mls qos trust dscp
!
interface GE-WAN4/1.15213
 encapsulation dot1Q 192
 bridge-domain 152 dot1q-tunnel 2059
```

```
 mls qos trust dscp
!
interface GE-WAN4/1.15214
 encapsulation dot1Q 193
 bridge-domain 152 dot1q-tunnel 2060
 mls qos trust dscp
!
interface GE-WAN4/1.15215
 encapsulation dot1Q 194
 bridge-domain 152 dot1q-tunnel 2061
 mls qos trust dscp
!
interface GE-WAN4/1.15216
 encapsulation dot1Q 195
 bridge-domain 152 dot1q-tunnel 2062
 mls qos trust dscp
!
interface GE-WAN4/1.15217
 encapsulation dot1Q 196
 bridge-domain 152 dot1q-tunnel 2063
 mls qos trust dscp
!
interface GE-WAN4/1.15218
 encapsulation dot1Q 197
 bridge-domain 152 dot1q-tunnel 2064
 mls qos trust dscp
!
interface GE-WAN4/1.15219
 encapsulation dot1Q 198
 bridge-domain 152 dot1q-tunnel 2065
 mls qos trust dscp
!
interface GE-WAN4/1.15220
 encapsulation dot1Q 199
 bridge-domain 152 dot1q-tunnel 2066
 mls qos trust dscp
!
interface GE-WAN4/1.15221
 encapsulation dot1Q 200
 bridge-domain 152 dot1q-tunnel 2067
 mls qos trust dscp
!
interface GE-WAN4/1.15222
 encapsulation dot1Q 201
 bridge-domain 152 dot1q-tunnel 2068
 mls qos trust dscp
!
interface GE-WAN4/1.15223
 encapsulation dot1Q 202
 bridge-domain 152 dot1q-tunnel 2069
 mls qos trust dscp
!
interface GE-WAN4/1.15224
 encapsulation dot1Q 203
 bridge-domain 152 dot1q-tunnel 2070
 mls qos trust dscp
!
interface GE-WAN4/1.15225
 encapsulation dot1Q 204
 bridge-domain 152 dot1q-tunnel 2071
 mls qos trust dscp
!
interface GE-WAN4/1.15226
 encapsulation dot1Q 205
```

```
 bridge-domain 152 dot1q-tunnel 2072
 mls qos trust dscp
!
interface GE-WAN4/1.15227
 encapsulation dot1Q 206
 bridge-domain 152 dot1q-tunnel 2073
 mls qos trust dscp
!
interface GE-WAN4/1.15228
 encapsulation dot1Q 207
 bridge-domain 152 dot1q-tunnel 2074
 mls qos trust dscp
!
interface GE-WAN4/1.15229
 encapsulation dot1Q 208
 bridge-domain 152 dot1q-tunnel 2075
 mls qos trust dscp
!
interface GE-WAN4/1.15230
 encapsulation dot1Q 209
 bridge-domain 152 dot1q-tunnel 2076
 mls qos trust dscp
!
interface GE-WAN4/1.15231
 encapsulation dot1Q 210
 bridge-domain 152 dot1q-tunnel 2077
 mls qos trust dscp
!
interface GE-WAN4/1.15232
 encapsulation dot1Q 211
 bridge-domain 152 dot1q-tunnel 2078
 mls qos trust dscp
!
! This creates an out-of-range configuration that matches CE VLANs
! that are out of the configured CE VLAN range of 2048 to 2079
interface GE-WAN4/1.15233
 encapsulation dot1Q 212
 bridge-domain 152 dot1q-tunnel out-range
 no mls qos trust
!
...
```

QinQ Translation Using Port-Channel Interfaces Example

The following shows a sample configuration of a QinQ link bundle that contains two GE-WAN physical interfaces. Note that the **bridge-domain** commands are configured on the subinterfaces of the port-channel virtual interface.

```
vlan internal allocation policy ascending
!
vlan 1, 100-1000, 2976-3008
!
policy-map pmap4
  class class-default
   set cos cos-inner
policy-map pmap1
  class class-default
    shape average 4000000
   set cos cos-inner
policy-map pmap2
  class class-default
```

```
    shape average 8000000 32000 32000
policy-map pmap3
  class class-default
    shape average 20000000 80000 80000
!
!
interface Port-channel1
 no ip address
 logging event link-status
 mode dot1q-in-dot1q access-gateway
!
interface Port-channel1.101
 encapsulation dot1Q 101
 bridge-domain 101 dot1q 101
 service-policy output pmap1
!
interface Port-channel1.102
 encapsulation dot1Q 102
 bridge-domain 102 dot1q 102
 service-policy output pmap2
!
interface Port-channel1.103
  encapsulation dot1Q 103
  bridge-domain 103 dot1q 103
!
interface Port-channel1.104
  encapsulation dot1Q 104
  bridge-domain 104 dot1q 104
!
interface Port-channel1.201
  encapsulation dot1Q 201
  bridge-domain 201 dot1q 201
!
!
! GigabitEthernet interfaces are not used for QinQ
!  link bundling, but can be used for
!  other purposes
interface GigabitEthernet4/1
 no ip address
 shutdown
!
interface GigabitEthernet4/2
 no ip address
 shutdown
!
interface GE-WAN4/1
 no ip address
 logging event link-status
 negotiation auto
 mls qos trust dscp
 channel-group 1 mode on
!
interface GE-WAN4/2
 no ip address
 logging event link-status
 negotiation auto
 mls qos trust dscp
 channel-group 1 mode on
!
```

```
interface GE-WAN4/3
 no ip address
 shutdown
!
interface GE-WAN4/4
 no ip address
 shutdown
...
```

CHAPTER 1

Configuring the Channelized OC-12/T3 SONET/SDH Optical Services Modules

This chapter describes how to configure the channelized 1-port OC-12 (OSM-1CHOC12/T3-SI) SONET/SDH Optical Services Modules (OSMs).

The chapter consists of these sections:

Understanding the Channelized OSMs

These sections describe the SONET/SDH mappings and multiplex hierachy and the features supported on the channelized OSMs:

Supported Multiplexing and Mappings

The OSM-1CHOC12/STM-4 T3-SI module supports channelized configurations down to OC-3, DS-3, and DS-3 subrate.

Figure 5-1 shows the SONET multiplexing hierarchy supported by the 1-port ChOC-12/STM-4 OSMs.

Figure 5-2 shows the SDH multiplexing hierarchy supported by the 1-ChOC-12/STM-4 OSMs.

Figure 1-1 Supported SONET Multiplexing Hierarchy on the 1-port ChOC-12 OSMs

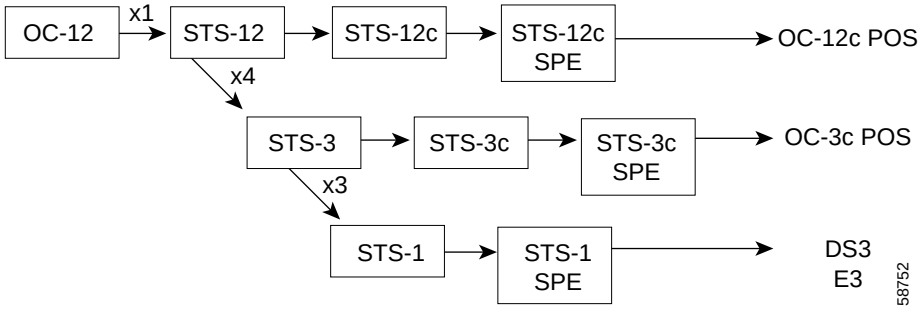
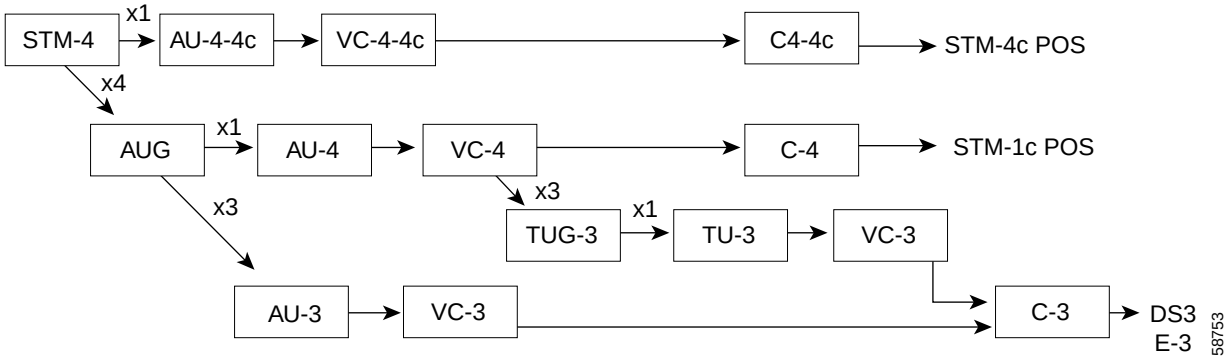


Figure 1-2 Supported SDH Multiplexing Hierarchy on the 1-port ChOC-12 OSMs



Supported Features on the Channelized OC-12/T3 OSMs

The OSM-1CHOC12/T3-SI support the following standard Cisco IOS SONET/SDH features:

SONET Compliance

The OSM-1CHOC12/T3-SI supports 1+1 SONET Automatic Protection Switching (APS).

SONET Errors, Alarms, and Performance Monitoring

This section lists the supported SONET errors, alarms, and perfomance monitoring features:

- Signal Failure Bit Error Rate (SF-ber)
- Signal Degrade Bit Error Rate (SD-ber)
- Signal Label Payload Construction (C2)
- Path Trace Byte (J1)
- Loss of Signal (LOS)
- Loss of Frame (LOF)
- Error Counts for B1
- Threshold Crossing Alarms (TCA) for B1
- Line Alarm Indication Signal (LAIS)
- Line Remote Defection Indication (LRDI)
- Line Remote Error Indication (LREI)
- Error Counts for B2
- Threshold Crossing Alarms (TCA) for B2
- Path Alarm Indication Signal (PAIS)
- Path Remote Defect Indication (PRDI)
- Path Remote Error Indication (PREI)
- Error Counts for B3
- Threshold Crossing Alarms (TCA) for B3
- Loss of Pointer (LOP)
- Path Unequipped (PUNEQ)
- Path Label Mismatch (PPLM)
- New Pointer Events (NEWPTR)
- Positive Stuffing Event (PSE)
- Negative Stuffing Event (NSE)

SONET Synchronization

This section lists the supported SONET synchronization features:

- Local timing (internal timing for inter-router connections over dark fiber or WDM equipment): +/- 4.6 ppm clock accuracy over full operating temperature
- Loop timing (loop timing for connecting to SONET/SDH equipment)

WAN Protocols

This section lists the supported WAN protocols:

- Multiprotocol Label Switching (MPLS) and MPLS/VPN

See Chapter 11, "Configuring Multiprotocol Label Switching on the Optical Services Modules" for information configuring MPLS/VPN on the channelized OSMs.

- Point-to-Point Protocol (PPP) IETF RFC 1661
- HDLC (IETF RFC 1662)
- PPP over SONET with $1+x^{43}$ Self-Synchronous Payload Scrambling
- Frame Relay

  Configure the channelized interfaces for Frame Relay as described in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.1 under "Configuring Frame Relay" and in the *Cisco IOS Wide-Area Networking Command Reference*, Release 12.1 at these URLs:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_c/wcdfrely.htm

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_r/wrdfrely.htm

  Configure traffic shaping for Frame Relay as described in the *Cisco IOS Quality of Service Solutions Configuration Guide* under "Configuring Distributed Traffic Shaping" at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt4/qcfdts.htm.
- Cisco Protection Group Protocol over UDP/IP (Port 172) for APS and MSP

  Configure the serial interface encapsulation as described in the *Cisco IOS Interface Configuration Guide* under "Configuring Serial Interfaces" and in the *Cisco IOS Interface Command Reference* publication at these URLs:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/inter_c/index.htm

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/inter_r/index.htm

Network Management

This section lists the supported network management features:

- Local Loopback
- Network Loopback
- NetFlow Data Export
- Performance Statistics for Timed Intervals (RFC 1595)

DS-3 Support

This section lists the supported DS-3 features:

- Framing control, C-bit or M23
- Local (internal) clocking mode
- Loopback modes
- Bit error rate test (BERT) diagnostics for each DS-3 channel
- Receive and transmit alarm processing
- Performance and error counters
- Far-End Alarm and Control (FEAC) support

DSU Mode

This section lists the supported DSU modes:

- Digital Link
- Verilink
- Adtran
- Larscom
- Kentrox

Quality of Service Protocols

For information on configuring QoS on the channelized OSMs, see Chapter 9, "Configuring QoS on the Optical Services Modules."

The following QoS features are supported on the channelized OSMs:

- PFC2 QoS on the LAN and WAN ports.
- Differentiated Services Control Point (DSCP)
- IP Precedence classification

  Configure class-based marking as described in the *Class-Based Marking* Feature Module at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm
- Classification and priority marking based on the following:
  - Ethertype
  - IP Source Address (SA)
  - IP Destintation Address (DA)
  - TCP port number
  - UDP port number
  - IP SA + TCP/UDP port number + IP DA + TCP/UDP port number
- Class-based weighted fair queuing (CBWFQ) on the WAN ports.
- Low latency queuing (LLQ) on the WAN ports.
- Hierarchical traffic shaping for Frame Relay, HDLC, and PPP encapsulations.

  For general information on classification, marking, and queuing in Cisco IOS, refer to the "Classificiation" chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.1 at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/index.htm

  For information about platform-independent Cisco IOS QoS commands, refer to the *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.1 at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/index.htm

Configuring the Channelized Modules

These sections describe how to configure the channelized modules:

Configuring the SONET Controller

To configure the SONET controller, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller sonet` *slot/port*	Selects a port and enters controller configuration mode.
Step 3	`Router(config-controller)# [no] framing {sonet | sdh}`	Configures the framing mode of the ChOC-12 to SONET or SDH. SDH is the ITU standards equivalent of SONET. SONET is the default.

Step 4	`Router(config-controller)#`
	`sts-1 {sts-1 number} serial {T3 | E3}`
	`sts-1 {start sts-1 number} - {end sts-1 number} pos`
	`au-3 {au-3 number} serial {T3 | E3}`
	`au-3 {start au-3 number} - {end au-3 number} pos`
	`au-4 {start-au4-number} vc-3 {vc3-number} serial [t3|e3]`
	`au-4 {start-au4-number} - {end-au4-number} pos`
		Provisions the channels for the interface. Select the channel provisioning command option appropriate to your needs.

	Command	Purpose
Step 5	`Router(config-controller)# clock source {internal [primary | secondary] | line [primary | secondary]}`	Configures the clock source used by the SONET controller. • **internal**—The clocking source is obtained from the port adapter line. • **line**—The clocking source is obtained from the network. • **primary**—Provides the first priority clock for internal circuitry. • **secondary**—Provides the second clock for internal circuitry when the primary clock fails. The network clocking source is the default.
Step 6	`Router(config-controller)# [no] loopback {internal | line}`	Enables or disables loopback mode on a SONET controller. • **internal**—Data is looped from the transmit path to the receive path allowing diagnostics to send data to itself without relying on any external connections. • **line**—Data is looped from the external port to the transmit port and back out the external port. No loopback enabled is the default.
Step 7	`Router(config-controller)# alarm-report {all | b1-tca | b2-tca | b3-tca | lais | lrdi | pplm | ptim | sd-ber | sf-ber | slof | slos}`	(Optional) Enables alarm reporting.
Step 8	`Router(config-controller)# threshold {b1-tca` *value* `| b2-tca` *value*`| b3-tca` *value*`| sd-ber` *value*`| sf-ber` *value*`}`	(Optional) Sets BER threshold values.
Step 9	`Router(config-controller)# [no] description` *string*	(Optional) Specifies up to 80 characters of text describing the SONET controller. No description is the default.

Configuring the POS Interface

After you verify the controller configuration, you can configure the POS interface. The configuration below is basic, and you may need to specify additional interface parameters depending on your network requirements.

To configure the POS interface, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# interface POS` *slot/port:channel#*	Specifies the serial port and channel to configure.
Step 3	`Router(config-if)# encapsulation hdlc | ppp`	Specifies the encapsulation type.

	Command	Purpose
Step 4	`Router(config-if)# pos flag j1 expect message rxpathmessagetext | length [16|64] | message txpathmessagetext`	(Optional) Specifies a path message for a channelized interface.
Step 5	`Router(config-if)# ip address ip-address mask [secondary]`	Assigns an IP address and subnet mask to the interface.
Step 6	`Router(config-if)# no shutdown`	Enables the interface.

Configuring the DS-3 Serial Interface

After you verify the controller configuration, you can configure the associated DS-3 channel and serial interfaces on the controller.

Note When connecting a T3 interface to a VeriLink DSU the minimum supported bandwidth for the T3 interface is 6316 Kbps and the bandwidth should be in multiples of 6316 Kbps up to 44210.

To configure the DS-3 interfaces, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# interface serial slot/port:channel#`	Specifies the serial port and channel to configure.
Step 3	`Router(config-if)# framing {c-bit | m23}`	Specifies the framing.
Step 4	`Router(config-if)# [no] dsu mode {0-4}`	Specifies the DSU mode: 0–Digital-Link 1–Kentrox 2–Larscom 3–Adtran 4–Verilink
Step 5	`Router(config-if)# [no] dsu remote [accept | fullrate]`	Specifies if the local (near-end) interface will accept incoming requests from the remote (far-end) interface, or if the local interface will request that the remote interface set its bandwidth to fullrate.
Step 6	`Router(config-if)# [no] dsu bandwidth Kilobits/sec`	Sets the DSU subrate bandwidth.
Step 7	`Router(config-if)# [no] scramble`	Enables payload scrambling.
Step 8	`Router(config-if)# [no] loopback {local | network | remote}`	Sets the loopback mode.
Step 9	`Router(config-if)# [no] bert pattern [2^15 | 2^20 ] interval [1-1440]`	(Optional) Configures bit-error-rate (BER) testing.
Step 10	`Router(config-if)# alarm-report {all | b3-tca | pais | pplm | plop | prdi | ptim | ptiu | puneq}`	(Optional) Enables reporting of path alarms.

	Command	Purpose
Step 11	`Router(config-if)# overhead {c2 byte value | j1 {expect message message-string | length 16-64 | message message-string}`	Specifies SONET path header byte value.
Step 12	`Router(config-if)# ip address ip-address mask [secondary]`	Assigns an IP address and subnet mask to the interface.
Step 13	`Router(config-if)# [no] keepalive`	Turns on and off keepalive messages.
Step 14	`Router(config-if)# no shutdown`	Enables the interface.

This is an example of an unchannelized DS-3 interface configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# controller t3 3/0
Router (config-controller)#no channelized
Router (config-controller)# exit
Router (config)# interface serial 1/0
Router (config-if)# dsu bandwidth 16000
Router (config-if)# encapsulation frame-relay
Router (config-if)# ip address 10.10.10.10.255.255.255.255
Router (config-if)# no shutdown
Router (config-if)# exit
Router(config)#
```

Configuring Interfaces Using SDH Framing with AU-3 Mapping

This section describes how to enable an interface under SDH framing with AU-3 mapping and specify IP routing on the OSM-1CHOC12/T3-SI channelized modules. To configure interfaces using SDH framing with AU-3 mapping, peform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller sonet slot/port`	Selects the controller.
Step 3	`Router(config-controller)# framing sdh`	Specifies the framing.
Step 4	`Router(config-controller)# aug mapping au-3`	Specifies the AUG mapping.
Step 5	`Router(config-controller)# au-3 au-3 number serial {T3 | E3}`	Provisions the AU-3 channels.
Step 6	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 7	`Router(config)# interface serial slot/port:au-3 number`	Selects the interface.
Step 8	`Router(config-if)# ip address ip-address mask [secondary]`	Assigns an IP address and subnet mask to the interface.
Step 9	`Router(config-if)# [no] shutdown`	Enables the interface.

In this example, a port is configured as 12 E3 interfaces.

Note When you connect an E3 interface to a Digital Link DL3100E E3 access multiplexer DSU, you must use the "clear channel" mode on the Digital Link DSU. When you connect an E3 interface to a Cisco 12000 Series 12-Port packet over E3 line card, you must configure **dsu mode kentrox** on the Cisco 12000 Series 12-Port packet over E3 line card. When you connect an E3 interface to a Cisco C7500 or a Cisco C7200 E3 port adaptor (PA), you must configure **dsu mode 1** on the E3 interface on the E3 PA.

Step 1 Enter the **configure terminal** EXEC command to enter global configuration mode as follows:

```
router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Step 2 Provisision the E3 channels:

```
router(config)# controller sonet 4/1
router(config-controller)# framing sdh
router(config-controller)# overhead s1s0 2
router(config-controller)# aug mapping au-3
router(config-controller)# au-3 1 serial e3
router(config-controller)# au-3 2 serial e3
router(config-controller)# au-3 3 serial e3
router(config-controller)# au-3 4 serial e3
router(config-controller)# au-3 5 serial e3
router(config-controller)# au-3 6 serial e3
router(config-controller)# au-3 7 serial e3
router(config-controller)# au-3 8 serial e3
router(config-controller)# au-3 9 serial e3
router(config-controller)# au-3 10 serial e3
router(config-controller)# au-3 11 serial e3
router(config-controller)# au-3 12 serial e3
```

Step 3 Configure the E3 interfaces:

```
Router(config)# interface serial 5/2:1
Router(config-if)# ip address 10.2.1.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:2
Router(config-if)# ip address 10.2.2.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:3
Router(config-if)# ip address 10.2.3.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:4
Router(config-if)# ip address 10.2.4.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:5
Router(config-if)# ip address 10.2.5.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:6
Router(config-if)# ip address 10.2.6.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:7
Router(config-if)# ip address 10.2.7.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:8
```

```
Router(config-if)# ip address 10.2.8.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:9
Router(config-if)# ip address 10.2.9.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:10
Router(config-if)# ip address 10.2.10.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:11
Router(config-if)# ip address 10.2.11.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial 5/2:12
Router(config-if)# ip address 10.2.12.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
```

Step 4 Write the new configuration to nonvolatile random-access memory (NVRAM) by using the **copy running-config startup-config** command:

```
router# copy running-config startup-config
[OK]
router#
```

Configuring Interfaces under SDH Framing with AU-4 Mapping

This section describes how to enable an interface under SDH framing with AU-4 mapping and specify IP routing on the OSM-1CHOC12/T3-SI channelized modules. In this example, a port is configured as 12 E3 interfaces.

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller sonet` *slot/port*	Selects the controller.
Step 3	`Router(config-controller)# framing sdh`	Specifies the framing.
Step 4	`Router(config-controller)# aug mapping au-4`	Specifies the AUG mapping.
Step 5	`Router(config-controller)#` **au-4** *start-au4-number* **vc-3** *VC3-number* **serial** [**t3**|**e3**] **au-4** *start-au4-number* **-** *end-au4-number* **pos**	Provisions the channels and defines the interface number.
Step 6	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 7	`Router(config)# interface serial` *slot/port. au-4:au-3*	Selects the interface.
Step 8	`Router(config-if)# ip address` *ip-address mask* [**secondary**]	Assigns an IP address and subnet mask to the interface.
Step 9	`Router(config-if)#` [**no**] **shutdown**	Enables the interface.

In this example, AU-4 mapping is used to configure one STM-4 POS interaface, two STM-1 interfaces, and two DS-3 serial interfaces. The DS-3 interface names are constructed from *slot/port. au-4:au-3*. The VC-3 number, ranging 1 through 3, is the TUG-3 (or VC3) number inside the selected AU-4.

The STM-4 interface name is constructed from the first AU-4 number.

Step 1 Enter the **configure terminal** EXEC command to enter global configuration mode as follows:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Step 2 Provision the channels:

```
Router(config)# controller sonet 5/1
Router(config-controller)# framing sdh
Router(config-controller)# aug mapping au-4
Router(config-controller)# au-4 1-4 pos
Router(config-controller)# au-4 5 pos
Router(config-controller)# au-4 6 vc-3 1 serial t3
Router(config-controller)# au-4 6 vc-3 2 serial t3
Router(config-controller)# au-4 7 pos
Router(config-controller)# end
```

Step 3 Configure the interfaces:

```
Router(config)# interface pos 5/1:1
Router(config-if)# encapsulation ppp
Router(config-if)# ip address 10.10.10.10 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# end
Router(config)# interface pos 5/1:5
Router(config-if)# encapsulation ppp
Router(config-if)# ip address 10.10.10.11 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# end
Router(config)# interface serial 5/1.6:1
Router(config_if)# framing c-bit
Router(config-if)# dsu mode 0
Router(config-if)# dsu remote accept
Router(config-if)# dsu bandwidth 30000
Router(config-if)# scramble
Router(config-if)# loopback remote
Router(config-if)# ip address 10.10.10.12. 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# end
Router(config)# interface serial 5/1.6:2
Router(config_if)# framing c-bit
Router(config-if)# dsu mode 0
Router(config-if)# dsu remote accept
Router(config-if)# dsu bandwidth 45000
Router(config-if)# scramble
Router(config-if)# loopback remote
Router(config-if)# ip address 10.10.10.12. 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# end
Router(config)# interface pos 5/1:7
Router(config-if)# encapsulation ppp
Router(config-if)# ip address 10.10.10.13 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# end
```

Step 4 Write the new configuration to nonvolatile random-access memory (NVRAM) by using the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
[OK]
Router#
```

Configuring Automatic Protection Switching

Automatic protection switching (APS) allows switchover of packet-over-SONET (POS) circuits and is often required when connecting SONET equipment to telecommunications equipment. When APS is configured, a protect POS interface is brought into the SONET network from the intervening SONET equipment and the protect POS interface becomes the working POS interface on the circuit.

Note Note that on the OSM-1CHOC12/T3-SI, APS is configured at the SONET controller level rather than at the interface level.

The protect interface is configured with the IP address of the router that has the working interface. The APS Protect Group Protocol provides communication between the process controlling the working interface and the process controlling the protect interface. When you use the APS Protect Group Protocol, POS interfaces can be switched in the event of a router failure, degradation or loss of channel signal, or manual intervention.

Two SONET connections are required to support APS. In a telecommuncations environment, the SONET circuits must be provisioned as APS. You must also provision the operation, mode, and revert options. If the SONET connections are homed on two separate routers (the normal configuration), an out-of-band (OOB) communications channel between the two routers needs to be set up for APS communication.

When configuring APS, we recommend you configure the working interface first, along with the IP address of the interface being used as the APS OOB communications path.

Note To prevent the protected interface from becoming the active circuit and disabling the working circuit when it is discovered, configure the working interface before configuring the protected interface.

For more information on APS, refer to the *Cisco IOS Interface Configuration Guide, Release 12.1* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/inter_c/index.htm

Configuring the Working Interface

To configure the working interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# controller sonet` *slot/port*	Enters SONET controller-configuration mode from the config prompt.
Step 2	`Router(config-controller)# aps working` *circuit-number*	Configures this interface as a working interface.
Step 3	`Router(config-controller)# end`	Exits configuration mode.
Step 4	`Router# show aps` `Router# show aps controller`	Displays information about the controllers so that you can verify the configuration.

Note If a router has two or more protect interfaces, the **aps group** command for each interface must precede the corresponding **aps protect** command.

Configuring the Protect Interface

To configure the protect interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller sonet` *slot/port*	Enters SONET controller-configuration mode from the config prompt.
Step 2	`Router(config-controller)# aps protect` *circuit-number ip-address*	Configures this interface as a protect interface. Specifies the IP address of the router that contains the working interface.
Step 3	`Router(config-controller)# end`	Exits configuration mode.
Step 4	`Router# show aps` `Router# show aps controller`	Displays information about the controllers so that you can verify the configuration.

Configuring Frame Relay and Frame Relay Traffic Shaping

This section describes Frame Relay configurations, platform-specific commands, and limitations:

- Frame Relay Limitations and Restrictions, page 5-14
- Frame Relay Traffic Shaping Configuration Example, page 5-15

Configure the channelized interfaces for Frame Relay as described in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.1 under "Configuring Frame Relay" and in the *Cisco IOS Wide-Area Networking Command Reference*, Release 12.1 at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_c/wcdfrely.htm

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_r/wrdfrely.htm

Configure traffic shaping for Frame Relay as described in the *Cisco IOS Quality of Service Solutions Configuration Guide* under "Configuring Distributed Traffic Shaping" at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt4/qcfdts.htm.

Frame Relay Limitations and Restrictions

The following limitations and restrictions apply to Frame Relay:

- Frame Relay is not supported on SVCs.
- IP addresses cannot be assigned to main interfaces configured for Frame Relay.
- Frame Relay is supported only on point-to-point connections.
- Frame Relay switching functionality is not supported. The frame-relay switching configuration is available only to configure the **frame-relay intf-type dce** option.
- Frame Relay fragmentation and compression is not supported.
- FECN and BECN statistics per DLCI are not supported.

- Only FIFO queuing is supported.
- DLCI is configurable on subinterfaces only and cannot be configured on the main interface.
- The maximum supported number of configured DLCIs per chassis is 4,000.
- Only class-based traffic shaping is supported. The following commands are not supported:
  - Router(config-pmap-c)# **shape** [**average** | **peak**] *mean-rate* [[*burst-size*] [*excess-burst-size*]]
  - Router (config-pmap-c)# **priority** {*kbps* | **percent** *percent*} [*bytes*]
  - Router (config-pmap-c)# **fair-queue** *number-of-queues*
  - Router(config-map-class)# **frame-relay adaptive-shaping** [**becn** | **foresight**]1
  - Router(config-map-class)# **frame-relay cir** {**in** | **out**} *bps*
  - Router(config-map-class)# **frame-relay** {**bc** | **be**} {**in** | **out**} *bits*
  - Router(config-map-class)# **frame-relay traffic-rate average** [**peak**]
  - Router(config-map-class)# **frame-relay priority-group** *list-number*
  - Router(config-map-class)# **frame-relay fragment** *fragment_size*
  - Router(config-if)# **frame-relay payload-compress packet-by-packet**
  - Router(config-if)# **frame-relay de-group** *group-number dlci*
  - Router# **show traffic-shape queue**

Frame Relay Traffic Shaping Configuration Example

	Command	Purpose
Step 1	`Router(config-pmap)# class-map [match-all | match-any]`	Creates a class map to be used for matching packets to a class you define and specifies the criteria to match on. Match criteria for classes can be based on IP DSCP or IP precedence.
Step 2	`Router(config-pmap)# match`	Identifies a match criterion.
Step 3	`Router(config)# policy-map policy_map`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
Step 4	`Router(config-pmap)# class class-name`	Defines the classes you want the service policy to contain.
Step 5	`Router(config-pmap-c)#shape average mean-rate [burst-size]`	Shapes traffic to the indicated bit rate.
Step 6	`Router(config)# map-class frame-relay map-class-name`	Specifies a map class to define quality of service (QoS) values.
Step 7	`Router(config-map-class)# no frame-relay adaptive-shaping`	Disables backward notification.
Step 8	`Router(config-map-class)# service-policy input policy-map`	Attaches the specified policy map to the input interface.
Step 9	`Router(config-map-class)# service-policy output policy-map`	Attaches the specified policy map to the output interface.
Step 10	`Router(config)#interface interface`	Specifies the interface to which the policy map will be applied.
Step 11	`Router(config-subif)# ip address ip_address mask`	Assigns an IP address to the subinterface.

	Command	Purpose
Step 12	`Router(config-subif)# no cdp enable`	Disables CDP.
Step 13	`Router(config-subif)# frame-relay interface-dlci` *dlci*	Assigns a data link connection identifier (DLCI) to a specified Frame Relay subinterface.
Step 14	`Router(config-fr-dlci)# class` *class-name*	Specifies the name a predefined map-class which was defined with the **map-class frame-relay** command.

We recommend that you explicitly disable CDP on the subinterfaces. Should CDP be required on the subinterfaces, the input-queue depth may need to be adjusted. To accommodate the number of incoming CDP packets, configure the input-queue depth on the main interface to be slightly larger than the number of subinterfaces on which you have enabled CDP. The default input-queue depth is 75 and it can be adjusted with the **hold-queue** interface command as follows:

```
Router(config-if)#hold-queue 300 in
```

In the following example, traffic leaving subinterface 6/1:1.1 or 6/1:1.2 is shaped to 1 Mbps:

```
Router(config)# class-map class-p2p-all
Router(config-cmap)# match any
Router(config-cmap)# exit
Router(config)# policy-map dts-p2p-all-action
Router(config-pmap)# class class-p2p-all
Router(config-pmap-c)# shape average 1000000
Router(config-pmap-c)# exit
Router(config)# interface Serial6/1:1.1 point-to-point
Router(config-subif)# service-policy output dts-p2p-all-action
Router(config-subif)# exit
Router(config)# interface serial6/1:1.2 point-to-point
Router(config-subif)# service-policy output dts-p2p-all-action
```

The following example shows a per-DLCI traffic configuration:

```
Router(config)# class-map match-all fr-classmap
Router(config-cmap)# match any
Router(config-cmap)# exit
Router(config)# policy-map fr-pmap
Router(config-pmap)# class fr-classmap
Router(config-pmap-c)# shape average 8000000 32000 32000
Router(config-pmap-c)# exit
Router(config)# interface Serial6/1:1.1 point-to-point
Router(config-subif)# ip address 72.0.0.1 255.255.0.0
Router(config-subif)# mls qos trust dscp
Router(config-subif)# frame-relay interface-dlci 18
Router(config-fr-dlci)# class fr-shaping
Router(config-fr-dlci)# exit
Router(config)# map-class frame-relay fr-shaping
Router(config-map-class)# no frame-relay adaptive-shaping
Router(config-map-class)# service-policy input fr-pmap
Router(config-map-class)# service-policy output fr-pmap
```

Configuration Examples

This following configurations are shown in this section:

- Configuring Channelized POS, page 5-17
- Configuring Channelized DS-3, page 5-17

- Configuring Basic APS, page 5-18
- Multiple APS Interface Configuration, page 5-18

Configuring Channelized POS

This example shows how to configure channelized POS. The sts-1 number is the logical STS-1 path inside the OC-12 frame and ranges from 1–12. For an OC-3 channel, the STS-1 numbers cannot cross the OC-3 boundary. For example, sts-1 1–6 would be an illegal configuration.

Perform the appropriate configurations for each POS interface after you have configured for channelized POS:

Step 1 Configure the SONET controller:

```
Router(config)# controller sonet 2/1
```

Step 2 Provision the channels:

```
Router(config-controller)# sts-1 1-3 pos
Router(config-controller)# sts-1 13-15 pos
Router(config-controller)# sts-1 16-18 pos
Router(config-controller)# exit
```

Step 3 Configure the POS interface:

```
Router(config)# interface pos 2/1:13
Router(config-if)# encapsulation ppp
Router(config-if)# ip address 10.10.10.10 255.255.255.0
Router(config-if)# end
```

Configuring Channelized DS-3

This example show how to configure channelized DS-3:

Step 1 Configure the SONET controller:

```
Router(config)# controller sonet 2/1
```

Step 2 Provision the channels:

```
Router(config-controller)# sts-1 4 serial t3
Router(config-controller)# exit
```

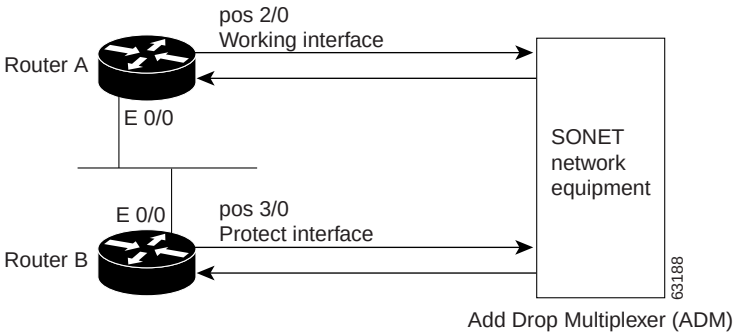
Step 3 Configure the serial interface:

```
Router(config)# interface serial 2/1:4
Router(config-if)# framing c-bit
Router(config-if)# dsu mode 0
Router(config-if)# dsu remote accept
Router(config-if)# dsu bandwidth 30000
Router(config-if)# scramble
Router(config-if)# loopback remote
Router(config-if)# ip address 10.1.1.1. 255.255.255.0
Router(config-if)# end
```

Configuring Basic APS

The following example shows the configuration of APS on router A and router B (see Figure 5-3). In this example, router A is configured with the working interface, and router B is configured with the protect interface. If the working interface on router A becomes unavailable, the connection will automatically switch over to the protect interface on router B. The working and protect interfaces are configured at the controller level.

Figure 1-3 Basic APS Configuration



Step 1 On router A, which contains the working interface, use the following configuration:

```
Router# configure terminal
Router(config)# interface ethernet 0/0
Router(config-if)# ip address 7.7.7.7 255.255.255.0
Router(config)# controller sonet 7/1
Router(config-controller)# aps working 1
Router(config-controller)# end
Router#
```

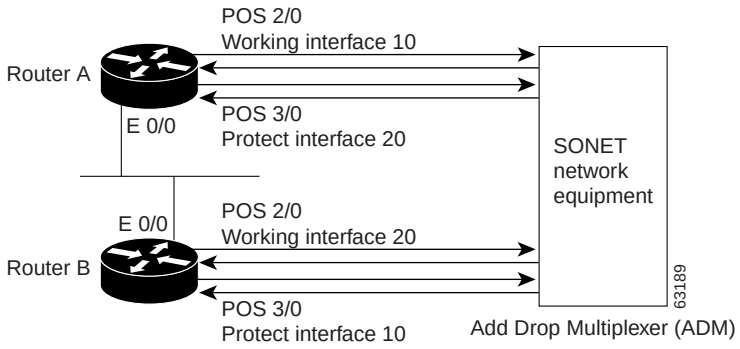
Step 2 On router B, which contains the protect interface, use the following configuration:

```
Router# configure terminal
Router(config)# interface ethernet 0/0
Router(config-if)# ip address 7.7.7.6 255.255.255.0
Router(config-controller)# controller sonet 3/1
Router(config-controller)# aps protect 1 7.7.7.7
Router(config-controller)# end
Router#
```

Multiple APS Interface Configuration

To configure more than one protect/working interface, use the **aps group** command. The following example in Figure 5-4 shows the configuration of grouping more than one working/protect interface. In this example, router A is configured with a working interface and a protect interface, and router B is configured with a working interface and a protect interface. If the working interface 2/0 on router A becomes unavailable, the connection will switch over to the protect interface 3/0 on router B because they are both in APS group 10. Similarly, if the working interface 2/0 on router B becomes unavailable, the connection will switch over to the protect interface 3/0 on router A because they are both in APS group 20.

Figure 1-4 Multiple Working and Protect Interfaces Configuration



Note Configure the working interface before configuring the protect interface to avoid the protect interface from becoming the active circuit and disabling the working circuit when it is discovered.

Step 1 On router A, which contains the working interface for group 10 and the protect interface for group 20, use the following configuration:

```
Router# configure terminal
Router(config)# interface ethernet 0/0
Router(config-if)# ip address 7.7.7.6 255.255.255.0
Router(config-if)# exit
Router(config)# controller sonet 7/1
Router(config-controller)# aps group 10
Router(config-controller)# aps working 1
Router(config-controller)# exit
Router(config)# contoller sonet 3/0
Router(config-controller)# aps group 20
Router(config-controller)# aps protect 1 7.7.7.7
Router(config-controller)# end
Router#
```

Step 2 On router B, which contains the protect interface for group 10 and the working interface for group 20, use the following configuration:

```
Router# configure terminal
Router(config)# interface ethernet 0/0
Router(config-if)# ip address 7.7.7.7 255.255.255.0
Router(config-if)# exit
Router(config)# controller sonet 2/0
Router(config-controller)# aps group 20
Router(config-controller)# aps working 1
Router(config-controller)# exit
Router(config)# controller sonet 3/0
Router(config-controller)# aps group 10
Router(config-controller)# aps protect 1 7.7.7.6
Router(config-controllers)# end
Router#
```

CHAPTER 1

Configuring the Channelized OC-12/T1 Optical Services Modules

This chapter describes how to configure the channelized OSM-1CHOC12/T1-SI SONET/SDH Optical Services Modules (OSMs).

The chapter consists of these sections:

Understanding the Channelized OC-12/T1 Optical Services Modules

These sections describe the SONET/SDH mappings and multiplex hierachy and the features supported on the channelized OSMs:

Channelized OC-12/T1 OSM Multiplexing and Mappings

The 1-port OSM-CHOC12/T1-SI module supports provisioning of OC-12, OC-3, DS-3, subrate DS-3, E-3, E1, DS1 and DS0 links.

Figure 6-1 illustrates the SONET multiplexing hierarchy supported on the OSM-1CHOC12/T1-SI module.

Figure 6-2 illustrates the SDH multiplexing hierarchy supported on the OSM-1CHOC12/T1-SI module.

Figure 1-1 Supported SONET Multiplexing Hierarchy on the Channelized OC-12/T1 OSMs

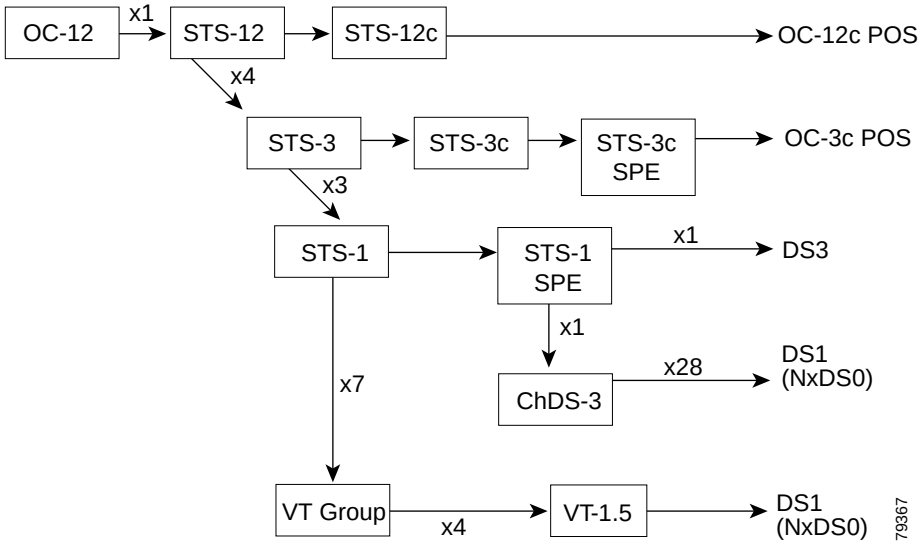
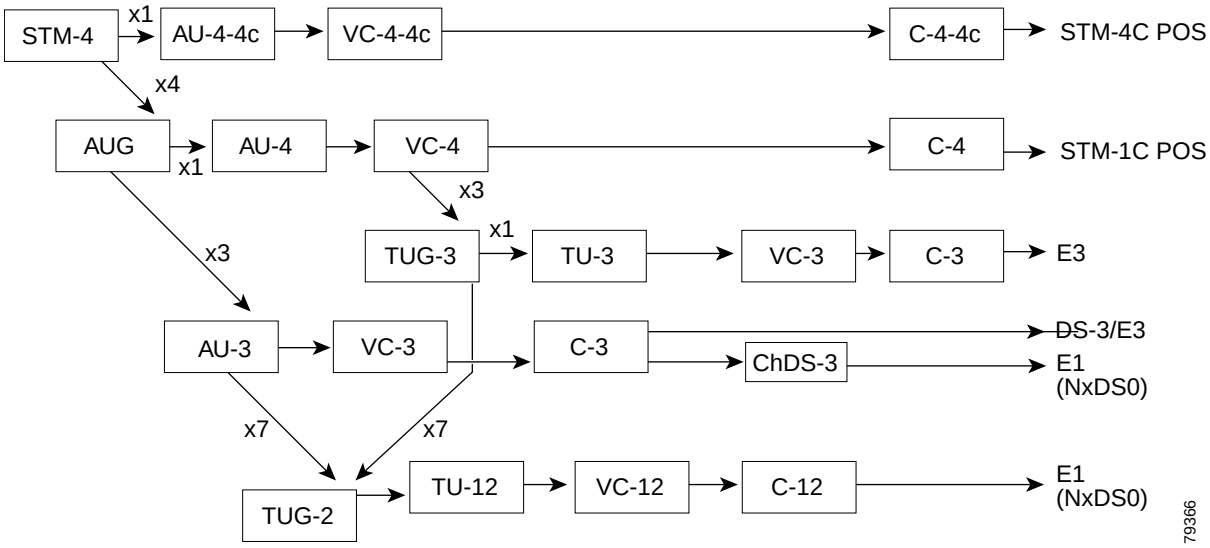


Figure 1-2 Supported SDH Multiplexing Hierarchy on the Channelized OC-12/T1 OSMs



Each OC-12/STM-4 port of OSM-CHOC12/T1 can support any allowed combination of OC-3c/STM-1/DS3/DS1/E3/E1/T1s up to 1023 channels.

For example, each port can support the following:

- 4 OC-3c/STM-1 at 155.52 Mbps
- 12 DS3 at 44.7 Mbps
- 12 E3s at 34.368 Mbps
- 252 E1s at 2.048Mbps
- 336 DS1s at 1.5 Mbps
- 1023 DS0s at 64 Kbps

Channelized OC-12/T1 OSM Features

The channelized 1-port OSM-CHOC12/T1-SI modules support the following standard Cisco IOS SONET/SDH features:

SONET Compliance

The OSM-CHOC12/T1 modules support 1+1 SONET Automatic Protection Switching (APS) and comply with ANSI T1.107DS1/DS-3 standards, ANSI T1.403 1998, G.703, G.704, AT&T 54014 (DS-3), 54016 (DS1).

Errors, Alarms, and Performance Monitoring

These sections lists the supported errors, alarms, and perfomance monitoring features:

Regenerator Section Errors and Alarms

- LOS

- LOF
- RS-TIM
- RS-BIP

Multiplex Section Errors and Alarms

- MS-AIS
- MS-REI
- MS-RDI,
- MS-BIP

Administrative Unit Errors and Alarms

- AU-AIS
- AU-LOP
- AU-BIP

Higher Order Path Errors and Alarms

- HP-UNEQ
- HP-TIM
- HP-BIP
- HP-REI
- HP-RDI
- HP-PLM

Lower-Order Path Errors and Alarms

- TU-LOP
- TU-NDP
- TU-AIS
- TU-LOM
- BIP-2/B3
- LP-UNEQ
- LP-RDI
- LP-REI
- LP-RFI
- LP-TIM
- LP-PLM

Section Errors and Alarms

- LOS
- LOF

- TIM-S
- BIP-S

Line Errors and Alarms

- AIS-L
- REI-L
- RDI-L
- BIP-L

STS Path Errors and Alarms

- AIS-P
- LOP-P
- UNEQ-P
- TIM-P
- REI-P
- RDI-P
- LOM
- BIP-P
- CV-P
- PLM-P

VT Path Errors and Alarms

- LOP-V
- NDF-V
- AIS-V
- CV-V
- UNEQ-V
- RDI-V
- REI-V
- RFI-V
- TIM-V
- PLM-V

Additional Errors and Alarms

- Signal Failure Bit Error Rate (SF-ber)
- Signal Degrade Bit Error Rate (SD-ber)
- Path Trace Byte (J1)
- Error Counts for B1

- Error Counts for B2
- Error Counts for B3
- Threshold Crossing Alarms (TCA) for B1
- Threshold Crossing Alarms (TCA) for B2
- Threshold Crossing Alarms (TCA) for B3

Performance Monitoring

The following SDH performance monitoring PMON data are collected for Regeneration Section, Multiplex Section, Path Section and Tributary Path Section:

- Errored Seconds (ES)
- Severely Errored Seconds (SES)
- Unavailable Seconds (UA)
- Code Violations

The following SONET PMON data are collected:

- Received path, section, and line BIP-8 error counts
- Received path remote error indication (REIs)
- Accumulated B2 errors and line remote errors (M1)

The following performance monitoring (PMON) data are collected for DS1/E1 Near End PMON:

- Line Errored Seconds (LES)
- Controlled Slip Seconds (CSS)
- Errored Seconds (ES)
- Bursty Errored Seconds (BES)
- Serverely Errored Seconds (SES)
- Severely Errored Framing Seconds (SEFS)
- Degraded Minutes (DM)
- Unavailable Seconds (UAS)
- Path Coding Violation (PCV)
- Controlled Slip (CS)
- Line Coding Violation (LCV) is not applicable

The following PMON data are collected for DS1/E1 Far End PMON:

- Line Errored Seconds (LES)
- Controlled Slip Seconds (CSS)
- Errored Seconds (ES)
- Bursty Errored Seconds (BES)
- Serverely Errored Seconds (SES)
- Severely Errored Framing Seconds (SEFS)
- Degraded Minutes (DM)
- Unavailable Seconds (UAS)

- Path Coding Violation (PCV)
- Controlled Slip (CS)

The following PMON data are collected for DS-3 Near End PMON:

- P-Bit Coding Violation (PCV)
- C-Bit Coding Violation (CCV)
- Line Errored Seconds (LES)
- P-Bit Errored Seconds (PES)
- P-Bit Severely Errored Seconds (PSES)
- C-Bit Errored Seconds (CES)
- C-Bit Severely Errored Seconds (CSES)
- Severely Errored Framing Seconds (SEFS)
- Unavailable Seconds (UAS)

The following PMON data are collected for E3 PMON:

- Line Errored Seconds (LES)
- Severely Errored Framing Seconds (SEFS)
- Unavailable Seconds (UAS)

WAN Protocols and Services

This section lists the supported WAN protocols and services:

- Point-to-Point Protocol (PPP) IETF RFC 1661
- Distributed Multilink PPP (dMLP)

  The OSM-CHOC12/T1 modules support dMLP, which means that the MLP encapsulation is performed on the module rather than the route processor. The following dMLP features are supported:

  – 168 bundles per port
  – A maximum of 12 DS1/E1 channels per bundle
  – All links in a bundle should operate at the same speed (fractional E1 and T1 links are not supported)
  – 100 ms of differential delay
  – Transmit fragment size can be 256 or 512 bytes

  For dMLP overview and configuration information, see the "Configuring Distributed MLPPP" section on page 7-11 in Chapter 7, "Configuring the Channelized 12-port CT3/T1 Optical Services Modules."
- HDLC (IETF RFC 1662)
- PPP over SONET with $1+x^{43}$ Self-Synchronous Payload Scrambling
- Frame Relay
  – For information on Frame Relay support on the OSM-CHOC12/T1 modules modules , see the "Configuring Frame Relay and Frame Relay Traffic Shaping" section on page 1-14 in Chapter 1, "Configuring the Channelized OC-12/T3 SONET/SDH Optical Services Modules."

  - For information on Multilink Frame Relay (FRF.16), see Multilink Frame Relay (FRF.16), page 1-6.

- MPLS/VPN

  For MPLS/VPN configuration information, see Chapter 11, "Configuring Multiprotocol Label Switching on the Optical Services Modules."

SONET/SDH Failure Recovery Support

- Cisco Protection Group Protocol over UDP/IP (Port 172) for APS and MSP

  For APS overview and configuration information, see the "Configuring Automatic Protection Switching" section on page 1-13 in Chapter 1, "Configuring the Channelized OC-12/T3 SONET/SDH Optical Services Modules."

MIB Support

This section lists the supported SNMP MIBs:

- Performance Statistics for Timed Intervals (RFC 1595)
- SONET/SDH MIB (RFC 1595)
- DS-3/E3 MIB (RFC 1407)
- DS1/E1 MIB (RFC1406)
- IF-MIB (RFC1573)
- OLD-CISCO-CHASSIS-MIB
- SNMP v2c (RFC1901-1907)
- SNMP v3
- Other standard MIBs are supported by the Cisco IOS software, including industry standard and Cisco proprietary MIBs.

OC-12 POS Interface Configuration

The OSM-CHOC12/T1 modules support the following features:

- Sonet or SDH framing mode
- Internal or line clock source
- APS 1:1 Protection Switching
- Network or local loopback
- Single Mode, Short/Intermediate Range (SM-IR) LC SFF (Small Form Factor) optics

E3 Lines

This section lists the supported E3 Ofeatures:

- Unchannelized E3
- G.751 E3 framing
- E3 transmit clock is synchronized to the internal Telcom Bus clock

- BERT pattern generation and detection (only two E3s can be tested simultaneously)
- Local and line loopback
- RFC 1407 MIB support

Note When you connect an E3 interface to a Digital Link DL3100E E3 access multiplexer DSU, you must use the "clear channel" mode on the Digital Link DSU. When you connect an E3 interface to a Cisco 12000 Series 12-Port packet over E3 line card, you must configure **dsu mode kentrox** on the Cisco 12000 Series 12-Port packet over E3 line card. When you connect an E3 interface to a Cisco C7500 or a Cisco C7200 E3 port adaptor (PA), you must configure **dsu mode 1** on the E3 interface on the E3 PA.

DS-3 Lines

This section lists the supported DS-3 features:

- Framing control can be C-bit, M23, or auto-detect
- Channelized mode can be T1 or E1 depending on SONET or SDH framing.
- Local (internal) clocking mode
- Loopback modes can be network, local, or remote.
- Generation and termination of DS-3 maintenance data link (MDL) in C-bit framing.
- For supported errrors, alarms, and performance monitoring, see the "Errors, Alarms, and Performance Monitoring" section on page 6-3.
- DSU Mode

  The following DSU modes are supported:

  - Digital Link
  - Verilink
  - Adtran
  - Larscom
  - Kentrox

T1 Lines

The T1 lines support the following:

- Local, line, and remote loopback.
- DS-1 facilities data link (FDL) in Extended Super Frame (ESF) framing
- Bit error rate testing (BERT) is supported on each of the T1 lines. It can be done over a framed or unframed DS-1 signal. The OSM-CHOC12/T1 modules allow a maximum of 6 simultaneous T1 BER tests in each group of 84 T1 lines associated with three contiguous STS-1s. Thus there may be up to 6 active T1 BER tests in an OC-3 port or 24 active T1 BER tests in an OC-12 port.

Note BERT channel-groups and timeslots are also supported for both SDH and SONET.

- In the channelized mode of operation, an OC-12/T1 OSM port can be channelized into a maximum of 336 T1 lines. Each of the T1 lines contains 24 timeslots or 64 or 56 kbps each. The T1 lines can support one or more user data channels, which appear to the system as serial interfaces. You can group the timeslots into individual logical channel groups, each of which may carry data with different data link layer protocol encapsulations. You are limited to a total of 1024 logical channel groups per port.
- Each logical channel group can be composed of individual 64-kbps or 56-kbps timeslots and ranges of timeslots, for example, 1, 9, 12-14. Each logical channel group can contain from 1-24 timeslots maximum; the same timeslot cannot be used in more than one logical channel group. Any unused timeslots are filled with programmable idle-channel data.

Note If you assign only one channel group to an T1 line, it is a fractional T1 line. If you assign more than one channel group to an T1 line, it is a channelized T1 line.

E1 Lines

The E1 lines support the following:

- Local, line, and remote loopback.
- Any of the E1 lines can be configured as either E1 frames or E1 cyclic redundancy check (CRC) multiframes, as specified by CCITT/ITU G.704 and G.706.
- Bit error rate testing (BERT) is supported on each of the E1 lines. It can be done over a framed or unframed E1 signal. The OSM-CHOC12/T1 modules allow up to 6 simultaneous E1 BER tests in each group of 63 E1 lines associated with each STM-1 port and 24 simultaneous E1 tests per STM-4 port.

Note BERT channel-groups and timeslots are also supported for both SDH and SONET.

- Channelized E1—Any of the E1 lines can be configured as channelized E1 lines, but you are limited to a total of 1023 logical channels. You can group the time slots in these E1 lines into several individual logical channel groups, each of which carries data with different data link layer protocol encapsulations.
- Each logical channel group can be composed of individual 64-kbps or 56-kbps timeslots and ranges of timeslots, for example, 1, 9, 12-14. Each logical channel group can contain from 1-31 timeslots maximum; the same timeslot cannot be used in more than one logical channel group. Any unused timeslots are filled with programmable idle-channel data.
- Fractional E1— A fractional E1 line is a subset of the full E1 bandwidth, which uses *n* x 64 kbps; where *n* is a timeslot in the range of 1-31.

  Fractional E1 lines contain only a single logical channel group that can be either a single 64-kbps timeslot or a range of timeslots; for example timeslot 1, or timeslots 15-23. Any unused timeslots are filled with programmable idle-channel data.

Note If you assign only one channel group to an E1 line, it is a fractional E1 line. If you assign more than one channel group to an E1 line, it is a channelized E1 line.

- Unframed E1—Any of the E1 lines can be configured as unframed E1 data lines. Each unframed E1 line contains no framing overhead and is not timeslot divided.

DS0 Lines

T1 and E1 lines can be channelized down to DS0/64kbps or DS0/56kbps. See the "Configuring the T1 Lines" section on page 6-17 for configuration information.

Quality of Service Protocols

The following QoS features are supported on the Channelized OSMs:

- Hierarchical traffic shaping for Frame Relay, HDLC, and PPP encapsulations.
- PFC2 QoS on the LAN and WAN ports.
- Differentiated Services Control Point (DSCP)
- IP Precedence classification

  Configure class-based marking as described in the *Class-Based Marking* Feature Module at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm
- Classification and priority marking based on the following:
  - Ethertype
  - IP Source Address (SA)
  - IP Destintation Address (DA)
  - TCP port number
  - UDP port number
  - IP SA + TCP/UDP port number + IP DA + TCP/UDP port number
- Weighted Random Early Detection (WRED)
- Class-based weighted fair queuing (CBWFQ) on the WAN ports.
- Low latency queuing (LLQ) on the WAN ports.

  For information on configuring QoS on the OSMs, refer to Chapter 9, "Configuring QoS on the Optical Services Modules."

  For general information on classification, marking, and queuing in IOS, refer to the "Classificiation" chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.1 at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/index.htm

  For information about platform-independent IOS QoS commands, refer to the *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.1 at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/index.htm

Configuring the Channelized OC-12/T1 OSMs

This chapter includes these sections:

Configuring the SONET Controller

To configure the SONET controller, peform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller sonet slot/port`	Selects a port and enters controller configuration mode.
Step 3	`Router(config-controller)# [no] framing {sonet | sdh}`	Configures the framing mode to SONET or SDH. SDH is the ITU standards equivalent of SONET. SONET is the default.
Step 4	`Router(config-controller)# [no] channelized`	Configures the link for channelization or clear channel POS. The **no channelized** command configures the link for clear channel OC-12 POS. The **channelized** command unprovisions the OC-12 POS clear channel.
Step 5	`Router(config-controller)# sts-1 {1-12} interface type`	Provisions the channels for serial or POS interface.
Step 6	`Router(config-controller)# clock source {internal | line | common [internal | primary | secondary]}`	Configures the clock source used by the SONET controller. • **internal**—sets sonet port to internal clock • **line**—sets sonet port to line (network) clock • **common internal**—sets internal clock for common telcom bus • **common primary**—sets port as primary source for common telcom bus • **common secondary**—sets port as secondary source for common telcom bus The network clocking source is the default.

	Command	Purpose
Step 7	`Router(config-controller)# [no] loopback {local | network}`	Enables or disables loopback mode on a SONET controller. • **local**—Sets the loopback after going through the framer toward the terminal. • **network**—Data is looped from the external port to the transmit port and back out the external port. No loopback enabled is the default.
Step 8	`Router(config-controller)# alarm-report {all | b1-tca | b2-tca | b3-tca | lais | lrdi | plop | pplm | prdi |ptim | puneq | sd-ber | sf-ber | slof | slos}`	Optional) Enables alarm reporting.
Step 9	`Router(config-controller)# threshold {b1-tca value | b2-tca value| b3-tca value| sd-ber value| sf-ber value}`	(Optional) Sets BER threshold values.
Step 10	`Router(config-controller)# ais-shut`	(Optional) Specifies an Alarm Indication Signal (AIS) to be sent when a POS interface is shut down.
Step 11	`Router(config-controller)# overhead [j0 {0-255 | expect 0-255} | s1s0 {0-3 | ignore}]`	(Optional) Sets the overhead bytes. **j0**—sets the j0 overhead bytes to transmit a number between 0 to 255 or to expect to receive a number between 0 to 255. **s1s0** —sets the s1s0 bits of H1 to a number between 0-3 or to ignore the s1s0 overhead bit.
Step 12	`Router(config-controller)# [no] description string`	(Optional) Specifies up to 80 characters of text describing the SONET controller. No description is the default.

Note The **clock source** [**internal** | **line**] commands set the OC-12 or OC-3 port's SONET transmit clock source to either a local on-board Stratum-3 oscillator or to the port's received line clock. The **clock source common** [**internal** | **primary** | **secondary**] commands set the clock source for a telcom bus that interconnects logic inside the board. The **internal** command option sets the bus clock to the local Stratum-3 oscillator. The **primary** and **secondary** command options set the bus clock to the port's received line clock. One port will be set to primary and the other port set to secondary. As long as the primary port does not experience a loss of signal (LOS), its received line clock will be the telcom bus clock source. If the primary port experiences a LOS, the telcom bus clock is switched to the secondary port. The clock source switches back to the primary port when it recovers from the LOS. If no s econdary port has been defined, or if both ports have a LOS, then the bus clock source iswitches to the local Stratum-3 oscillator until the LOS condition is cleared. To avoid pointer adjustments on a port, its transmit clock should be from the same source as the common clock, that is, **clock source internal** and **clock source common internal** or **clock source line** and **clock source common primary**.

Configuring STS-1 Path Attributes under SONET Framing

To enter STS-1 path configuration mode to configure an STS-1 path under SONET framing, perform the following task:

	Command	Purpose
Step 1	`Router(config)# controller sonet slot/port`	Selects a port and enters controller configuration mode.
Step 2	`Router(config-controller)# [no] framing {sonet | sdh}`	Configures the framing mode to SONET or SDH. SDH is the ITU standards equivalent of SONET. SONET is the default.
Step 3	`Router(config-controller)# sts-1 {number | number range POS}`	Enters STS-1 path configuration mode. Use the number range option is for OC-3 POS link configuration.
Step 4	`Router(config-crtlr-sts1)# [no] mode {ct3 | t3 | vt-15 }`	Specifies the mode of operation for the STS-1. Use the **no mode ct3** | **t3** command to unprovision sts-1 links configured for these modes.

When you select **ct3**, the specified STS-1 will carry a DS3 signal divided into 28 T1s (multiplexed asynchronously). When you select **t3**, the specified STS-1 will carry an unchannelized T3 signal. When you select **vt-15**, the specified STS-1 is divided into seven virtual tributary groups (VTGs). Each of those VTGs is then divided into four VT1.5s, each carrying one T1. To configure the T1 lines, use the STS-1 path configuration commands described in the "Configuring the T1 Lines" section on page 6-17.

The example that follows selects **vt-15** as the STS-1 mode of operation:

```
Router(config)# controller sonet 6/1
Router(config-controller)# framing sonet
Router(config-controller)# sts-1 3
Router(config-ctrlr-sts1)# mode vt-15
```

Configuring the POS Interface

After you have configured a POS link from the controller level, you can configure the POS interface. The configuration below is basic, and you may need to specify additional interface parameters depending on your network requirements.

To configure the POS interface, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# interface POS slot/port:channel#`	Specifies the serial port and channel to configure.
Step 3	`Router(config-if)# encapsulation hdlc | ppp`	Specifies the encapsulation type.
Step 4	`Router(config-if)# pos flag j1 {expect message message-string | length 16-64 | message message-string}`	(Optional) Specifies a path message for a channelized interface.
Step 5	`Router(config-if)# ip address ip-address mask [secondary]`	Assigns an IP address and subnet mask to the interface.
Step 6	`Router(config-if)# no shutdown`	Enables the interface.

Configuring T3 Links Under SONET Framing

To select T3 as the STS-1 mode ofOSMs, Channelized OC-12/T1 configuration: operation before you can configure the T3 links, perform this task:

	Command	Purpose
Step 1	`Router(config-controller)# [no] framing sonet`	Configures the framing mode to SONET.
Step 2	`Router(config-controller)# sts-1 number`	Enters STS-1 path configuration mode
Step 3	`Router(config-crtlr-sts1)# [no] mode t3`	Specifies T3 as the mode of operation for the STS-1
Step 4	`Router(config-crtlr-sts1)# t3 framing [auto-detect | c-bit | m23]`	Specifies the framing type for the T3 link. Use the **no** version of the command to unprovision the link.
Step 5	`Router(config-crtlr-sts1)# t3 loopback {local | network | remote | line | payload}`	(Optional) Sets the loopback mode.

```
Router(config)# controller sonet 6/1
Router(config-controller)# framing sonet
Router(config-controller)# sts-1 3
Router(config-ctrlr-sts1)# mode t3
Router(config-ctrlr-sts1)# t3 framing auto-detect
Router(config-ctrlr-sts1)# t3 loopback network line
```

Configuring the Unchannelized and Subrate DS-3 Serial Interface

After you verify the controller configuration, you can configure the associated DS-3 interfaces on the controller.

To configure the unchannelized or subrate DS-3 serial interfaces, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# interface serial slot/port`	Specifies the serial port to configure.
Step 3	`Router(config-if)# framing {c-bit | m23}`	Specifies the framing.
Step 4	`Router(config-if)# [no] dsu bandwidth Kilobits/sec`	Sets the DSU subrate bandwidth.
Step 5	`Router(config-if)# encapsulation hdlc | ppp`	Specifies the encapsulation type.
Step 6	`Router(config-if)# [no] loopback {local | network | remote | line | payload}`	(Optional) Sets the loopback mode. Default is **no loopback**.
Step 7	`Router(config-if)# [no] bert pattern [2^11 | 2^15 | 2^20 O.153 | 2^20 QRSS | 2^23 | 0s | 1s | alt-0-1] interval [1-1440]`	(Optional) Configures bit-error-rate (BER) testing.

	Command	Purpose
Step 8	`Router(config-if)# [no] mdl string {eic | fic | generator | lic | pfi | port | unit}`	(Optional) Specifies the maintenance data link (MDL) messages. **eic**—equipment ID code **fic**—frame ID code **generator**—generator number in MDL test signal **lic**—location ID code **pfi**—facility ID code in MDL path message **port**—port number in MDL idle string message **unit**—unide code Default is **no mdl string**.
Step 9	`Router(config-if)# [no] mdl transmit {path | idle-signal | test-signal}`	Enables MDL message transmission. Default is **no mdl transmit**.
Step 10	`Router(config-if)# [no] cablelength` *feet*	Specifies the cable length. Default is 224.
Step 11	`Router(config-if)# ip address` *ip-address mask* `[secondary]`	Assigns an IP address and subnet mask to the interface.
Step 12	`Router(config-if)# [no] keepalive`	Turns on and off keepalive messages.
Step 13	`Router(config-if)# no shutdown`	Enables the interface.

This is an exmple of an unchannelized DS-3 interface configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# controller sonet 3/0
Router (config-controller)# exit
Router (config)# interface serial 1/0
Router (config-if)# dsu bandwidth 16000
Router (config-if)# encapsulation frame-relay
Router (config-if)# ip address 10.10.10.10.255.255.255.255
Router (config-if)# no shutdown
Router (config-if)# exit
Router(config)#
```

Configuring CT3 Links Under SONET Framing

To select **ct3** as the STS-1 mode of operation before you can configure the DS-3 lin, perform this task:

	Command	Purpose
Step 1	`Router(config-controller)# [no] framing sonet`	Configures the framing mode to SONET.
Step 2	`Router(config-controller)# sts-1` *number*	Enters STS-1 path configuration mode
Step 3	`Router(config-crtlr-sts1)# [no] mode ct3`	Specifies the mode of operation for the STS-1. Use the **no** version of the command to unprovision the link.
Step 4	`Router(config-crtlr-sts1)# t3 framing [auto-detect | c-bit | m23]`	Specifies the framing type for the T3 link.
Step 5	`Router(config-crtlr-sts1)# t3 loopback {local | network | remote {line | payload}}`	(Optional) Sets the loopback mode.

```
Router(config)# controller sonet 6/1
Router(config-controller)# framing sonet
Router(config-controller)# sts-1 3
Router(config-ctrlr-sts1)# mode ct3
Router(config-ctrlr-sts1)# t3 framing auto-detect
Router(config-ctrlr-sts1)# t3 loopback network line
```

Configuring the T1 Lines

After you have selected **ct3** as the STS-1 mode of operation, you can configure the T1 link using the following steps:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller` *slot/port*	Selects the controller.
Step 3	`Router (config-controller)# framing sonet`	Configures the framing mode to SONET.
Step 4	`Router(config-controller)# sts-1` *number*	Enters STS-1 path configuration mode
Step 5	`Router(config-crtlr-sts1)# [no] mode ct3`	Specifies the mode of operation for the STS-1. Use the **no** version of the command to unprovision the link.
Step 6	`Router (config-crtlr-sts1)# [no] t1` *number* `clock source {internal | line}`	Defines clock source for the specified T1 line.
Step 1	`Router(config-crtlr-sts1)# t1` *t1-line-number* `channel-group` *channel-group-number* `timeslots` *list-of-timeslots* `[speed {56 | 64}]`	Creates a logical channel group on a T1 line
Step 2	`Router(config-crtlr-sts1)# t1` *t1-line-number* `framing {esf | sf [hdlc-idle {0x7E | 0xFF}]}`	Specifies the T1 framing format.
Step 3	`Router(config)# [no] interface serial` *slot/port.sts1 number/T1 number: channel-group number*	Selects the interface and configures the channel group.

The following example configures 4 T1s for CT3 operation:

```
Router(config-crtlr-sts1)# t1 1 channel-group 0 timeslots 1-24
Router(config-crtlr-sts1)# t1 1 framing esf
Router(config-crtlr-sts1)# t1 1 clock source line
Router(config-crtlr-sts1)# exit
```

Configuring VT-15 Links Under SONET Framing

To select **vt-15** as the STS-1 mode of operation before you can configure the T1 links, perform this task:

	Command	Purpose
Step 1	`Router(config-controller)# [no] framing sonet`	Configures the framing mode to SONET.
Step 2	`Router(config-controller)# sts-1` *number*	Enters STS-1 path configuration mode
Step 3	`Router(config-crtlr-sts1)# mode vt-15`	Specifies VT-15 as the STS-1 mode of operation.

	Command	Purpose
Step 4	`Router(config-ctrlr-sts1)# #vtg 1 t1 1 bert channel-group 0 pattern 2^11 interval 1`	(Optional) Configures bit error rate testing on channel-group 0 (all timeslots under channel group 0).
Step 5	`Router(config-ctrlr-sts1)# #vtg 1 t1 1 bert timeslots 21,24 pattern 2^11 interval 1`	(Optional) Configures bit error rate testing on timeslots 21 and 24 only.

This example shows BERT timeslots:

```
show controller t3 - t1 section
Shows the timeslots that the bert is running/finished

  T1 1 is up
  timeslots: 1-24
  FDL per AT&T 54016 spec.
  No alarms detected.
  Framing is ESF, Clock Source is Internal
  BERT done on timeslots 1,2,3,4,5,6,7,8,9,10<-----Timeslots shown.
  BERT test result (done)
     Test Pattern : 2^11, Status : Not Sync, Sync Detected : 1
     Interval : 1 minute(s), Time Remain : 0 minute(s)
     Bit Errors (since BERT started): 0 bits,
     Bits Received (since BERT started): 85 Mbits
     Bit Errors (since last sync): 0 bits
     Bits Received (since last sync): 85 Mbits
```

Configuring the T1 Links in VT-1.5 Mapping

To configure the T1 links under VT-15 mapping, perform this task:

	Command	Purpose
Step 1	`Router(config-crtlr-sts1)# vtg` *vtg-number* `t1` *t1-line-number* `channel-group` *channel-group-number* `timeslots` *list-of-timeslots* [`speed` {`56` | `64`}]	Creates a logical channel group on a T1 line
Step 2	`Router(config-crtlr-sts1)# vtg` *vtg-number* `t1` *t1-line-number* `framing` {`esf` | `sf` [`hdlc-idle` {`0x7E` | `0xFF`}]}	Specifies the T1 framing format.
Step 3	`Router(config-crtlr-sts1)# [no] vtg` *vtg-number* `t1` *t1-line-number* `clock` *source* {`internal` | `line`}	Sets the internal or line (network) clock source. Use the **no** version of the command to unprovision the link.

The following example configures 4 T1s for VT1.5 operation:

```
Router(config-crtlr-sts1)# vtg 1 t1 1 channel-group 0 timeslots 1-24
Router(config-crtlr-sts1)# vtg 1 t1 1 framing esf
Router(config-crtlr-sts1)# vtg 1 t1 1 clock source line
Router(config-crtlr-sts1)# exit
```

Configuring Interfaces Using SDH Framing with AU-3 Mapping

This section describes how to enable an interface under SDH framing with AU-3 mapping and specify IP routing on the channelized 1-port OSM-CHOC12/T1-SI modules. To configure interfaces using SDH framing with AU-3 mapping, peform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller sonet slot/port`	Selects the controller.
Step 3	`Router(config-controller)# framing sdh`	Specifies the framing.
Step 4	`Router(config-controller)# aug mapping au-3`	Specifies the AUG mapping.
Step 5	`Router(config-controller)# au-3 au-3 number`	Provisions the AU-3 channel.
Step 6	`Router(config-ctrlr-au3)# mode c-12 | ct3-e1 | e3 | t3`	(Optional) Specifies the channelization mode for the link.
Step 7	`Router(config-ctrlr-au3)# [no] tug-2 number e1 number`	(Optional) Specifies Tug-2 configuration mode for the link. Use the **no** version of this command to unprovision the link.
Step 8	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 9	`Router(config)# interface serial slot/port:au-3 number`	Selects the interface.
Step 10	`Router(config-if)# ip address ip-address mask [secondary]`	Assigns an IP address and subnet mask to the interface.
Step 11	`Router(config-if)# [no] shutdown`	Enables the interface.

In this example, a link is configured for 12 T3 channels:

Step 1 Enter the **configure terminal** EXEC command to enter global configuration mode as follows:

```
router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Step 2 Provisision T3 channels:

```
Router(config)# controller sonet 4/1
Router(config-controller)# framing sdh
Router(config-controller)# overhead s1s0 2
Router(config-controller)# aug mapping au-3
Router(config-controller)# au-3 1
Router(config-ctrlr-au3)# mode t3
Router(config-ctrlr-au3)# au-3 2
Router(config-ctrlr-au3)# mode t3
Router(config-ctrlr-au3)# au-3 3
Router(config-ctrlr-au3)# mode t3
Router(config-ctrlr-au3)# au-3 4
Router(config-ctrlr-au3)# mode t3
Router(config-ctrlr-au3)# au-3 5
Router(config-ctrlr-au3)# mode t3
Router(config-ctrlr-au3)# au-3 6
Router(config-ctrlr-au3)# mode t3
Router(config-ctrlr-au3)# au-3 7
Router(config-ctrlr-au3)# mode t3
Router(config-ctrlr-au3)# au-3 8
Router(config-ctrlr-au3)# mode t3
Router(config-ctrlr-au3)# au-3 9
Router(config-ctrlr-au3)# mode t3
Router(config-ctrlr-au3)# au-3 10
Router(config-ctrlr-au3)# mode t3
```

```
Router(config-ctrlr-au3)# au-3 11
Router(config-ctrlr-au3)# mode t3
Router(config-ctrlr-au3)# au-3 12
Router(config-ctrlr-au3)# mode t3
```

Configuring Interfaces Using SDH Framing with AU-4 Mapping

This section describes how to enable an interface under SDH framing with AU-4 mapping and specify IP routing on the channelized 1-port OSM-CHOC12/T1-SI modules. In this example, a port is configured as 12 DS-3 interfaces.

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller sonet` *slot/port*	Selects the controller.
Step 3	`Router(config-controller)# framing sdh`	Specifies the framing.
Step 4	`Router(config-ctrlr-au-4)# aug mapping au-4`	Specifies the AUG mapping.
Step 5	`Router(config-controller)#` **au-4** *au-4 number* {**overhead** | **pos** | **tug-3**}	Provisions the au-4 channel.
Step 6	`Router(config-ctrlr-tug3)# mode` {**c-12** | **e3**} | **tug-2** *number* **e1** *number*	Specifies the TUG-3 channelization mode.

This example configures the first TUG-3 of the AU-4 in port 6/1:

Step 1 Enter the **configure terminal** EXEC command to enter global configuration mode as follows:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Step 2 Configure the SONET controller:

```
Router(config)# controller sonet 5/1
Router(config-controller)# framing sdh
```

Step 3 Specify the AUG mapping:

```
Router(config-controller)# aug mapping au-4
```

Step 4 Select a TUG-3 for configuration:

```
Router(config-controller)# au-4 1 tug-3 1
```

Step 5 Speficy the channelization mode:

```
Router(config-ctrlr-tug3)# mode e3
```

CHAPTER 1

Configuring the Channelized 12-port CT3/T1 Optical Services Modules

This chapter describes how to configure the 12-port channelized/unchannelized DS3 Optical Services Modules (OSM-12CT3/T1).

The chapter consists of these sections:

Understanding the Channelized/Unchannelized CT3/T1 Modules

The OSM-12CT3/T1 module provides 12DS3 interface connections using mini-SMB connectors.

Each DS3 port can be independently configured all the way from a clear channel DS3 down to and including DS1, E1,and DS0 connections on the same port, with a maximum of 128 channels per DS3 port.

Each OSM-12CT3/T1 can support any combination of DS3/DS1/E1/T1s up to 1023 channels. For example, each port can support the following:

- 1 DS3 at 44.7 Mbps
- 21 E1s at 2.048Mbps
- 28 DS1s at 1.5 Mbps
- 128 DS0s at 64 Kbps

An unchannelized DS3 connection provides a single serial interface data channel that may be configured to use all of the DS3 bandwidth or a fractional portion of it. This mode is compatible with several vendors of fractional (subrate) DS3 data service units (DSUs).

A cost, depending on the bandwidth, is associated with a serial interface. The total cost possible for 12 DS3 ports is 4096. That is, the sum of the costs of all the serial interfaces configured on 12 DS3 ports must be no more than 4096. The **show controller t3** command displays the VC cost, which indicates the available cost of 12 DS3 ports. Table 7-1 shows the cost associated with each link type.

Table 1-1 Serial Interface Cost

Link Type	Cost
DS3	336
Unframed E1	16
1-6 DS-/TS	4
7-9 DS0/TS	6
10-16 DS0/TS	8
17-24 DS0/TS	12
25-31 TS	16

Channelized DS3 Overview

In the channelized mode of operation, an OSM-12CT3/T1 link is channelized into 28 DS1 or 21 E1 data lines in an industry standard multiplexing format.

Each of the DS1 lines contain 24 timeslots of 64 or 56 kbps each, and each of the E1 lines contain 32 timeslots of 64 or 56 kbps each. The DS1and E1 lines can support one or more user data channels which appear to the system as serial interfaces. Each serial interface is assigned one or more of the timeslots giving the serial interface a bandwidth of n x 56 kbps or n x 64 kbps, where n is the number of timeslots assigned. Any unused timeslots of the lines are filled with an idle channel pattern.

Unchannelized DS3 Overview

In the unchannelized mode of operation, a DS3 link provides a single high speed user data channel, rather than being multiplexed into 28 DS1 or 21 E1 lines. The data channel appears to the system as a serial interface that may be configured to use the full DS3 bandwidth or a smaller portion of the DS3 bandwidth.

In unchannelized DS3 mode, the OSM-12CT3/T1 supports the maintenance data link (MDL) channel when using c-bit parity framing as well as local and network loopbacks.

Supported Features

The 12-port channelized/unchannelized DS3 OSMs support the following features:

- General Features, page 7-3
- Serial Encapsulation Protocols, page 7-3
- DSU Mode, page 7-4
- T1 Configuration Options, page 7-4
- E1 Configuration Options, page 7-4
- DS3 Alarms, page 7-5
- Network Management, page 7-5
- Quality of Service Protocols, page 7-5

General Features

The general features are as follows:

- Channelized DS3 with 28 DS1 lines or 21 E1 lines multiplexed into a DS3 line.
- C-bit parity, M23, and auto-detect framing.
- Internal clocking configurable.
- Bit error rate test (BERT) diagnostics for each DS-3 channel
- Local, line, and remote loopback.
- Generation and termination of DS-3 maintenance data link (MDL) in C-bit framing.
- Compliance with ANSI T1.231-1993, Bellcore GR820, ANSI T1.107, ANSI T1.403 1989, G.703 Section 2, G.704 Section 3, AT&T Pub. 62411 1990, and Facilities Data Link (ANSI T1.403 and AT&T 54016).

Serial Encapsulation Protocols

The OSMs support for the following serial encapsulation protocols:

- Point-to-Point Protocol (PPP)
- Distributed Multilink PPP (dMLPPP)

  The OSM-12CT3/T1 module supports dMLPPP, which means that the MLPPP encapsulation is performed on the OSM-12CT3/T1 module rather than the route processor. The following dMLPP features are supported:

  – 336 bundles per module
  – A maximum of 12 DS1/E1 channels per bundle
  – All links in a bundle should operate at the same speed
  – 100 ms of differential delay
- High-Level Data Link Control (HDLC)
- Frame Relay

  Configure the OSM-12CT3/T1 interfaces for Frame Relay as described in the *Cisco IOS Wide-Area Networking Configuration Guide, Release 12.1* under “Configuring Frame Relay” and in the *Cisco IOS Wide-Area Networking Command Reference, Release 12.1* at:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_c/wcdfrely.htm

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_r/wrdfrely.htm

  Configure traffic shaping for Frame Relay as described in the *Cisco IOS Quality of Service Solutions Configuration Guide* under “Configuring Distributed Traffic Shaping” at:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt4/qcfdts.htm.

  For information on Multilink Frame Relay (FRF.16), see Multilink Frame Relay (FRF.16), page 1-6.
- MPLS/VPN

  For MPLS/VPN configuration information, see Chapter 11, “Configuring Multiprotocol Label Switching on the Optical Services Modules.”

DSU Mode

Unchannelized DS-3 supports subrate and scrambling formats for the following DSUs.

- Digital Link
- Verilink
- Adtran
- Larscom
- Kentrox

T1 Configuration Options

The T1 lines have the following configuration options:

- Local, line, and remote loopback.
- Generation and termination of DS-3 maintenance data link (MDL) in C-bit framing.
- In T1 mode, the channelized CT3/T1 OSMs support facilities data link (FDL) in Extended Superframe (ESF) framing, as well as various loopbacks. Bit error rate testing (BERT) is supported on each of the T1 lines. It can be done over a framed or unframed DS-1 signal. The OSM-ChOC12C/T1 module has four framers, each connected to three contiguous ports. BER testing can be done simultenously on 6 T1 interfaces in addition to any DS3 interfaces configured on the three ports to which a framer is connected.
- In the channelized mode of operation, a channelized CT3/T1 OSM link can be channelized into maximum of 336 T1 data lines in an industry standard multiplexing format. Each of the T1 lines contains 24 timeslots of 64 or 56 kbps each. The T1 lines can support one or more user data channels, which appear to the system as serial interfaces. Each serial interface is assigned one or more of the timeslots giving the serial interface a bandwidth of *n* x 56 kbps or *n* x 64 kbps, where *n* is the number of timeslots assigned. Any unused timeslots of the T1 are filled with an idle channel pattern.

E1 Configuration Options

The E1 lines have the following configuration options:

- Channelized E1—Any of the 21 E1 lines can be configured as channelized E1 lines, but you are limited to a total of 128 logical channels. You can group the time slots in these E1 lines into several individual logical channel groups, each of which carries data with different data link layer protocol encapsulations. You can configure timeslot 16 as a data channel, although it is typically used for common channel signaling. (Channel associated signaling (CAS) for voice channels and E1 Facilities Data Link [FDL] on timeslot 16 are not supported.)
- Each logical channel group can be composed of individual 64-kbps timeslots and/or ranges of timeslots, for example, 1, 9, 12-14. Each logical channel group can contain from 1-31 timeslots maximum; the same timeslot cannot be used in more than one logical channel group. Any unused timeslots are filled with programmable idle-channel data.
- Fractional E1— Any of the 21 E1 lines can be configured as fractional E1 lines, each of which can be either E1 frames or E1 cyclic redundancy check (CRC) multiframes, as specified by CCITT/ITU G.704 and G.706. A fractional E1 line is a subset of the full E1 bandwidth, which uses *n* x 64 kbps; where *n* is a timeslot in the range of 1-31.

Fractional E1 lines contain only a single logical channel group that can be either a single 64-kbps timeslot or a range of timeslots; for example timeslot 1, or timeslots 15-23. Any unused timeslots are filled with programmable idle-channel data.

Note If you assign only one channel group to an E1 line, it is a fractional E1 line. If you assign more than one channel group to an E1 line, it is a channelized E1 line.

- Unframed E1—Any of the 21 E1 lines can be configured as unframed E1 data lines. Each unframed E1 line contains no framing overhead and is not timeslot divided.

Note For channelized, fractional, and unframed configurations each configured channel group, which might contain individual timeslots and/or ranges of timeslots, uses only one of the 128 available logical channels. For example, if you assign the range of timeslots 3-7 to a channel group, only one logical channel is used. Likewise, if you assign just timeslot 3 to a channel group, only one logical channel is used.

DS3 Alarms

This section lists the supported DS3 alarms:

- Alarm indication signal (AIS)
- Out of frame (OOF)
- Far End Remote Failure (FERF)
- Loss of Signal (LOS)
- Far End Block Errors (FEBE)
- Far-End Alarm and Control (FEAC) support

Network Management

This section lists the supported network management features:

- Local Loopback.
- Network Loopback.
- RFC 1407 MIB support.

Quality of Service Protocols

For information on configuring QoS on the Channelized OSMs, refer to Chapter 9, “Configuring QoS on the Optical Services Modules.”

The following QoS features are supported on the Channelized OSMs:

- Hierarchical traffic shaping for Frame Relay, HDLC, and PPP encapsulations
- PFC2 QoS on the LAN and WAN ports
- Differentiated Services Control Point (DSCP)
- IP Precedence classification

Configure class-based marking as described in the *Class-Based Marking* Feature Module at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm

- Classification and priority marking based on the following:
  - Ethertype
  - IP Source Address (SA)
  - IP Destintation Address (DA)
  - TCP port number
  - UDP port number
  - IP SA + TCP/UDP port number + IP DA + TCP/UDP port number
- Class-based weighted fair queuing (CBWFQ) on the WAN ports.
- Low latency queuing (LLQ) on the WAN ports.

For general information on classification, marking, and queuing in IOS, refer to the "Classificiation" chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.1 at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/index.htm

For information about platform-independent IOS QoS commands, refer to the *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.1at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/index.htm

Configuring the Interfaces

This chapter includes these sections:

Configuring the T3 Controller

To configure the controller, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller t3` *slot/port*	Selects a port and enters controller configuration mode.
Step 3	`Router(config-controller)# [no] channelized [mode {t1 | e1}]`	Specifies the channelization mode. **no channelized**—configures the interface as unchannelized. **channelized mode t1**—configures the interfaces for T1 channelized mode **channelized mode e1**—configures the interfaces for E1 channelized mode Default is **t1**.
Step 4	`Router(config-controller)# [no] bert pattern [2^11 | 2^15 | 2^20 O.153 | 2^20 QRSS | 2^23 | 0s | 1s | alt-0-1] interval [1-1440]`	(Optional) Configures bit-error-rate (BER) testing.
Step 5	`Router(config-controller)# [no] mdl string {eic | fic | generator | lic | pfi | port | unit}`	(Optional) Specifies the maintenance data link (MDL) messages. **eic**—equipment ID code **fic**—frame ID code **generator**—generator number in MDL test signal **lic**—location ID code **pfi**—facility ID code in MDL path message **port**—port number in MDL idle string message **unit**—unit code Default is **no mdl string**.
Step 6	`Router(config-controller)# [no] mdl transmit {path | idle-signal | test-signal}`	(Optional) Enables MDL message transmission. Default is **no mdl transmit**.
Step 7	`Router(config-controller)# t1 1 bert channel-group 0 pattern 2^11 interval 1`	(Optional) Configures bit error rate testing on channel-group 0 (all timeslots under channel group 0).
Step 8	`Router(config-controller)# t1 1 bert timeslots 21,24 pattern 2^11 interval 1`	(Optional) Configures bit error rate testing on timeslots 21 and 24 only.

This example shows how to configure the controller:

```
Router# configure terminal
Router(config)# controller T3 3/0
Router(config-controller)# no channelized
Router(config-controller)# exit
Router(config)#
```

Configuring the Unchannelized DS3 Interface

After you verify the controller configuration, you can configure the associated DS3 interfaces.

Note IP addresses can not be assigned to main interfaces configured for Frame Relay.

To configure the unchannelized DS3 interfaces, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# interface serial` *slot/port*	Specifies the serial port to configure.
Step 3	`Router(config-if)# framing {c-bit | m23}`	Specifies the framing.
Step 4	`Router(config-if)# [no] dsu bandwidth` *Kilobits/sec*	Sets the DSU subrate bandwidth.
	`Router(config-if)# [no] dsu mode {0|1|2|3|4}`	Specifies the DSU mode: 0–Digital-Link 1–Kentrox 2–Larscom 3–Adtran 4–Verilink
Step 5	`Router(config-if)# [no] dsu remote [accept | fullrate]`	Specifies if the local (near-end) interface will accept incoming requests from the remote (far-end) interface, or if the local interface will request that the remote interface set its bandwidth to fullrate.
Step 6	`Router(config-if)# encapsulation hdlc | ppp`	Specifies the encapsulation type. HDLC is the default encapsulation.
Step 7	`Router(config-if)# [no] clock source {internal | line}`	(Optional) Specifies the clock source. Default is **line**.
Step 8	`Router(config-if)# [no] loopback {local | network | remote {line | payload}}`	(Optional) Sets the loopback mode. Default is **no loopback**.
Step 9	`Router(config-if)# [no] cablelength` *feet*	Specifies the cable length. Default is 224.
Step 10	`Router(config-if)# [no] keepalive`	Turns on and off keepalive messages.
Step 11	`Router(config-if)# no shutdown`	Enables the interface.

This is an example of an unchannelized DS3 interface configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# controller t3 3/0
Router (config-controller)# no channelized
Router (config-controller)# exit
Router (config)# interface serial 1/0
Router (config-if)# dsu bandwidth 16000
Router (config-if)# encapsulation ppp
Router (config-if)# no shutdown
Router (config-if)# exit
Router(config)#
```

Configuring the Channelized DS3 Interface

The DS3 Interface can be channelized down to T1 and E1 links. These sections provide configuration information for T1 and E1 link configuration:

- Configuring the T3 Controller for Channelization, page 7-9
- Configuring the T1/NxDS0 Lines, page 7-9
- Configuring the E1 Lines, page 7-10

Note Bit error rate testing (BERT) is supported on each of the T1 and E1 lines. It can be done over a framed or unframed DS-1 signal. The OSM-12CT3/T1 module has four framers, each connected to three contiguous ports. BER testing can be done simultenously on 6 T1 or E1 interfaces in addition to any DS3 interfaces configured on the three ports to which a framer is connected.

Note BERT channel-groups and timeslots are also supported.

Configuring the T3 Controller for Channelization

To configure the T3 controller for T1 of E1 channelization, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller` *slot/port*	Selects the controller.
Step 3	`Router (config-controller)# [no] channelized [mode {t1 | e1}]`	Specifies the channelization mode. Default is **t1**.

This example configures the T3 controller for T1 channelization mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# controller T3 2/0
Router(config-controller)# channelized mode t1
Router(config-controller)# CNTL/Z
Router#
```

Configuring the T1/NxDS0 Lines

To configure T1/NxDS0 lines, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller t3 slot/port`	Enters controller configuration mode.
Step 3	`Router(config-controller)# [no] t1 t1-number channel-group channel-group-number timeslots list-of-timeslots [speed 56 | 64]`	Configures the channel group.
Step 4	`Router (config-if)# [no] t1 t1-number framing {sf | esf}`	Specifies the framing.
Step 5	`Router (config-controller)# [no] t1 t1-number fdl ansi`	(Optional) Enables the once-second transmission of the remote performance reports via facility data link (FDL).
Step 6	`Router (config-controller)# [no] t1 t1-number yellow {detection | generation}`	(Optional) Enables detection and generation of T1 yellow alarms.
Step 7	`Router (config-controller)# [no] t1 t1-number clock source {internal | line}`	(Optional) Defines clock source for the specified T1 line.
Step 8	`Router (config-controller)# [no] t1 t1-number loopback {local | network | remote {line fdl {ansi | bellcore}| payload fdl ansi}}`	(Optional) Sets the loopback mode.
Step 9	`Router (config-controller)# [no] [e1 | t1] [e1-number | t1 number] bert pattern {2^11 | 2^15 | 2^20 O.153 | 2^20 QRSS | 2^23 | 0s | 1s | alt-0-1} interval [1-1440]`	(Optional) Configures bit-error-rate (BER) testing.
Step 10	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 11	`Router(config)# [no] interface serial slot/port/t1 number: channel-group number`	Selects the interface and configures the channel group.

This example configures a T1/NxDS0 line:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# controller T3 3/1
Router(config-controller)# t1 6 channel-group 0 timeslots 1-5, 20-23
Router(config-controller)# t1 6 framing sf
Router(config-controller)# t1 6 clock source line
Router(config-controller)# t1 6 loopback remote line
Router(config-controller)# t1 6 bert pattern 2^23 interval 5 unframed
Router(config-controller)# exit
Router(config)# interface serial t1 3/1/6:0
Router (config-if)# CNTL/Z
Router#
```

Configuring the E1 Lines

To configure E1 lines, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller slot/port`	Selects the controller.
Step 3	`Router (config-controller)# [no] channelized [mode {t1 | e1}]`	Specifies the channelization mode. Default is **t1**.
Step 4	`Router (config-controller)# [no] e1 e1number clock source {internal | line}`	Defines clock source for the specified E1 line.
Step 5	`Router (config-controller)# [no] e1 e1 number unframed`	Configures the E1 line as unframed.
Step 6	`Router (config-controller)# [no] e1 e1 number framing {crc4 | no-crc4}`	Specifies the framing.
Step 7	`Router (config-controller)# [no] e1 e1 number national bits pattern`	Specifies the national bits reserved for country-specific control information.
Step 8	`Router (config-controller)# [no] e1 e1 number channel-group channel-group number timeslots list-of-timeslots speed [56 | 64]`	Configures a channel-group.
Step 9	`Router (config-controller)# [no] e1 e1 number loopback {local | network}`	(Optional) Sets the loopback mode.
Step 10	`Router (config-controller)# [no] [e1 | t1] [e1 number | t1 number] bert pattern {2^11 | 2^15 | 2^20 O.153 | 2^20 QRSS | 2^23 | 0s | 1s | alt-0-1} interval [1-1440]`	(Optional) Configures bit-error-rate (BER) testing.
Step 11	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 12	`Router(config)# [no] interface serial slot/port/t1 number: channel-group number`	Selects the interface and configures the channel group.

This example configures an E1 line:

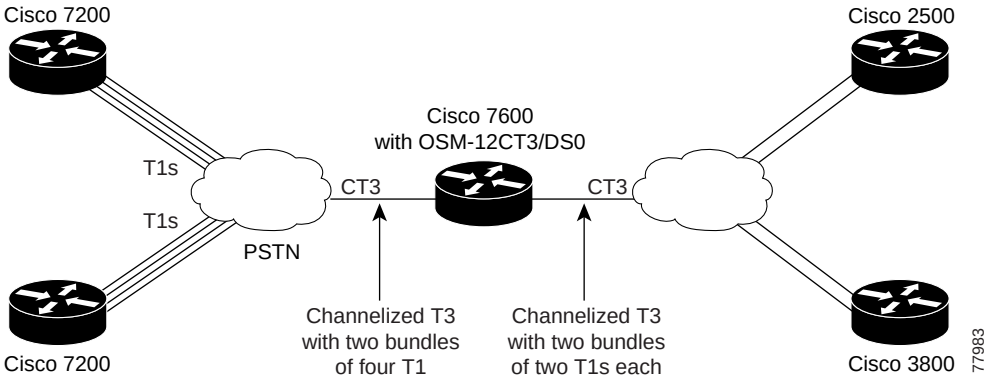
```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router)# controller T3 4/1
Router(config-controller)# channelized mode e1
Router(config-controller)# e1 10 channel-group 0 timeslots 1-5, 20-23
Router(config-controller)# e1 10 framing crc4
Router(config-controller)# e1 10 national bits
Router(config-controller)# e1 10 clock source line
Router(config-controller)# e1 10 loopback network line
Router(config-controller)# e1 10 bert pattern 2^23 interval 5 unframed
Router (config-controller)# exit
Router(config)# interface serial e1 4/0/10:0
Router(config-if)# CNTL/Z
Router#
```

Configuring Distributed MLPPP

Distributed MLPPP allows you to combine T1 or E1 lines into a bundle that has the combined bandwidth of multiple T1 or E1 lines. Bundling T1 or E1 lines enables you to increase the bandwidth of your network links beyond that of a single T1 or E1 line without having to purchase a T3 line.

Figure 7-1 shows a network using an MLPPP link. The Cisco 7600 Series router is connected to the network with an OSM-12CT3/T1 module. An OSM-12CT3/DS interface has been configured with MLPPP to carry two bundles of four T1 lines each. Each of these bundles goes out to separate remote Cisco 7200 series routers, which each have one MLPPP bundle of four T1 lines. Another OSM-12CT3/DS interface has been configured with MLPPP to carry two bundles of two T1 lines. One of these bundles goes out to a Cisco 2500 series router and the other goes out to a Cisco 3800 series router.

Figure 1-1 MLPPP Topology



Creating a Multilink Bundle

To create a multilink bundle, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface multilink` *bundle-id*	Statically configures link bundles. **Note** The Multilink PPP command is available on serial interfaces only.
Step 2	`Router(config-if)# ip address` *address mask*	Assigns an IP address to the multilink interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# ppp multilink`	(Optional) Enables Multilink PPP.
Step 5	`Router(config-if)# [no] multilink-group` *bundle-id*	(Optional) Designates an interface as part of a multilink bundle. *bundle-id* is the bundle number created with the **interface multilink** *bundle-id* command.

The following example shows how to create a multilink bundle:

```
Router(config)# interface multilink 1
Router(config-if)# ip address 10.0.0.0 10.255.255.255
Router(config-if)# encapsulation ppp
Router(config-if)# ppp multilink
Router(config-if)# multilink-group 1
Router(config-if)# CNTL/Z
Router#
```

Assigning an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial slot/port/DS0 number: channel-group number`	Selects the interface.
Step 2	`Router(config-if)# no ip address`	Removes any specified IP address.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# multilink-group bundle-id`	Designates an interface as part of a multilink bundle. *bundle-id* is the bundle number created with the **interface multilink** *bundle-id* command.
Step 5	`Router(config-if)# ppp authentication chap`	(Optional) Enables Challenge Handshake Authentication Protocol (CHAP) authentication.
Step 6	`Router(config-if)# ppp chap hostname name`	(Optional) Enables the presence of multiple bundles between two hosts.

The following example shows how to create a multilink interface and add it to a multilink bundle:

```
Router(config)# interface serial 1/0/0:1
Router(config-if)# no ip address
Router(config-if)# encapsulation ppp
Router(config-if)# multilink-group 1
Router(config-if)# CNTL/Z
Router#
```

Enabling PPP Multilink Fragmentation

By default, PPP multilink fragmentation is enabled. To enable PPP multilink fragmentation after it has been disabled, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# [no] ppp multilink fragmentation`	Enables fragmentation. By default fragmentation is enabled.
`Router(config-if)# [no] multilink fragment-delay milliseconds`	Specifies the tolerable delay in sending the fragment on the multilink. This delay is used to compute fragment size of the packet. Three supported fragment sizes are 128, 256, 512. Default delay is 30ms.

Enabling fragmentation reduces the delay latency among bundle links, but adds some load to the CPU. Disabling fragmentation may result in better throughput.

If your data traffic is consistently of a similar size, we recommend disabling fragmentation. In this case, the benefits of fragmentation may be outweighed by the added load on the CPU.

Configuring Multilink PPP Minimum Links Mandatory

To configure the Multilink PPP Minimum Links Mandatory feature, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ppp multilink`	Enables Multilink PPP.
Step 2	`Router(config-if)# multilink min-links` *links* `mandatory`	Specifies the required minimum number of links in a Multilink PPP (MLP) bundle. If the minimum number of links in the MLP bundle falls below the number specified by the *links* attribute, the MLP bundle is disabled. • *links*—Minimum number of links, in the range 0 to 12. • The **mandatory** keyword is required on the OSM-12CT3/T1.

This example shows how to configure the Multilink PPP Minimum Links Mandatory feature:

```
Router(config-if)# ppp multilink
Router(config-if)# multilink min-links 5 mandatory
```

For additional information about the Multilink PPP Minimum Links Mandatory feature, refer to this URL:

```
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121limit/121e/121e11/12e_mlp.htm#xtocid0
```

The commands listed at this URL are all supported on the OSM-12CT3/T1, except the following:

- **multilink max-link**
- **multilink load-threshold**
- **multilink max-fragments**
- **debug ppp multilink fragments**

Verifying the Configuration

Use the **show ppp multilink** command to display information about the newly created multilink bundle:

```
Router# show ppp multilink

Multilink1, bundle name is group1
Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links:4 active, 0 inactive (max not set, min not set)
 Serial1/0/0:1
```

Creating a Multilink Interface Example

```
Interface multilink1
 ip address 100.0.0.1 255.255.255.0

interface serial6/12/1:0
 no ip address
 encapsulation ppp
```

```
 ppp chap hostname mul1
 multilink-group 1

interface serial6/12/2:0
 no ip address
 encapsulation ppp
 ppp chap hostname mul1
 multilink-group 1

interface serial6/12/3:0
 no ip address
 encapsulation ppp
 ppp chap hostname mul1
 multilink-group 1

show ppp multilink

Multilink1, bundle name is mul1
  Bundle up for 19:58:37
  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:3 active, 0 inactive (max not set, min not set)
    Serial6/12/1:0, since 19:58:37, no frags rcvd  384 weight
    Serial6/12/2:0, since 19:58:37, no frags rcvd  384 weight
    Serial6/12/3:0, since 19:58:37, no frags rcvd  384 weight
```

CHAPTER 1

Configuring the OC-12 ATM Optical Services Modules

This chapter describes the 2-port OC-12 ATM WAN Optical Services Modules (OSMs).

This chapter consists of these sections:

ATM Overview

Asynchronous Transfer Mode (ATM) uses cell-switching and multiplexing technology that combines the benefits of circuit switching (constant transmission delay and guaranteed capacity) with the benefits of packet switching (flexibility and efficiency for intermittent traffic).

ATM is a connection-oriented environment. All traffic to or from an ATM network is prefaced with a virtual path identifier (VPI) and virtual channel identifier (VCI). A VPI/VCI pair is considered a single virtual circuit. Each virtual circuit is a private connection to another node on the ATM network. Each virtual circuit is treated as a point-to-point mechanism to another router or host and can support bidirectional traffic.

Each ATM node is required to establish a separate connection to every other node in the ATM network that it must communicate with. All such connections are established using a permanent virtual circuit (PVC), which a network operator configures, or a switched virtual circuit (SVC), which is set up and torn down with an ATM signaling mechanism. This signaling is based on the ATM Forum User-Network Interface (UNI) specification V3.x, 4.0.

Supported Features

The WAN ports on the 2-port OC-12 ATM OSMs support the following features:

- Multiprotocol label switching (MPLS) VPNs
- Permanent virtual circuits (PVCs)
- Switched virtual circuits (SVCs); up to 100 SVCs
- Maximum of 1000 VCs per module; 500 per physical ATM interface
- VPI range 0 through 255 (the default is 15)
- VCI range 1 through 1023 (the default is 1023)
- RFC 1577 classical IP over ATM
- RFC 1483 bridging support for PVCs only
- Bridging of 1483 Routed Encapsulations (BRE)
- Hardware switching of multicast packets on point-to-point subinterfaces
- Software switching of multicast packets on point-to-multipoint subinterfaces
- UNI 3.x and UNI 4.0
- ILMI 1.0
- Per-VC Layer 3 queuing
- Layer 3 traffic shaping
  - CIR
  - EIR
- PFC QoS with OSM-2OC12-ATM-MM/SI+
- Per-VP shaping
- Per-VC shaping
- UBR and VBR-NRT
- Per-VC class-based weighted fair queuing (CBWFQ)
- Low latency queuing (LLQ)
- Weighted Random Early Detection (WRED)
- Committed Access Rate (CAR)
- SONET Linear APS 1+1
- Multipoint Bridging
- PVST+ to 802.1d BPDU conversion

Note For the OC-12 ATM OSM, the Common Part Convergence Sublayer User-to-User (CPCS-UU) field in the AAL5 CPCS PDU cannot be set, cleared, or transported correctly. This affects custom use of the field as well as FRF8.1, which uses the CPCS-UU byte to transport the Frame Relay command response (C/R) bit.

For QoS configuration information and examples for the WAN OSM ports, see the "Configuring QoS on the OSMs" section on page 9-2.

For MPLS QoS configuration information and examples for the WAN OSM ports, see the "Configuring MPLS QoS" section on page 11-13.

For general information on how to configure Cisco IOS QoS, refer to these Cisco IOS publications:

Cisco IOS Quality of Service Solutions Configuration Guide at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/index.htm

Cisco IOS Quality of Service Solutions Command Reference at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/index.htm

Configuring the OC-12 ATM Interfaces

This section provides procedures for initial configuration of an OC-12 ATM OSM interface:

Initial Configuration for the OC-12 ATM OSM

On power-on, the interfaces on a new OC-12 ATM OSM are shut down. To enable an interface, you must enter the **no shutdown** command in configuration mode. When the OC-12 ATM interface is enabled with no additional configuration, the default interface configuration file parameters are used. These default parameters are listed in Table 8-1.

Table 1-1 OC-12c/STM-4c ATM Module Configuration Default Values

Parameter	Configuration Command	Default Value
Maximum transmission unit	[**no**] **mtu** *bytes*	4470 bytes
Loopback	[**no**] **loopback** [**diagnostic** | **line**]	no loopback
ATM VCs per VP	**atm vc-per-vp**	1023

After you verify that the new OC-12 ATM module is installed correctly (the active LED goes on and all cables are correctly connected), you can use the **configure** command to configure the ATM interfaces.

Enabling the ATM Interface

A Cisco Catalyst 6000 family switch and Cisco 7600 series router identifies an interface address by its slot number and port number in the format *slot/port*. For example, the slot/port address of an interface on a 2-port OC-12 ATM OSM installed in slot 4 is 4/1.

Before using the **configure** command, you must enter the privileged level mode of the EXEC command interpreter by using the **enable** command. The system prompts you for a password if one is set.

Use the following procedure to configure the 2-port OC-12 ATM OSMs. Press the **Return** key after each configuration step unless otherwise noted.

To configure the ATM interfaces, perform this task:

	Command	Purpose
Step 1	`Router# show module`	Confirms that the system recognizes the module.
Step 2	`Router# show interface atm slot/port`	Checks the status of each port.
Step 3	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of configuration subcommands.
Step 4	`Router(config)# interface atm slot/port`	Specifies the new interface to configure.
Step 5	`Router(config-if)# ip address ip-address mask [secondary]`	Assigns an IP address and subnet mask to the interface.
Step 6	`Router(config-if)# no shutdown` `Router(config-if)# end`	Changes the interface state to up and enables the interface.
Step 7	`Router# copy running-config startup-config`	Writes the new configuration to memory.

This example shows how to configure an OC-12 ATM OSM interface:

```
Router# configure terminal
Router(config)# interface atm 4/0
Router(config-if)# ip address 1.2.3.4 255.255.255.0
Router(config-if)# no shutdown
Router# copy running-config startup-config
```

Valid VCI and VPI Configurations

The default number of VPIs per ATM interface is 15. The maximum number of VCIs per VPI is 1023.

Table 8-2 shows the valid VCs per VP and maximum VPI configurations.

Table 1-2 Valid VCI and VPI Configurations

VCs per VP	Maximum VPIs
1024	15
512	31
256	63
128	127
64	255
32	255
16	255

Configuring the Maximum VCs per VP

The ATM interfaces are configured by default to allow a maximum of 1023 VCs per VP. To change this value, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/port`	Enters interface configuration mode and specifies the ATM interface to configure.
Step 2	`Router(config-if)# atm vc-per-vp`	Configures the maximum number of VCs per VP to 16, 32, 64, 128, 256, 512, or 1024. The default is 1024.
Step 3	`Router(config-if)# no shutdown`	Enables the interface with the above configuration.

Configuring Virtual Connections

This section provides basic information for configuring PVCs, bridged PVCs (RFC 1483), PVC traffic parameters, and SVCs. In addition, this section documents commands and configurations that are unique to the 2-port OC-12 ATM OSMs.

For all other Cisco IOS features and commands supported on the OC-12 ATM OSMs, refer to the "Configuring ATM" chapter in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.1 at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_c/wcdatm.htm

For complete command syntax information, refer to the "ATM commands" chapter in the *Cisco IOS Wide-Area Networking Command Reference*, Release 12.1 at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_r/wratm/index.htm

Creating a PVC

To create a PVC on the ATM interface and enter interface-ATM-VC configuration mode, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/port`	Specifies the new interface to configure.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Assigns an IP address and subnet mask to the interface.
Step 3	`Router(config-if)# pvc [name] vpi/vci [ilmi | qsaal]`	Configures a new ATM PVC by assigning a name (optional) and VPI/VCI numbers. Enters interface-ATM-VC configuration mode. Optionally configures ILMI or QSAAL encapsulation.
Step 4	`Router(config-if-atm-vc)# protocol protocol protocol-address [[no] broadcast]`	Maps a protocol address to a PVC.
Step 5	`Router(config-if-atm-vc)# encapsulation {aal5mux | aal5snap}`	(Optional) Configures the ATM adaptation layer (AAL) and encapsulation type. The default is **aal5snap**.

This example shows how to create a PVC:

```
Router(config)# interface atm 4/0
Router(config-if)# ip address 10.212.13.4 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# pvc cisco 0/56
Router(config-if-atm-vc)# protocol ip 10.212.13.5 broadcast
Router(config-if-atm-vc)# encapsulation aal5snap
Router(config-if-atm-vc)#
```

Configuring Bridging of RFC 1483 Routed Encapsulations

Bridging of routed encapsulations (BRE) enables the OC-12 ATM OSM to receive RFC 1483 routed encapsulated packets and forward them as Layer 2 frames.

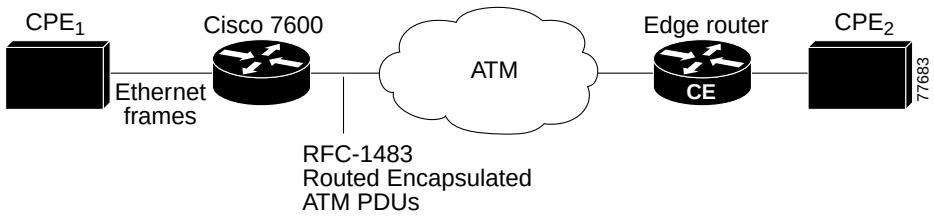
Note Concurrent configuration of RFC 1483 bridging and BRE on the same PVC and VLAN is not supported.

When you configure BRE on an ATM PVC on the OC-12 ATM OSM:

1. The PVC receives the routed PDUs,
2. The PVC removes the RFC 1483 routed encapsulation header,
3. The PVC adds an Ethernet MAC header to the packet.
4. The supervisor engine then switches the Layer 2 encapsulated packet to the Layer 2 interface determined by the VLAN number and destination MAC.

Figure 8-1 shows a topology in which an OC-12 ATM OSM receives routed PDUs, encapsulates them as Layer 2 frames, and forwards these frames to a Layer 2 customer device.

Figure 1-1 Example BRE Topology



To configure BRE on a PVC, use the following commands:

	Command	Purpose
Step 1	`Router(config)# vlan`	Configures the Layer 2 VLAN.
Step 2	`Router(config)# interface atm` *slot/port* `[.`*subinterface-number* `point-to-point]`	Specifies the subinterface on which to configure the PVC.
Step 3	`Router(config-subif)# pvc` [*name*] *vpi/vci*	Configures a new ATM PVC by assigning a name (optional) and VPI/VCI numbers.
Step 4	`Router(config-if-atm-vc)# bre-connect` *vlan* `[ip` *ip address*`]`	Enables BRE on the PVC. Using the **ip** keyword allows the BRE device to generate an ARP request to learn the CPE MAC address the first time a packet needs to be sent from the BRE switch to the CPE.

	Command	Purpose
Step 5	`Router(config)# interface GigabitEthernet slot/port`	Specifies the Gigabit Ethernet port to configure.
Step 6	`Router(config-if)# switchport`	Configures the Gigabit Ethernet port for Layer 2 switching.
Step 7	`Router(config-if)# switchport access vlan vlan_id`	(Optional) Specifies the default VLAN, which is used if the interface stops trunking.
Step 8	`Router(config-if)# switchport mode access`	Places the interface into nontrunking mode.

Step 1 Configure the Layer 2 VLAN.

```
Router# configure terminal
Router(config)# vlan 10
Router(config-vlan)# exit
```

Step 2 Configure the VLAN interface:

```
Router(config)# interface vlan 10
Router(config-if)# no shutdown
Router(config-if)# end
Router#
```

Step 3 Configure the default VLAN on the Ethernet interface:

```
Router(config)# interface GigabitEthernet3/3
Router(config-if)# no ip address
Router(config-if)# switchport
Router(config-if)# switchport access vlan 10
Router(config-if)# switchport mode access
Router(config-if)# end
Router#
```

Step 4 Enable the ATM main interface:

```
Router(config)# interface atm3/1
Router(config-if)# no ip address
Router(config-if)# no shutdown
Router(config-if)# end
Router#
```

Step 5 Configure the PVC:

```
Router(config)# interface atm3/1.1 point-to-point
Router(config-if)# no ip address
Router(config-subif)# mtu 1500
Router(config-subif)# pvc 1/101
Router(config-if-atm-vc)# bre-connect 10
Router(config-subif)# end
Router#
```

Note If an ATM interface has only BRE VLANs configured, you must enter the **spanning-tree bpdufilter enable** command on the main ATM interface. Entering this command blocks all spanning tree BPDUs on the ATM interface. If RFC 1483 bridged VLANs are also configured on the same ATM interface or on one of its subinterfaces, do not enter the **spanning-tree bpdufilter enable** command unless you want to specifically block BPDUs on that interface.

Configuring PVC Traffic Parameters

The supported traffic parameters are part of the following service categories: Unspecified Bit Rate (UBR) and Variable Bit Rate Non Real-Time (VBR-NRT). Only one of these categories can be specified per PVC connection; if a new one is entered, it replaces the existing one.

To configure PVC traffic parameters, perform one of these tasks beginning in interface-ATM-VC configuration mode:

Command	Purpose
`Router(config-if-atm-vc)# ubr`	Configures UBR. The default is UBR.
`Router(config-if-atm-vc)# vbr-nrt` *pcr scr mbs*	Configures VBR-NRT.

The *-pcr*, *-scr,* and *-mbs* arguments are the peak cell rate, sustainable cell rate, and maximum burst size.

The maximum configurable speed of a VBR-NRT PVC is 299520 Mbps. The maximum MBS is 255 cells.

This example shows how to configure VBR-NRT with a peak cell rate of 1000, a sustainable cell rate of 500, and a maximum burst size of 64:

```
Router(config-if-atm-vc)# vbr-nrt 1000 500 64
Router(config-if-atm-vc)#
```

Configuring SVCs

ATM SVCs are created and released dynamically, providing user bandwidth on demand. This service requires a signaling protocol between the router and the switch.

The ATM signaling software provides a method of dynamically establishing, maintaining, and clearing ATM connections at the UNI. The ATM signaling software conforms to ATM Forum UNI 3.x, 4.0 depending on the version selected by the ILMI or the configuration.

In UNI mode, the Cisco 7600 series router does not perform ATM-level call routing. Instead, the ATM switch performs the ATM call routing, and the router routes packets through the resulting circuit. The router is viewed as the user and the LAN interconnection device at the end of the circuit, and the ATM switch is viewed as the network.

You must complete the tasks in the following sections to use SVCs:

- Configuring Communication with the ILMI, page 8-9 (Required)
- Configuring the PVC that Performs SVC Call Setup, page 8-10 (Required)
- Configuring the NSAP Address, page 8-11 (Required)
- Creating an SVC, page 8-12 (Optional)

Configuring Communication with the ILMI

In an SVC environment, you must configure a PVC for communication with the Integrated Local Management Interface (ILMI) so the router can receive SNMP traps and new network prefixes. The recommended vpi and vci values for the ILMI PVC are 0 and 16, respectively. To configure ILMI communication, perform the following task in interface configuration mode:

Command	Purpose
Router(config-if)# **pvc** [*name*] *vpi*/*vci* [**ilmi** | **qsaal**]	Creates an ILMI PVC on an ATM main interface.

This example shows how to create an ILMI PVC on an ATM main interface:

```
Router(config-if)# pvc cisco 0/16 ilmi
```

Note This ILMI PVC can be set up only on an ATM main interface, not on ATM subinterfaces.

After you configure an ILMI PVC, you can optionally enable the ILMI keepalive function by performing the following task in interface configuration mode:

Command	Purpose
Router(config-if)# **atm ilmi-keepalive** [*seconds*]	Enables ILMI keepalives and sets the interval between keepalives.

ILMI address registration for receipt of SNMP traps and new network prefixes is enabled by default. The ILMI keepalive function is disabled by default; when enabled, the default interval between keepalives is 3 seconds.

This example shows how to set the ILMI keepalive interval to 3 minutes:

```
Router(config-if)# atm ilmi-keepalive 180
```

Configuring the PVC that Performs SVC Call Setup

One dedicated PVC exists between the router and the ATM switch, over which all SVC call-establishment and call-termination requests flow. After the call is established, data transfer occurs over the SVC, from router to router.

Before any SVCs can be set up, a signaling PVC must be configured.

To configure the signaling PVC for all SVC connections, perform this task in interface configuration mode:

Command	Purpose
Router(config-if)# **pvc** [*name*] *vpi*/*vci* [**ilmi** | **qsaal**]	Configures the signaling PVC for an ATM main interface that uses SVCs.

This example shows how to configure the signaling PVC:

```
Router(config-if)# pvc 0/5 qsaal
Router(config-if-atm-vc)#
```

Note This signaling PVC can be set up only on an ATM main interface, not on ATM subinterfaces.

The VPI and VCI values must be configured consistently with the ATM switch. The standard value for VPI is 0 and VCI is 5.

Configuring the NSAP Address

Each ATM interface involved with signaling must be configured with a network service access point (NSAP) address. The NSAP address is the ATM address of the interface and must be unique across the network.

To configure an NSAP address, complete the tasks described in one of the following sections:

- Configuring the ESI and Selector Fields, page 8-11
- Configuring the Complete NSAP Address, page 8-11

Configuring the ESI and Selector Fields

If the switch is capable of delivering the NSAP address prefix to the router by using ILMI, and the router is configured with a PVC for communication with the switch through ILMI, you can configure the end station ID (ESI) and selector fields using the **atm esi-address** command. The **atm esi-address** command allows you to configure the ATM address by entering the ESI (12 hexadecimal characters) and the selector byte (2 hexadecimal characters). The NSAP prefix (26 hexadecimal characters) is provided by the ATM switch.

To configure the router to get the NSAP prefix from the switch and use locally entered values for the remaining fields of the address, perform this task in Interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# pvc [name] vpi/vci [ilmi | qsaal]`	Configures an ILMI PVC on an ATM main interface for communicating with the switch by using ILMI.
Step 2	`Router(config-if-atm-vc)# exit`	Returns to Interface configuration mode.
Step 3	`Router(config-if)# atm esi-address esi.selector`	Enters the ESI and selector fields of the NSAP address.

```
Router(config-if)# pvc 0/16 ilmi
Router(config-if-atm-vc)# exit
Router(config-if)# atm esi-address 3456.7890.1234.12
```

The recommended vpi value for the ILMI PVC is 0 and the recommended vci value is 16.

You can also specify a keepalive interval for the ILMI PVC.

Configuring the Complete NSAP Address

When you configure the ATM NSAP address manually, you must enter the entire address in hexadecimal format because each digit entered represents a hexadecimal digit. To represent the complete NSAP address, you must enter 40 hexadecimal digits in the following format:

```
xx.xxxx.xx.xxxxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx
```

Note All ATM NSAP addresses may be entered in the dotted hexadecimal format shown, which conforms to the ATM Forum UNI specification. The dotted method provides some validation that the address is a legal value. If you know your address format is correct, the dots may be omitted.

Because the interface has no default NSAP address, you must configure the NSAP address for SVCs. To set the ATM interface source NSAP address, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# atm nsap-address nsap-address`	Configures the ATM NSAP address for an interface.

This example shows how to configure the NSAP address:

```
Router(config-if)# atm nsap-address BC.CDEF.01.234567.890A.BCDE.F012.3456.7890.1334.13
Router(config-if)#
```

The **atm nsap-address** and **atm esi-address** commands are mutually exclusive. Configuring the router with the **atm nsap-address** command negates the **atm esi-address** setting and vice versa. You can display the ATM address for the interface by executing the **show interface atm** command.

Creating an SVC

To create an SVC, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# svc [name] nsap address`	(Optional) Creates an SVC and specifies the destination NSAP address.
Step 2	`Router(config-if-atm-vc)# encapsulation {aal5mux | aal5snap}`	(Optional) Configures the ATM adaptation layer (AAL) and encapsulation type. The default is **aal5snap**.
Step 3	`Router(config-if-atm-vc)# protocol protocol protocol-address [[no] broadcast]`	Maps a protocol address to an SVC.

After you specify a name for an SVC, you can reenter interface-ATM-VC configuration mode by entering the **svc** *name* command; you can remove an SVC configuration by entering the **no svc** *name* command.

For a list of AAL types and encapsulations supported for the *aal-encap* argument, refer to the **encapsulation aal5** command in the "ATM Commands" chapter of the *Cisco IOS Wide-Area Networking Command Reference*. The default is AAL5 with SNAP encapsulation.

This example shows how to create an SVC:

```
Router(config-if)# svc nsap BC.CDEF.01.234567.890A.BCDE.F012.3456.7890.1334.13
Router(config-if-atm-vc)# encapsulation aal5mux
Router(config-if-atm-vc)# protocol ip 1.1.1.5 broadcast
Router(config-if-atm-vc)#
```

Configuring Multipoint Bridging

Multipoint bridging enables point-to-multipoint bridging for ATM permanent virtual circuits (PVCs). This feature allows the use of multiple VCs per VLAN for bridging.

Multipoint bridging allows service providers to add support for Ethernet-based Layer 2 services to the proven technology of their existing ATM legacy networks. Customers can then use their current VLAN-based networks over the ATM 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 which provides 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.

In Cisco IOS Release 12.2(18)SXE, multipoint bridging supports the following modes of operation:

- Raw (default)—Default bridging access mode, in which the bridged connection acts on and transmits bridge protocol data unit (BPDU) packets.
- Access—Access-only bridging access mode, in which the bridged connection does not act on or transmit BPDU packets.
- 802.1Q—Performs IEEE 802.1Q tagging to preserve the class of service (CoS) information from the Ethernet frames across the ATM network.
- 802.1Q Tunnel—IEEE 802.1Q tunneling mode, so that service providers can use one VLAN to support customers with multiple VLANs, as well as accept untagged frames.

Restrictions and Usage Guidelines

The following restrictions apply to the Multipoint Bridging feature:

- Supported only on Enhanced OSMs; non-enhanced OSMs are not supported.
- On ATM interfaces, only permanent virtual circuits (PVCs) are supported. Switched virtual circuits (SVCs) are not supported.
- Up to 10,000 bridged PVCs are supported per router.
- VLAN ID 1 is not available as a bridge domain for doing multipoint bridging.
- Multipoint bridging supports an absolute maximum of 60 VCs per each VLAN, and an absolute maximum number of VLANs per peer is 4096. We recommend configuring at most 30 VCs per VLAN, with at most 1024 VLANs per VC.
- Do not use the **range pvc** command with stateful switchover (SSO).

Prerequisites

The following prerequisites apply to multipoint bridging:

- VLANs must be manually added to the VLAN database, using the **vlan** command, to be able to use those VLANs in multipoint bridging.

- Cisco IOS Release 12.2(18)SXE, and later releases, have renamed the **bridge-vlan** command to **bridge-domain**, and have added options to support multipoint bridging. Existing configurations of the **bridge-vlan** command are automatically renamed to **bridge-domain** in the running-config when upgrading to Cisco IOS Release 12.2(18)SXE. Be sure to save the running-config to startup-config to make these changes permanent.

Configuring Multipoint Bridging for ATM Interfaces

This section describes how to configure multipoint bridging on ATM interfaces. You can configure multipoint bridging manually on individual PVCs, or you can configure a range of PVCs to configure all of the PVCs at one time. To perform either task or both, use the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan** {*vlan-id* | *vlan-range*}
4. **interface atm** *slot/port*
5. **no ip address**

Note It is not recommended to configure an IP address on a bridged interface.

6. **interface atm** *slot/port.subinterface* [**point-to-point** | **multipoint**]

Note Use the following two commands (**pvc** and **bridge-domain**) to create and configure PVCs individually. Repeat these commands as desired.

7. **pvc** [*name*] *vpi/vci*
8. **bridge-domain** *vlan-id* [**access** | **dot1q** | **dot1q-tunnel**] [**ignore-bpdu-pid**] [**split-horizon**]

Note Use the following two commands (**range pvc** and **bridge-domain**) to create and configure a range of PVCs. Repeat these commands as desired.

9. **range** [*range-name*] **pvc** [*start-vpi/*]*start-vci* [*end-vpi/*]*end-vci*

Note Do not use the **range pvc** command with stateful switchover (SSO).

10. **bridge-domain** *start-vlan-id* [**access** | **dot1q** | **dot1q-tunnel**] [**ignore-bpdu-pid**] [**increment**] [**split-horizon**]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vlan** {*vlan-id* | *vlan-range*} **Example:** `Router(config)# vlan 2,5,10-12,20,25,4000`	Adds the specified VLAN IDs to the VLAN database and enters VLAN configuration mode. • *vlan-id*—Specifies a single VLAN ID. The valid range is from 2 to 4094. • *vlan-range*—Specifies multiple VLAN IDs, as either a list or a range. The *vlan-range* can contain a list of the VLAN IDs, separated either by a comma (,), dash (-), or both. **Note** You must manually enter a VLAN ID into the VLAN database before you can use that VLAN for multipoint bridging.
Step 4	**interface atm** *slot/port* **Example:** `Router(config)# interface atm 8/0`	Enters configuration mode for the specified ATM interface.
Step 5	**no ip address** **Example:** `Router(config-if)# no ip address`	Removes the IP address on the main interface.
Step 6	**interface atm** *slot/port.subinterface* [**point-to-point** | **multipoint**] **Example:** `Router(config-if)# interface atm 8/0.100`	(Optional) Enters configuration mode for the specified subinterface, assuming that you are using subinterfaces to organize the PVCs. • **point-to-point**—Creates a point-to-point connection. • **multipoint**—Allows multiple PVCs to use the same VLAN.
	Use the following two commands (**pvc** and **bridge-domain**) to create and configure PVCs individually. Repeat these commands as desired.	
Step 7	**pvc** [*name*] *vpi/vci* **Example:** `Router(config-if)#`	Configures a new ATM PVC with the specified VPI and VCI numbers: • *name*—(Optional) Descriptive name to identify this PVC. • *vpi/vci*—Virtual path identifier (vpi) and virtual channel identifier (VCI) for this PVC.

	Command or Action	Purpose
Step 8	**bridge-domain** *vlan-id* [*access* | **dot1q** | **dot1q-tunnel**] [**ignore-bpdu-pid**] [**split-horizon**] **Example:** `Router(config-if-atm-vc)#`	Enables RFC 1483 bridging to map a bridged VLAN to a PVC. The following options are supported: **Note** This command has additional options that are not supported in a multipoint bridging configuration. • *vlan-id*—Number of VLAN to be used in this bridging configuration. The valid range is from 2 to 4094 (but the VLAN ID must have been previously added to the VLAN database in Step 3). **Note** This is the default configuration; frames are not tagged with the dot1q header but STPs and BPDUs are transmitted. • **access**—Enables bridging access mode, so that the bridged connection does not act on or transmit BPDUs. • **dot1q**—(Optional) Terminates dot1q traffic. Also enables IEEE 802.1Q tagging to preserve the class of service (CoS) information from the Ethernet frames across the ATM network. Without this option, the CoS values are not preserved. • **dot1q-tunnel**—(Optional) Enables IEEE 802.1Q tunneling mode, so that service providers can use one VLAN to support customers with multiple VLANs, as well as accept untagged frames. **Note** The **access**, **dot1q**, and **dot1q-tunnel** options are mutually exclusive. If you do not specify any of these options, the connection operates in "raw" bridging access mode, which is similar to **access**, except that the connection does act on and transmit BPDU packets. • **ignore-bpdu-pid**—(Optional, ATM interfaces only) Ignores bridge protocol data unit (BPDU) PID and treats all BPDU packets as data packets to allow interoperation with ATM customer premises equipment (CPE) devices that do not distinguish BPDU packets from data packets. • **split-horizon**—(Optional) Enables RFC 1483 split-horizon mode to globally prevent bridging between PVCs in the same VLAN.

Note Previous software releases used the **atm bridge-enable** command to enable ATM RFC 1483 bridging, but Cisco IOS Release 12.2(18)SXE and later 12.2 SX releases have deprecated this command for OSM interfaces.

Note Use the following two commands (**range pvc** and **bridge-domain**) to create and configure a range of PVCs. Repeat these commands as desired.

Note Do not use the **range pvc** command with stateful switchover (SSO).

	Command or Action	Purpose
Step 9	**range** [*range-name*] **pvc** [*start-vpi/*]*start-vci* [*end-vpi/*]*end-vci* **Example:** `Router(config-if-atm-vc)# range pvc 1/121 1/180`	Creates a range of PVCs, and enters PVC range configuration mode: • *range-name*—(Optional) Descriptive name of the range, up to a maximum of 15 characters. • *start-vpi/*—(Optional) Beginning value for the range of virtual path identifiers (VPIs). The valid range is from 0 to 255, with a default of 0. • *start-vci*—Beginning value for a range of virtual channel identifiers (VCIs). The valid range is from 32 to 65535. • *end-vpi*—End value for the range of VPIs. The valid range is from 0 to 255, with a default that is equal to the start-vpi value. • *end-vci*—End value for a range of virtual channel identifiers (VCIs). The VCI value ranges from 32 to 65535.

Step 10	**bridge-domain** *vlan-id* [**access** | **dot1q** | **dot1q-tunnel**] [**ignore-bpdu-pid**] [**increment**] [**split-horizon**]
	
	Example:
	`Router(config-if-atm-range)# bridge-domain 121 increment`
	`Router(config-if-atm-range)#`
		Enables RFC 1483 bridging to map a bridged VLAN to the configured range of PVCs. In addition to the options that are shown in Step 8, this command supports the following option when used in PVC range configuration mode:
		• **increment**—Increments the bridge domain number for each PVC in the range.

Step 11	**end**
	
	Example:
	`Router(config-if-atm-range)# end`
	`Router#`
		Exits VC or PVC range configuration mode and returns to privileged EXEC mode.

The following example shows both a range of PVCs and an individual PVC being configured for multipoint bridging:

```
interface ATM3/1.101 multipoint
 range pvc 102/100 102/102
  bridge-domain 102 increment
 !
 mls qos trust dscp
```

Verification

To display information about the PVCs that have been configured on ATM interfaces, use the following commands:

- **show atm pvc**—Displays a summary of the PVCs that have been configured.
- **show atm vlan**—Displays the connections between PVCs and VLANs.

Tip Use the **show atm vlan** command instead of the **show interface trunk** command to display information about ATM interfaces being used for multipoint bridging.

The following shows an example of each command:

```
Router# show atm pvc
           VCD /                                          Peak  Avg/Min Burst
Interface  Name          VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells  Sts
3/1.100    3             101   100  PVC    SNAP     UBR  599040                UP
3/1.100    4             111   100  PVC    SNAP     UBR  599040                UP
3/2.100    3             102   100  PVC    SNAP     UBR  599040                UP
3/2.100    4             112   100  PVC    SNAP     UBR  599040                UP

Router# show atm vlan

 Options Legend: DQ - dot1q;  DT - dot1q-tunnel;  MD - multi-dot1q;
                 AC - access; SP - split-horizon; BR - broadcast;
                 IB - ignore-bpdu-pid;
                 DEF - default

Interface              VCD    VPI      Network   Customer   PVC     Options
                              /VCI     Vlan ID   Dot1Q-ID   Status

ATM3/1.100              3   101/100       101        -        UP    DEF
ATM3/1.100              4   111/100       111        -        UP    DEF
ATM3/2.100              3   102/100       102        -        UP    DEF
ATM3/2.100              4   112/100       112        -        UP    DEF
```

RFC 1483 Spanning-Tree Interoperability Enhancements

The RFC 1483 Spanning-Tree Interoperability Enhancements feature allows interoperability between two different BPDU formats (PVST+ and 802.1d) between Cisco 7600 series routers and legacy Catalyst 5500 ATM switches, Cisco 7200 routers, and Cisco 7500 routers.

This section describes an interoperability feature for the various Spanning-Tree implementations across 1483 Bridge-Mode ATM PVCs. Historically, vendors have not implemented Spanning-Tree across 1483 encapsulation consistently; furthermore, some Cisco IOS versions may not support the full range of Spanning-Tree options. This feature attempts to smooth some of the practical challenges of interworking common variations of Spanning-tree over RFC 1483 Bridged-Mode encapsulation.

Note This feature set is only supported on RFC 1483 Bridged-Mode ATM PVCs.

Let's first define the basic terms:

- *IEEE 802.1D* is a standard for interconnecting LANs through MAC bridges (see Figure 8-4 on page 8-21). 802.1D uses the Spanning-Tree Protocol to eliminate loops in the bridge topology, which cause broadcast storms.
- *Spanning-Tree Protocol (STP)* as defined in the IEEE 802.1D is a link management protocol that provides path redundancy while preventing undesirable loops in the network. An IEEE 802.1D spanning tree makes it possible to have one Spanning Tree instance for the whole switch, regardless of the number of VLANs configured on the switch.
- *Bridge Protocol Data Unit (BPDU)* is the generic name for the frame used by the various spanning-tree implementations. The Spanning-Tree Protocol uses the BPDU information to elect the root switch and root port for the switched network, as well as the root port and designated port for each switched segment.

- *Per VLAN Spanning Tree (PVST)* is a Cisco proprietary protocol that allows a Cisco device to support multiple spanning tree topologies on a per-VLAN basis. PVST uses the BPDUs defined in IEEE 802.1D (see Figure 8-4 on page 8-21), but instead of one STP instance per switch, there is one STP instance per VLAN.
- *PVST+* is a Cisco proprietary protocol that creates one STP instance per VLAN (as in PVST). However, PVST+ enhances PVST and uses Cisco proprietary BPDUs with a special 802.2 SNAP OUI (see Figure 8-2 on page 8-20) instead of the standard IEEE 802.1D frame format used by PVST (see Figure 8-4). PVST+ BPDUs are also known as SSTP (Shared Spanning Tree Protocol) BPDUs.

Note RFC 1483 is referenced throughout this section, although it has been superseded by RFC 2684.

Supported Supervisors and Line Cards

The Cisco 7600 router supports PVST to PVST+ BPDU interoperability with the following Supervisors and line cards:

Supervisor 720 Line Cards

- Enhanced FlexWAN
- FlexWAN
- 7600 SIP-200
- ATM Optical Services Module (OSM)

Supervisor 2 Line Cards

- Enhanced FlexWAN
- FlexWAN
- ATM Optical Services Module (OSM)

Prerequisites

The RFC 1483 Spanning-Tree Interoperability Enhancements feature requires Cisco IOS Release 12.2(18)SXF1 or later.

The Interoperability Problem Summarized

The current interoperability problem can be summarized as follows:

- When transmitting STP BPDUs, many vendors' implementations of ATM-to-Ethernet bridging are not fully compliant with the specifications of RFC 1483, Appendix B. The most common variation of the standard is to use an ATM Common Part Convergence Sublayer (CPCS) SNAP PDU with OUI: 00-80-C2 and PID: 00-07. Appendix B reserved this OUI/PID combination for generic Ethernet frames without BPDUs. Appendix B specifies OUI: 00-80-C2, PID: 00-0E for frames with BPDU contents.
- There are several varieties of the Spanning-Tree protocol used by Cisco products on ATM interfaces. The Catalyst 5000 supports only PVST on ATM interfaces. The Cisco 7600 and Catalyst 6500 support only PVST+ on ATM interfaces. Most other Cisco routers implement classic IEEE 802.1D on ATM interfaces.

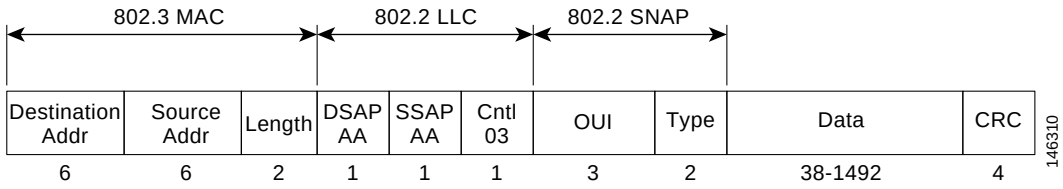
When the Cisco 7600 router and the Catalyst 6500 switch first implemented 1483 Bridging (on Cisco IOS 12.1E) on the Cisco 7600 FlexWAN module, the platform uses OUI: 00-80-C2 and PID: 00-0E to maximize interoperability with all other Cisco IOS products.

However, there are so many implementations that do not send PVST or 802.1D BPDUs with PID: 00-0E that the Cisco 7600 and the Catalyst 6500 reverted to the more common implementation of RFC 1483 (with PID: 00-07) in Cisco IOS 12.2SX. This feature provides the option of encapsulating BPDUs across 1483 with either PID: 00-07 or PID: 00-0E.

BPDU Packet Formats

In this section, the various BPDU packet formats are described. Figure 8-2 shows the generic IEEE 802.2/802.3 frame format, which is used by PVST+—but is not used by PVST.

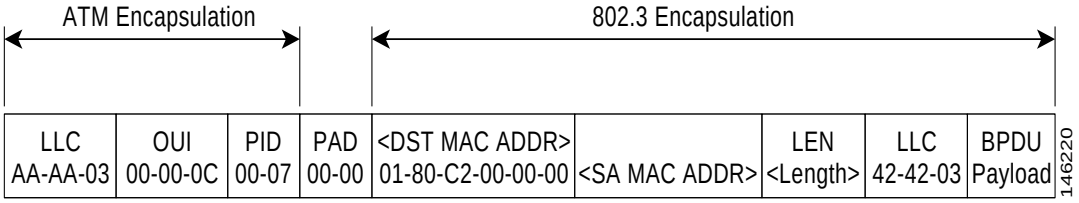
Figure 1-2 IEEE 802.2/802.3 SNAP Encapsulation (RFC 1042)



Catalyst 5000 PVST BPDU Packet Format

The Catalyst 5000 Series switches send and receive BPDUs in PVST format on ATM interfaces (see Figure 8-3):

Figure 1-3 BPDU PVST Frame Used by the Catalyst 5000 Switch



BPDUs sent by the Catalyst 5000 use a PID of 0x00-07, which does not comply with RFC1483. The Cisco 7600 router also has the ability to send BPDUs in this data format.

By using the **bridge-domain** command's **ignore-bpdu-pid** optional keyword, the Catalyst 5000 switch sends this frame by default.

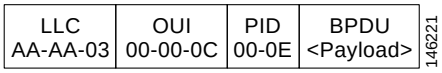
The Catalyst 5000 cannot accept the PVST+ BPDUs and blocks the ATM port, giving the following error message:

```
%SPANTREE-2-RX_1QNON1QTRUNK: Rcved 1Q-BPDU on non-1Q-trun port 6/1 vlan 10
%SPANTREE-2-RX_BLKPORTPVID: Block 6/1 on rcving vlan 10 for inc peer vlan 0
```

Cisco 7200/7500 IEEE BPDU Frame Format

Figure 8-4 shows the Cisco 7200/7500 routers IEEE BPDU frame format:

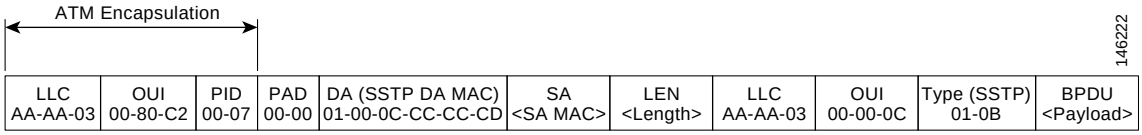
Figure 1-4 Frame Format for the Cisco 7200/7500 IEEE BPDU



Cisco 7600 PVST+ BPDU Frame Format

The Cisco 7600 router PVST+ BPDU packet format is as shown in Figure 8-5. These BPDUs are not IEEE BPDUs, but Cisco proprietary SSTP BPDUs.

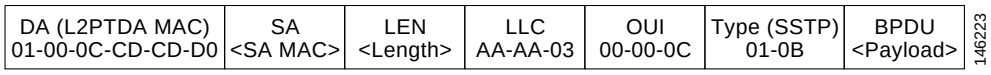
Figure 1-5 Cisco 7600 PVST+ BPDU Frame (1483 Bridged)



Cisco L2PT BPDU Frame Format

Figure 8-6 shows the Cisco Layer 2 Protocol Tunneling (L2PT) BPDU SNAP frame format:

Figure 1-6 L2PT BPDU SNAP Frame Format



BPDU Translation Command Line Interface Summary

In order to resolve the interoperability problem as described in the previous section, Cisco has introduced the following new keywords for the **bridge-domain** command:

- **ignore-bpdu-pid**
- **pvst-tlv**

The ignore-bpdu-pid Keyword

Without the **ignore-bpdu-pid** keyword, the permanent virtual circuit (PVC) between the devices operates in an RFC 1483 compliant manner, which is referred to as *strict mode*. Using the **ignore-bpdu-pid** keyword is known as *loose mode*.

- Without this keyword, IEEE BPDUs are sent out using a PID of 0x00-0E, which complies with RFC 1483.
- With this keyword, IEEE BPDUs are sent out using a PID of 0x00-07, which is normally reserved for 1483 data.

  For details, refer to the "BPDU Packet Formats" section on page 8-20.

Cisco proprietary PVST+ BPDUs are always sent out on data frames using a PID of 0x00-07, regardless of whether the **ignore-bpdu-pid** keyword is used.

Use the <**ignore**> keyword when connecting to devices that send PVST (or 802.1D) BPDUs with PID: 00-07. This includes the vast majority of CPE devices, such as ATM DSL modems.

The pvst-tlv Keyword

The **pvst-tlv** keyword enables BPDU translation when interoperating with devices that understand only PVST or IEEE Spanning Tree Protocol. Since the Cisco 7600 ATM modules support PVST+ only, the **pvst-tlv** keyword must be used when connecting to a Catalyst 5000 switch, which only understands PVST on its ATM modules, or when connecting with other Cisco IOS routers, which understand IEEE format only.

- When transmitting, the **pvst-tlv** keyword translates PVST+ BPDUs into IEEE BPDUs.
- When receiving, the **pvst-tlv** keyword translates IEEE BPDUs into PVST+ BPDUs.

When a Cisco 7600 Router Is Connected to a Cisco 7200 Router

For example, the Cisco 7200 router is a device that only understand IEEE BPDUs in an RFC 1483 compliant manner. Thus, when a Cisco 7600 router is connected to a Cisco 7200 router, the keywords used should be as follows:

```
bridge-domain <vlan> pvst-tlv <vlan>
```

The **ignore-bpdu-pid** keyword is not used in this case because the Cisco 7200 router must operate in an RFC 1483 compliant manner for IEEE BPDUs.

When a Cisco 7600 Router Is Connected to a Catalyst 5500 ATM Module

The Catalyst 5500 ATM module is a device that only understands PVST BPDUs in a non-RFC1483 compliant manner. Therefore, when a Cisco 7600 router is connected to a Catalyst 5500 ATM module, we need to use both keywords:

```
bridge-domain <vlan> ignore-bpdu-pid pvst-tlv <vlan>
```

Layer 2 Protocol Tunneling Topology CLI

To enable BPDU translation for the Layer 2 Protocol Tunneling (L2PT) topologies, use the following command line:

```
bridge-domain <PE vlan> dot1q-tunnel ignore-bpdu-pid pvst-tlv <CE vlan>
```

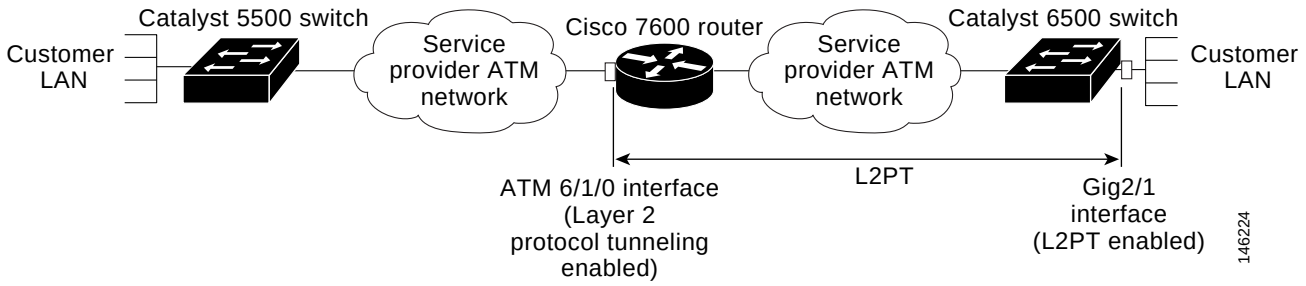
Typical Topologies Requiring BPDU Translation

This section describes the most common network scenarios and provides the configuration commands necessary to enable BPDU translation between the devices in each example.

Layer 2 Protocol Tunneling Topology with a Cisco 7600, Catalyst 5500, and Catalyst 6500

Figure 8-7 shows one sample network topology in which data packets are sent between a Catalyst 5500 switch and a Cisco 7600 router:

Figure 1-7 Catalyst 5500 Switch and Cisco 7600 Routers in an L2PT Topology



As shown in Figure 8-7, Layer 2 Protocol Tunneling (L2PT) is configured at the Cisco 7600 ATM6/1/0 interface and also at the Catalyst 6500 Ethernet2/1 interface.

PVST packets are sent from the Catalyst 5500 switch to the Cisco 7600 router. The Cisco 7600 router transports those BPDUs via L2PT and sends them to the Catalyst 6500. Those BPDUs are decapsulated and restored before sending the packets out to the customer network.

Assume that the 7600 and the Catalyst 6500 are PE devices and the rest are CE devices.

ATM Configuration Example

Any traffic coming in must be sent via a dot1q-tunnel. Assuming the PE-VLAN is 200 and the CE-VLAN is 100, we have the following configuration:

```
Router(config)#int atm 6/1/0
Router(config-if)#pvc 6/200
Router(config-if-atm-vc)#bridge-domain 200 dot1q-tunnel ignore-bpdu-pid pvst-tlv 100
Router(config-if-atm-vc)#
```

Ethernet Configuration Example

The Ethernet configuration is something like this:

```
Router(config)#int gig2/1/0
Router(config-if)#switchport
Router(config-if)#switchport access vlan 200
Router(config-if)#switchport mode dot1q-tunnel
Router(config-if)#l2protocol-tunnel
```

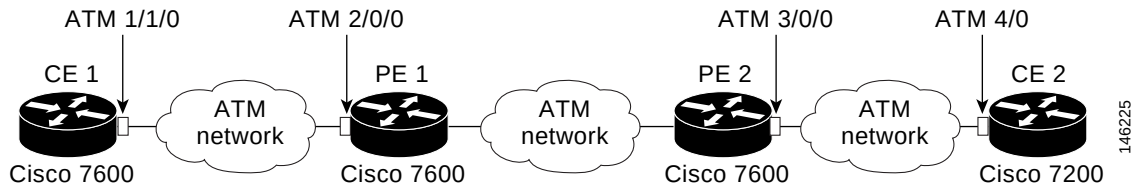
CE-VLAN 100 is what is used at the customer sites. The Catalyst 5500 sends the IEEE BPDU in data format. The Cisco 7600 router receives the BPDU and first converts it to PVST+ format. Then the DA MAC of the frame is changed to the protocol tunnel MAC address and sent out into the Layer 2 cloud.

At the other end, when the frame leaves the gige2/1/0 interface, the DA MAC is changed back to the PVST+ DA MAC and the PVST+ BPDU is sent to the CPE device.

Layer 2 Protocol Tunneling Topology with a Cisco 7600 and Cisco 7200

In the example shown in Figure 8-8, a Cisco 7600 router needs to communicate with a Cisco 7200 router:

Figure 1-8 Cisco 7600 and Cisco 7200 Routers in an L2PT Topology



PE Configuration

On the PEs, the configuration looks something like this:

```
!On PE 1
interface ATM2/0/0
    no ip address
    atm mtu-reject-call
    pvc 7/101
    bridge-domain 200 dot1q-tunnel
    !
end
!On PE 2
interface ATM3/0/0
    no ip address
    pvc 2/101
    bridge-domain 200 dot1q-tunnel pvst-tlv 100
    !
end
```

Cisco 7600 CE Configuration

The configuration for the Cisco 7600 CE 1 would be as follows:

```
!On CE 1
interface ATM1/1/0
    no ip address
    atm mtu-reject-call
    pvc 7/101
    bridge-domain 101
    !
end
```

Cisco 7200 CE Configuration

The configuration for the Cisco 7200 (CE 2) router would be like this:

```
!On CE 2
interface ATM4/0
    no ip address
    no atm ilmi-keepalive
    pvc 2/101
    !
    bridge-group 101
end
```

Data Transmission Sequence from the Cisco 7200 CE to the Cisco 7600 CE

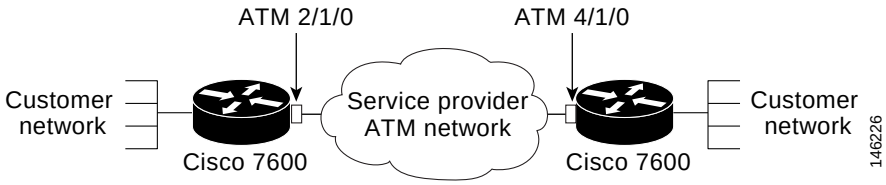
Given the configurations and topologies summarized here, the data transmission sequence from the Cisco 7200 CE to the Cisco 7600 CE is as follows:

1. The Cisco 7200 CE 2 sends BPDUs without the MAC header in RFC1483 format.
2. The Cisco 7600 PE receives it and then translates the IEEE BPDU into PVST+ BPDU format.
3. VLAN 100 is inserted into the PVST+ BPDU.
4. Then the frame's DA MAC is rewritten to use the protocol tunnel destination address (DA) MAC and is sent out into the ATM network cloud.
5. The L2PT BPDU needs to go out of PE 1's ATM2/0/0 interface. The DA MAC is restored to the PVST+ DA MAC.
6. Finally, the PVST+ BPDU is sent to the 7600 CE 1 device.

7600 Basic Back-to-Back Scenario

The basic back-to-back scenario in Figure 8-9 is as follows:

Figure 1-9 Cisco 7600 Routers in Basic Back-to-Back Topology



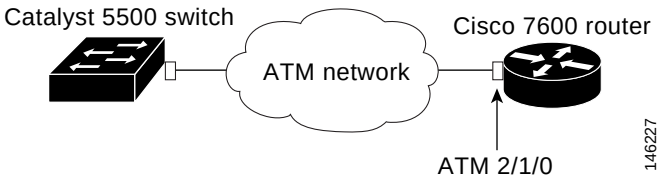
The PDUs exchanged are PVST+ BPDUs. The PVST+ BPDUs are sent using a PID of 0x0007. Here is the configuration:

```
Router(config)#int atm 2/1/0
Router(config-if)#pvc 2/202
Router(config-if-atm-vc)#bridge-domain 101
Router(config-if-atm-vc)#
```

Catalyst 5500 Switch and Cisco 7600 Routers in Back-to-Back Topology

Another sample topology, shown in Figure 8-10, is a simple back-to-back setup, which serves to test basic Catalyst 5500 and Cisco 7600 interoperability.

Figure 1-10 Catalyst 5500 Switch and Cisco 7600 Routers in Back-to-Back Topology



When connected to a device that sends and receives IEEE BPDUs in data format (PID 0x0007) like the Catalyst 5000's ATM module, the configuration must be something like this:

```
Router(config)#int atm 2/1/0
Router(config-if)#pvc 2/202
```

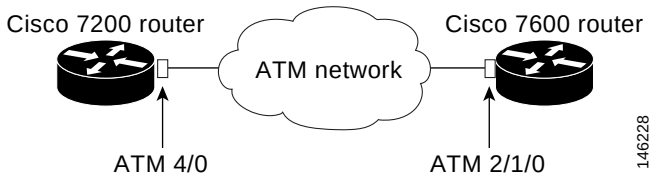
```
Router(config-if-atm-vc)#bridge-domain 101 ignore-bpdu-pid pvst-tlv 101
Router(config-if-atm-vc)#
```

The Cisco 7600 router translates its outgoing PVST+ BPDUs into IEEE BPDUs. Because the **ignore-bpdu-pid** keyword is also enabled, it uses a PID of 0x0007, which is exactly what the Catalyst 5500 switch expects.

Cisco 7600 and Cisco 7200 in Back-to-Back Topology

When connecting to a device that is completely RFC1483 compliant, in which the IEEE BPDUs are sent using a PID of 0x000E, you must use the new **ignore-bpdu-pid** keyword in the **bridge-domain** command.

Figure 1-11 Cisco 7600 Router and Cisco 7200 Router in Back-to-Back Topology



For example, when a Cisco 7600 is connected to a Cisco 7200 router, you would have this configuration:

```
Router(config)#int atm 2/1/0
Router(config-if)#pvc 2/202
Router(config-if-atm-vc)#bridge-domain 101 pvst-tlv 101
Router(config-if-atm-vc)#
```

Note In this case, the CE-VLAN must be the same as the bridge-domain VLAN.

Configuring Automatic Protection Switching

The Automatic Protection Switching (APS) feature supports Linear 1+1 APS as described in section 5.3 of the Telcordia publication"*GR-253-CORE SONET Transport Systems: Common Generic Criteria*."

Linear APS is defined to provide protection at the line layer. All of the STS synchronous payload envelopes (SPEs) carried in an OC-N signal are protected so that if a protection switch occurs, all of the VCs are switched simultaneously.

One port on an OC-12 ATM OSM can be protected by:

- Another port on the same OC-12 ATM OSM
- A different port on another OC-12 ATM OSM in the same router
- A different port on another OC-12 ATM OSM in a different router

Note Installing APS in different routers to protect against router failure requires that you update the protect interface with the current VC context of the working interface. To accomplish this, you must configure the VC configuration on the protect interface manually.

When configuring APS, we recommend that you configure the working interface first, along with the IP address of the interface being used as the APS OOB communication path.

Note To prevent the protected interface from becoming the active circuit and disabling the working circuit when it is discovered, configure the working interface before configuring the protected interface.

For more information on APS and configuration information for additional APS features, refer to the *Cisco IOS Interface Configuration Guide*, Release 12.1 at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/inter_c/index.htm

Configuring the Working Interface

To configure the working interface, perform this task:

Note A typical ATM interface configuration contains several subinterfaces with each having a different IP address and VCs. When configuring a protect interface, ensure that it contains same VCs as the working interface. Because the IP address cannot be the same on two interfaces, the IP addresses you configure on the protect interface should be in same subnet.

	Command	Purpose
Step 1	`Router(config)# interface atm slot/port`	Specifies an ATM interface and enters interface configuration mode.
Step 2	`Router(config-controller)# aps working circuit-number`	Configures this interface as a working interface.
Step 3	`Router(config)# end`	Exits configuration mode.

Step 4	`Router# show controllers atm`
	`Router# show interface atm`
	`Router# show aps`
	`Router# show aps controller`
		Displays information about the ATM controllers and interface so that you can verify that the interface is configured correctly.

Note If a router has two or more protect interfaces, the **aps group** command for each interface must precede the corresponding **aps protect** command.

Configuring the Protect Interface

To configure the protect interface, perform this task beginning in global configuration mode:

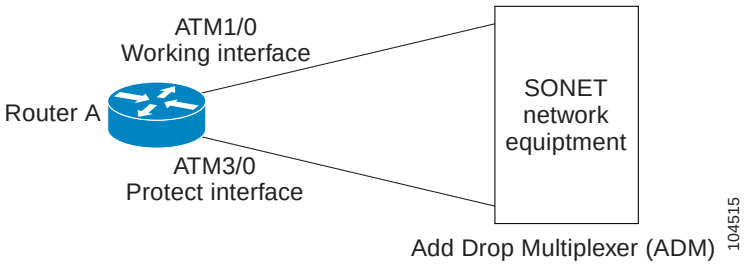
	Command	Purpose
Step 1	`Router(config)# interface atm slot/port`	Specifies an ATM interface and enters interface configuration mode.
Step 2	`Router(config-if)# aps protect circuit-number ip-address`	Configures this interface as a protect interface. Specifies the IP address of the router that contains the working interface.
Step 3	`Router(config-if)# end`	Exits configuration mode.

Step 4	`Router# show controllers atm`
	`Router# show interface atm`
	`Router# show aps`
		Displays information about the ATM controllers and interface so that you can verify that the interface is configured correctly.

Configuring Basic APS on a Single Router

Figure 8-12 shows the configuration of APS on router A and router B. Router A has both the working and protect interfaces. If the working interface ATM1/0 becomes unavailable, the connection automatically switches over to the protect interface ATM3/0. Single router APS configuration is typically used to protect line card failures.

Figure 1-12 Basic Single Router APS Configuration



Step 1 Configure a loopback interface on Router A:

```
RouterA# configure terminal
RouterA(config)# interface Loopback 0/0
RouterA(config-if)# ip address 7.7.7.7 255.255.255.255
RouterA(config-if)# end
RouterA#
```

Step 2 Configure the working and protect interfaces on router A:

```
RouterA# configure terminal
RouterA(config)# interface ATM 1/0
RouterA(config-if)# aps working 1
RouterA(config-if)# exit
RouterA(config)# interface ATM 3/0
RouterA(config-if)# aps protect 1 7.7.7.7
RouterA(config-if)# end
```

Step 3 Configure VCs on both working and protect interfaces:

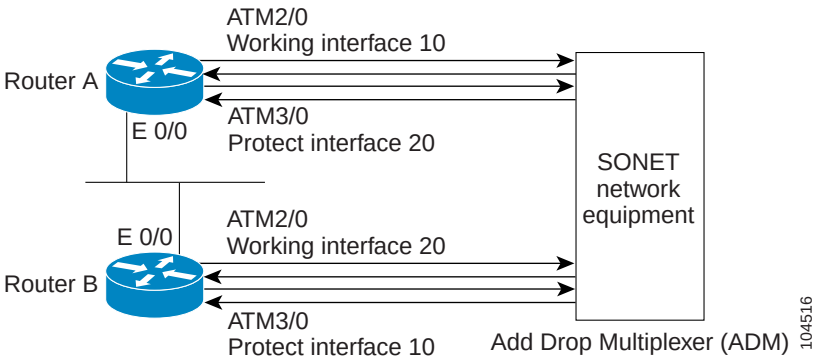
```
RouterA# configure terminal
RouterA(config)# int atm 1/0.1 point-to-point
RouterA(config-subif)# ip address 10.1.1.1 255.255.255.0
RouterA(config-subif)# pvc 0/100
RouterA(config-subif)# exit
RouterA(config)# int atm 3/0.1 point-to-point
RouterA(config-subif)# ip address 10.1.1.1 255.255.255.0
RouterA(config-subif)# pvc 0/100
RouterA(config-subif)# exit
RouterA(config)# end
RouterA#
```

Note Because APS protocol provides redundancy on the line (not end to end), you must configure a duplicate IP address on the protect interface. There is no automatic configuration of the protect interface. You must manually configure all of the VCs that require redundancy on the protect interface.

Basic Multiple Router APS Configuration

Figure 8-13 shows the configuration of APS on router A and router B. Router A is configured with the working interface and router B is configured with the protect interface. If the working interface on router A becomes unavailable, the connection automatically switches over to the protect interface on router B. This is typically used to protect against both line card and router failures.

Figure 1-13 Basic Multiple Router APS Configuration



Step 1 Configure the working interface on RouterA:

```
RouterA# configure terminal
RouterA(config)# interface ethernet 0/0
RouterA(config-if)# ip address 7.7.7.7 255.255.255.0
RouterA(config-if)# exit
RouterA(config)# interface ATM 1/0
RouterA(config-if)# aps working 1
RouterA(config-if)# end
RouterA#
```

Step 2 Configure the protect interfaces on router B:

```
RouterB# configure terminal
RouterB(config)# interface ethernet 0/0
RouterB(config-if)# ip address 7.7.7.6 255.255.255.0
RouterB(config)# interface ATM 3/0
RouterB(config-if)# aps protect 1 7.7.7.7
RouterB(config-if)# end
RouterB#
```

Step 3 Configure the VCs on router A:

```
RouterA# configure terminal
RouterA(config)# int atm 1/0.1 point-to-point
RouterA(config-subif)# ip address 10.1.1.1 255.255.255.0
RouterA(config-subif)# pvc 0/100
RouterA(config-subif)# exit
RouterA(config)# end
RouterA#
```

Step 4 Configure the same VCs on router B:

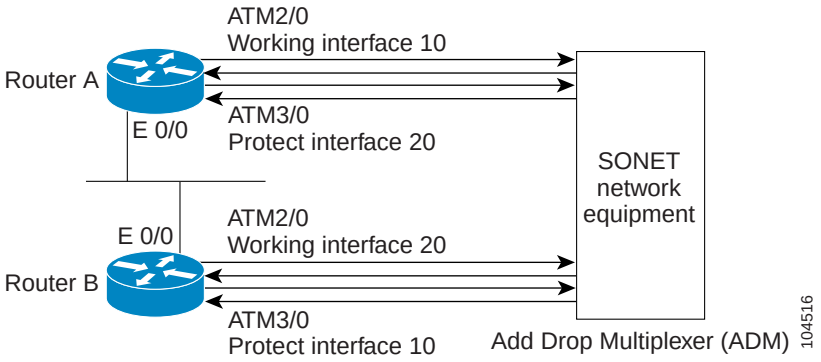
```
RouterB# configure terminal
RouterB(config)# int atm 3/0.1 point-to-point
RouterB(config-subif)# ip address 10.1.1.1 255.255.255.0
RouterB(config-subif)# pvc 0/100
```

```
RouterB(config-subif)# exit
RouterB(config)# end
RouterB#
```

Multiple APS Interface Configuration

To configure more than one protect/working interface on a router, use the **aps group** command. Figure 8-14 shows the configuration for grouping more than one working/protect interface on a router. Both router A and B are configured with a working interface and a protect interface. If the working interface 2/0 on router A becomes unavailable, the connection switches over to the protect interface 3/0 on router B because they are both in APS group 10. Similarly, if the working interface 2/0 on router B becomes unavailable, the connection switches over to the protect interface 3/0 on router A because they are both in APS group 20.

Figure 1-14 Multiple APS Interface Configuration



Note Before you configure the protected interface, configure the working interface to avoid the protected interface from becoming the active circuit and disabling the working circuit when it is discovered.

Step 1 On router A, which contains the working interface for group 10 and the protect interface for group 20, use the following configuration:

```
RouterA# configure terminal
RouterA(config)# interface ethernet 0/0
RouterA(config-if)# ip address 7.7.7.6 255.255.255.0
RouterA(config)# interface ATM2/0
RouterA(config)# aps group 10
RouterA(config-if)# aps working 1
RouterA(config)# interface ATM3/0
RouterA(config-if)# aps group 20
RouterA(config-if)# aps protect 1 7.7.7.7
RouterA(config-if)# end
RouterA#
```

Step 2 On router B, which contains the protect interface for group 10 and the working interface for group 20, use the following configuration:

```
RouterB# configure terminal
RouterB(config)# interface ethernet 0/0
RouterB(config-if)# ip address 7.7.7.7 255.255.255.0
RouterB(config)# interface ATM2/0
RouterB(config)# aps group 20
RouterB(config-if)# aps working 1
```

```
RouterB(config)# interface ATM3/0
RouterB(config-if)# aps group 10
RouterB(config-if)# aps protect 1 7.7.7.6
RouterB(config-if)# end
RouterB#
```

APS Commands

The commands below are applicable to APS with ATM. For information on using these commands, see http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/posaps.htm#xtocid13.

Note Be sure you change the interface from POS to ATM when using these commands with ATM.

- aps authenticate
- aps force
- aps group
- aps lockout
- aps manual
- aps protect
- aps revert
- aps timers
- aps unidirectional
- aps working
- show aps

SONET and SDH Configuration Commands

The default framing on the 2-port OC-12 ATM OSMs is SONET, but the modules also support SDH. Use the following commands to change the mode of operation, specify the BER threshold values, and enable alarm reporting.

atm framing sonet | sdh

Use the **atm framing sonet** | **sdh** command to specify the framing. The default framing is SONET.

```
Router(config-if)# atm framing sdh
Router(config-if)#
```

atm sonet stm-4

Use the **atm sonet stm-4** interface configuration command to set the mode of operation and control the type of ATM cell used for cell-rate decoupling on the SONET physical layer interface module (PLIM). The **no** form of this command restores the default Synchronous Transport Signal level 12 (STS-12c) operation.

Note The **atm sonet stm-4** and **atm framing sdh** commands both change the framing to SDH. The commands are functionally identical.

[no] atm sonet stm-4

This example shows how to change the mode from STS-12 to STM-4 and verify the configuration:

```
Router(config-if)# atm sonet stm-4
Router(config-if)# end
Router# show controllers atm 3/1
Interface ATM3/1 is up

hwidb addr:42773F94, instance addr:42780F64

Framing mode:SDH (STM-4)
Clock source:Line

VPIs in use:3, max VPIs:15

  VPI  # VCs      VPI  # VCs      VPI  # VCs
  ---  -----      ---  -----      ---  -----
    0    201        1      3      255      1

ATM framing errors:
  HCS (correctable):  1058413
  HCS (uncorrectable):3851467
  LCD:                18

SONET Subblock:
SECTION
  LOF = 0          LOS =    2          RDOOL = 0          BIP(B1) = 1020363
  Active Alarms:None
  Active Defects:None
  Alarm reporting enabled for:LOF LOS B1-TCA
LINE
  AIS = 0          RDI =    5          FEBE =  437717490  BIP(B2) = 457516655
  Active Alarms:None
  Active Defects:None
  Alarm reporting enabled for:B2-TCA SF
PATH
  AIS = 0          RDI =    13         FEBE =  345027     BIP(B3) = 1229383
  LOP = 2          NEWPTR = 0          PSE =   0          NSE =     0
  Active Alarms:None
  Active Defects:None
  Alarm reporting enabled for:LOP B3-TCA

BER thresholds: SF = 10e-3, SD = 10e-6
TCA thresholds: B1 = 10e-6, B2 = 10e-6, B3 = 10e-6
Router#
```

atm sonet report

Use the **atm sonet report** interface configuration command to set the ATM SONET alarm reporting. The **no** form of this command removes the alarm reporting.

[**no**] **atm sonet report** {**all** | **b1-tca** | **b2-tca** | **b3-tca** | **default** | **lais** | **lrdi** | **pais** | **plop** | **pplm** | **prdi** | **ptim** | **puneq** | **sd-ber** | **sf-ber** | **slof** | **slos**}

This example shows how to enable alerts for B1 threshold crossings:

```
Router(config-if)# atm sonet report b1-tca
Router(config-if)#
```

atm sonet threshold

Use the **atm sonet threshold** interface configuration command to set the BER threshold values. Use the **no** form of the command to remove the configuration:

[**no**] **atm sonet-threshold** {**b1-tca** *value* | **b2-tca** *value*| **b3-tca** *value*| **sd-ber** *value*| **sf-ber** *value*}

This example shows how to set the B1 threshold:

```
Router(config-if)# atm sonet threshold b1-tca 9
Router(config-if)#
```

show controllers atm

Use the **show controllers atm** command to display information about physical port hardware information.

show controllers atm [***slot****/port-adapter/port*]

This example shows how to show the output for an OC-12ATM linecard.

```
Router# show controllers atm 6/1
```

The output of this command is as follows:

```
~~~~~~~~~
btaps2#sh cont atm 6/1
Interface ATM6/1 is up
hwidb: 0x4316B388, instance: 0x4316EC30, 5 i/f transitions
Framing mode: SONET (STS-12c) Clock source: Internal -- Reason: Configured
VPIs in use: 0, max VPIs: 15
ATM framing errors:
  HCS (correctable):   8
  HCS (uncorrectable): 4375
  LCD:                 0

SONET Subblock:
APS                          <============APS info displayed...
  COAPS = 0          PSBF = 0
  State: PSBF_state = false
  Rx(K1/K2): 0 /0   Tx(K1/K2): 0 /5
SECTION
  LOF = 0          LOS = 0                              BIP(B1) = 65
LINE
  AIS = 0          RDI =    0          FEBE =  712        BIP(B2) = 0
PATH
  AIS = 0          RDI =    1          FEBE =  65535      BIP(B3) = 134
```

```
  LOP = 0          NEWPTR = 0          PSE = 0           NSE = 0

Active Defects: None
Active Alarms:  None
Alarm reporting enabled for: SF SLOS SLOF B1-TCA B2-TCA PLOP B3-TCA

BER thresholds:  SF = 10e-3, SD = 10e-6
TCA thresholds:  B1 = 10e-6, B2 = 10e-6, B3 = 10e-6

Rx S1S0 = 00, Rx C2 = 13

PATH TRACE BUFFER : STABLE
  Remote hostname : btaps1
  Remote interface: ATM6/1
  Remote IP addr  : 0.0.0.0
  Remote Rx(K1/K2): 00/00  Tx(K1/K2): 00/00
~~~~~~~~
```

CHAPTER 9

Configuring QoS on the Optical Services Modules

This chapter describes how to configure Quality of Service (QoS) on the Optical Services Modules (OSMs).

This chapter consists of these sections:

Understanding QoS on the OSMs

QoS for the OSMs is distributed between the OSMs and the Policy Feature Card. This chapter discusses the Layer 3 QoS implementations on the OSM WAN ports.

For the OSM WAN ports, the OSMs support the following Layer 3 QoS implementations:

- Class-based traffic shaping
- Class-based weighted fair queuing (CBWFQ)
- Low latency queuing (LLQ)
- Weighted Random Early Detection (WRED)
- Hierarchical Traffic Shaping

You configure OSM WAN port QoS using a subset of the Modular QoS CLI (MQC). For information on MQC, refer to the *Modular Quality of Service Command-Line Interface Overview* at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xe/120xe5/mqc/mcli.htm.

Note Though the OSM-2OC12-POS-SI+ card contains 2 POS ports and 4 GigE ports, the GigE ports do not support mqc queuing or shaping.

Note Using the **fair-queue** and **random-detect** commands on the main interface is not supported with the OSMs.

For OSM WAN port Layer 3 QoS configuration information and examples, see the "Configuring QoS on the OSMs" section on page 9-2 and the Configuring MPLS QoS, page 1-13.

Additional QoS Features and Resources

- For information about configuring AToM QoS on the OSM WAN ports, see the "How to Configure QoS with AToM" section on page 1-79.
- For information about configuring the Catalyst Layer 2 QoS implementation for 1p1q4t ingress queues and 1p2q2t egress queues on the OSM Gigabit Ethernet LAN ports, refer to the links below for Policy Feature Card QoS on the Cisco 7600 series router and the Catalyst 6000 series switch.
- For information about configuring the Policy Feature Card for Layer 2 and Layer 3 policing and marking of traffic for the OSM WAN ports and the OSM Gigabit Ethernet LAN ports, refer to the links below for Policy Feature Card QoS on the Cisco 7600 series router and the Catalyst 6000 series switch.

Note The Policy Feature Cards 3BXL and PFC3B add support for MPLS policing and marking.

- For information about configuring Policy Feature Card QoS on the Cisco 7600 series router, see:
  - *Cisco 7600 Series Cisco IOS Software Configuration Guide, 12.2SX* http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/swcg/index.htm
  - *Cisco 7600 Series Cisco IOS Command Reference, 12.2SX* http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/cmdref/index.htm
- For information about configuring Policy Feature Card QoS on the Catalyst 6000 series switch running Cisco IOS software on the supervisor engine and on the MSFC, see:
  - *Catalyst 6500 Series Cisco IOS Software Configuration Guide, 12.2SX* http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/index.htm
  - *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* at this URL: http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm
- For general information on how to configure Cisco IOS QoS, see:
  - *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2* http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/index.htm.
  - *Cisco IOS Quality of Service Solutions Command Reference, Release 12.2* http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_r/index.htm.

Configuring QoS on the OSMs

These sections describe configuring QoS on the OSMs:

For information on shaping features supported with Destination Sensitive Services (DSS), see Chapter 1, "Configuring Destination Sensitive Services on the Optical Services Modules."

Enabling QoS Globally

Before you can configure QoS on the OSMs, you must enable the QoS functionality globally. To enable QoS globally, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos`	Enables QoS on the switch. Use the **no mls qos** command to globally disable QoS.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos`	Verifies the configuration.

This example shows how to enable QoS globally:

```
Router(config)# mls qos
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos
  QoS is enabled globally
  Microflow QoS is enabled globally

QoS global counters:
    Total packets: 544393
    IP shortcut packets: 1410
    Packets dropped by policing: 0
    IP packets with TOS changed by policing: 467
    IP packets with COS changed by policing: 59998
    Non-IP packets with COS changed by policing: 0

Router#
```

Configuring Classification

This section contains information for configuring classification for the OSM QoS features.

Classification is the selection of traffic for QoS. OSMs classify IP traffic based on the packet IP precedence or IP DSCP value. OSMs classify MPLS traffic based on the MPLS EXP value in the topmost label in the MPLS label stack. Use the **class-map** command in global configuration mode to specify the name of the class map and define the class-matching criteria.

Restrictions and Usage Guidelines

The classification restrictions and usage guidelines are as follows:

- **Traffic types** —The classification information in this section is for IP and MPLS traffic. For information about configuring classification for EoMPLS and AToM traffic, refer to the "How to Configure QoS with AToM" section on page 1-79.
- **Traffic classes** —You can configure up to 64 discrete traffic classes in a single policy map, one class for each IP DSCP value. In addition to the traffic classes you specify, the class-default class is predefined when you create the policy map. It is the class to which traffic is directed if that traffic does not satisfy the match criteria of other classes defined in the policy map.
- **Using the match command** —Only the **match ip precedence**, **match ip dscp**, and **match mpls experimental** commands are supported. Access control lists and other criteria are not supported as match criteria for traffic classes.

Note In 122-18.SXF7 and subsequent SXF releases, the **show policy-map interface** command does not display packet counters for policy-map classes containing no Actions.

Configuration Tasks

To configure classification, perform this task in global configuration mode on the Mutilayer Switch Feature Card (MSFC):

	Command	Purpose
Step 1	`Router(config)# class-map [match-all | match-any] class-name`	Creates a class map to be used for matching packets to a class.
Step 2	`Router(config-cmap)# match [ip dscp ip-dscp-value | ip precedence ip-precedence-value | mpls experimental mpls-exp-value]`	Configures a specific IP precedence, IP DSCP, or MPLS EXP value as a match criterion.

Configuring Class-Based Traffic Shaping

This section contains information for configuring class-based traffic shaping.

QoS on the OSMs supports both inbound and outbound class-based traffic shaping. Class-based traffic shaping on the OSMs limits the traffic to the configured rate. To configure class-based traffic shaping, use the **shape average** command.

Restrictions and Usage Guidelines

The class-based traffic shaping restrictions and usage guidelines are as follows:

- **Traffic shaping granularity**—On the OSMs, the granularity of the shaping rate is 1/255 of the link rate or the hierarchical shape rate. The OSMs automatically round the configured rate to the nearest multiple of 1/255. The **show policy-map interface** command displays the rounded shaping rate.

Note For the Enhanced GE WAN, Enhanced OC3 POS, and Enhanced OC12 POS, the granularity of the shaping rate is 64,000 bps.

- **Minimum traffic shaping rate**—As shown in Table 9-1, the **shape average** command has a minimum rate.
  - For OC-3c and above interfaces, the minimum rate is 1/255 of the physical interface speed.
  - For T3 and lower interfaces, the minimum rate is 256,000 bps.
  - For hierarchical shaped interfaces, the minimum rate of the parent policy is 1,000,000 bps and the minimum rate of the child policy is the greater of (a) 256,000 bps or (b) 1/255 of the parent shape rate.
- **Traffic shaping bursts**—The OSMs have a fixed burst size; therefore, the OSMs do not the support **shape average** command committed burst (Bc) and excess burst (Be) parameters. When you monitor a shaping policy with the **show policy interface** command, the output shows values for the Bc and Be parameters that the **show** command generates. The OSMs do not use these values.

Table 9-1 Minimum QoS Rates for shape average Command

Physical Interface	Minimum rate for shape average command (bps)
T3 and below	256,000
OC3 POS	607,000
Enhanced OC3 POS	256,000
OC12 POS	2,439,000
Enhanced OC12 POS	256,000
OC48 POS	9,756,000
GE WAN	1,000,000
Enhanced GE WAN	256,000
hierarchical traffic shaping (parent)	1,000,000
hierarchical traffic shaping (child)	Greater of (a) 256,000 bps or (b) 1/255 of the parent rate

Configuration Tasks

To configure class-based traffic shaping, use the Modular QoS CLI. Define the class of traffic with the **class-map** command as shown previously, create a policy map as shown previously that contains the **shape average** command, and apply the policy to the appropriate interface with the **service-policy** command.

To configure the class of traffic, see the "Configuring Classification" section on page 9-3. To configure a policy with class-based traffic shaping and to assign the policy to an interface, perform the following tasks in global configuration mode.

Note The following tasks assume that a traffic class has already been created.

	Command	Purpose
Step 1	`Router(config)# policy-map policy_name`	Specifies the name of the policy map to configure.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a predefined class included in the service policy.
Step 3	`Router(config-pmap-c)# shape average cir`[1] [2]	Shapes traffic to the indicated bit rate for the specified class.
Step 4	`Router(config)# interface interface-name`	Specifies the interface to which the policy map will be applied.
Step 5	`Router(config-if)# service-policy [input | output policy-name]`	Attaches the specified policy map to the interface.

1. Only supported parameters are shown.
2. See Table 9-1 on page 9-5.

Configuration Example

This example shows how to configure a low shape-rate queue and a high shape-rate queue:

Step 1 Create traffic classes containing the match criteria for the two classes by defining the class maps as follows:

```
Router(config)# class-map match-all gold-data
Router(config-cmap)# match ip dscp 40
Router(config-cmap)# exit
Router(config)# class-map match-all bronze-data
Router(config-cmap)# match ip dscp 8
Router(config-cmap)# exit
```

Step 2 Define a service policy to contain policy specifications for the two classes—gold-data and bronze-data. The match criteria for these classes are defined in Step 1.

```
Router(config)# policy-map policy1
Router(config-pmap)# class gold-data
Router(config-pmap-c)# shape average 2000000
Router(config-pmap-c)# exit
Router(config-pmap)# class bronze-data
Router(config-pmap-c)# shape average 5000000
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 3000000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)#
```

Step 3 Apply the policy map to the appropriate interface. The following example attaches the policy as an output policy:

```
Router(config)# interface pos7/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

Step 4 Display the policy information for the interface:

Note The OSMs do not use the Bc and Be values in the output below. The CLI generates these values.

```
Router# show policy interface pos7/1
 POS7/1

  Service-policy output: policy1

    Class-map: gold-data (match-all)
      795533 packets, 271276753 bytes
      30 second offered rate 17269000 bps, drop rate 2939000 bps
      Match: ip dscp cs5
      queue size 0, queue limit 128
      packets output 660256, packet drops 135277
      tail/random drops 135277, no buffer drops 0, other drops 0
      shape (average) cir 20000000 bc 80000 be 80000
      target shape rate 20000000
```

(shape parameter is rounded to 19,513,000 bps due to granularity)

```
    Class-map: bronze-data (match-all)
      795533 packets, 271276753 bytes
      30 second offered rate 17269000 bps, drop rate 13687000 bps
      Match: ip dscp cs1
      queue size 0, queue limit 128
      packets output 165164, packet drops 630369
      tail/random drops 630369, no buffer drops 0, other drops 0
      shape (average) cir 5000000 bc 20000 be 20000
      target shape rate 5000000
```

(shape parameter is rounded to 4,878,000 bps due to granularity)

```
    Class-map: class-default (match-any)
      3182121 packets, 1085103261 bytes
      30 second offered rate 69075000 bps, drop rate 47581000 bps
      Match: any
      queue size 0, queue limit 128
      packets output 990320, packet drops 2191801
      tail/random drops 2191801, no buffer drops 0, other drops 0
      shape (average) cir 30000000 bc 120000 be 120000
      target shape rate 30000000
```

(shape parameter is rounded to 29,270,000 bps due to granularity)

Note For information on hierarchical traffic shaping policies, see the "Configuring Hierarchical Traffic Shaping" section on page 9-17.

Configuring Class-Based Weighted Fair Queuing

This section contains information for configuring Class-Based Weighted Fair Queueing (CBWFQ). CBWFQ provides guaranteed bandwidth rate to a non-priority class. Under congestion conditions, the class receives the guaranteed bandwidth. To configure CBWFQ, use the **bandwidth** command.

Note Low Latency Queueing (LLQ) provides guaranteed bandwidth for the priority classes. The sum of all bandwidth on a link guaranteed by CBWFQ for non-priority classes and LLQ for priority classes cannot exceed 99% of the total available link bandwidth. For more information on LLQ, see the "Configuring Low Latency Queuing" section on page 9-12.

Restrictions and Usage Guidelines

The CBWFQ restrictions and usage guidelines are as follows:

- **OSM support**— CBWFQ is not supported in DPT mode. CBWFQ is supported in POS mode on all OSMs except the OSM-2OC48/1DPT and OSM-4GE-WAN-GBIC.
- **Bandwidth granularit**y—On the OSMs, the granularity of the CBWFQ rate is 1/255 of the link rate or the hierarchical shape rate. The OSMs automatically round the configured rate to the nearest multiple of 1/255. The **show policy-map interface** command displays the rounded CBWFQ rate.
- **Minimum bandwidth rate**—On the OSMs, the minimum CBWFQ rate is the greater of (a) 256 Kbps or (b) 1% of the link rate or the hierarchical shape rate.
- **Bandwidth allocation**—When a link is not experiencing congested conditions, the unused (or excess) bandwidth is shared among all classes. The excess bandwidth available to a class is in proportion to its guaranteed bandwidth specified by the **priority** or **bandwidth** commands. For example, if one class is guaranteed 20% of the link and a second class is guaranteed 10% of the link, then the first class receives twice as much excess bandwidth as the second class.
- **Using class-default**—The guaranteed bandwidth of the class-default class is by default equal to the link bandwidth minus the guaranteed bandwidth allocated to the user-defined traffic classes (with the **bandwidth** and **priority** commands). At least 1% of the link bandwidth is always reserved for the default traffic class. You can alter the bandwidth allocated to the default traffic by using the **bandwidth** command with the class-default class.
- Because the MSFC and PFC do not support CBWFQ, configuring CBWFQ on a system configured with a Supervisor Engine 720 and an OSM-1CHOC12/T1-SI, might cause the output of the show policy-map interface command to display a packet counter of 0 for a serial interface.
- When you configure a Class Based Weighted Fair Queueing or Low Latency Queueing in combination with any Feature requiring MSFC/PFC on a OSM-1CHOC12/T1-SI, the counters for **show policy-map interface** does not increment. This is because some of the features require MSFC/PFC processing, and MSFC/PFC does not support WAN Class Based Weighted Fair Queueing or Low Latency Queueing and the packets are not accounted for QoS. Examples of such Features are:
  - Frame-relay ip tcp header-compression
  - Frame-relay ip rtp header-compression
  - Access-list 101 permit ip any any log

Note MSFC/PFC processes the **log** keyword in the access-lists results of a packet.

Configuration Tasks

These sections describe the configuration tasks for CBWFQ:

- Configuring a Service Policy in the Policy Map, page 9-9

Configuring a Service Policy in the Policy Map

To configure CBWFQ, use the Modular QoS CLI. Define the class of traffic with the **class-map** command, create a policy map that contains the **bandwidth** command, and apply the policy to the appropriate interface with the **service-policy** command.

To configure the class of traffic, see the "Configuring Classification" section on page 9-3. To configure a policy with CBWFQ and assign the policy to an interface, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to configure.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a predefined class included in the service policy.
Step 3	`Router(config-pmap-c)# bandwidth bandwidth-kbps | percent % of available bandwidth`[1] [2]	Specifies the percentage of available bandwidth in kilobits per second to be assigned to packets that meet the match criteria of the associated traffic class.
Step 4	`Router(config)# interface interface-name`	Specifies the interface to which the policy map will be applied.
Step 5	`Router(config-if)# service-policy [output policy-name]`	Attaches the specified policy map to the interface.

1. Only the parameters shown are supported.
2. The **bandwidth** command has a minimum rate that is one percent of the physical interface speed or the hierarchical shaped rate. See Table 9-1.

Displaying the CBWFQ Configuration and Statistics

Use the commands below to display the configuration of a service policy and its associated traffic classes:

Command	Purpose
`Router# show policy-map`	Displays all configured service policies.
`Router# show policy-map policy-map-name`	Displays the user-specified service policy.
`Router# show policy-map interface`	Displays statistics and configurations of all input and output policies, which are attached to an interface.
`Router# show policy-map interface interface-spec`	Displays configuration and statistics of the input and output policies attached to a particular interface.
`Router# show policy-map interface interface-spec output`	Displays configuration statistics of the output policy attached to an interface.
`Router# show policy-map [interface [interface-spec [output] [ class class-name]`	Displays the configuration and statistics for the class name configured in the policy.

This example shows the information displayed when you enter the **show policy-map interface** command:

```
Router1-PE# show policy-map interface

 POS6/2
  service-policy output:s

    queue stats for all priority classes:
      queue size 0, queue limit 32655
      packets output 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0

    class-map:dscp0 (match-all)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      match:ip dscp 0
      queue size 0, queue limit 610
      packets output 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0
      shape:cir 2440000,  Bc 9760,  Be 9760
```

(shape parameter is rounded to 2,439,000 bps due to granularity)

```
        output bytes 0, shape rate 0 bps

    class-map:dscp1 (match-any)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      match:ip dscp 1
        0 packets, 0 bytes
        30 second rate 0 bps
      queue size 0, queue limit 100000
      packets output 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0
      bandwidth:kbps 400000, weight 64
```

(bandwidth parameter is rounded to 397,592 kbps due to granularity)

```
    class-map:dscp2 (match-all)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      match:ip dscp 2
      Priority:21% (130620 kbps), burst bytes 3265500, b/w exceed drops:0
```

(Priority parameter is rounded to 129,278 kbps due to granularity)

```
    class-map:class-default (match-any)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      match:any
        0 packets, 0 bytes
        30 second rate 0 bps
      queue size 0, queue limit 11422
      packets output 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0
```

Configuration Example

This example shows how to configure traffic classes, create a service policy, and attach it to an interface:

Step 1 Two traffic classes are created and their match criteria are defined. For the first traffic class, called class1, DSCP 30 is used as the match criterion. For the second traffic class, called class2, DSCP 10 is used as the match criterion. Packets are checked against the contents of these criteria to determine if they belong to the traffic class.

```
Router(config)# class-map class1
Router(config-cmap)# match ip dscp 30
Router(config-cmap)# exit

Router(config)# class-map class2
Router(config-cmap)# match ip dscp 10
Router(config-cmap)# exit
```

Step 2 A service policy called policy1 is defined to associate QoS features with the two traffic classes—class1 and class2. The match criteria for these traffic classes were defined in Step 1.

```
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 30000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# class class2
Router(config-pmap-c)# bandwidth 20000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)#
```

Step 3 After you define a service policy, you can attach it to one or more interfaces to specify a service policy for those interfaces. Although you can assign the same service policy to multiple interfaces, each interface can have only one service policy attached at the input and one policy map attached at the output at one time.

```
Router(config)# interface pos 2/0
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

In the following example, CBWFQ is configured on an OC-3 link. Class foo and class bar are guaranteed minimum bandwidth of 20 percent and 30 percent of link rate, respectively. The remaining bandwidth is equally distributed between class-default and class baz, each gets 25 percent of link rate. However, because class baz is shaped to receive a maximum of 15 Mbps (or 10 percent of the link rate), the bandwidth allocated to class baz is capped by this specified shape value, and class baz is guaranteed a minimum bandwidth of 15 Mbps.

```
class-map foo
    match ip dscp 10
class-map bar
    match ip dscp 20
class-map baz
    match ip dscp 30

policy-map foobar
    class foo
        bandwidth percent 20%
    class bar
        bandwidth percent 30%
    class baz
        shape average 15000000

int pos2/0
```

```
service-policy output foobar
```

Configuring Low Latency Queuing

Low latency queuing (LLQ) lets you specify low-latency behavior for a traffic class. LLQ allows delay-sensitive data to be given preferential treatment. You can give one or more classes priority status. You configure LLQ with the **priority** command.

The **priority** command configures guaranteed bandwidth to a priority class under worst-case congestion scenarios.

Restrictions and Usage Guidelines

The LLQ restrictions and usage guidelines are as follows:

- LLQ is not supported in the input direction.
- CBWFQ and LLQ both provide guaranteed bandwidth for their respective classes. The sum of all bandwidth on a link guaranteed by CBWFQ for non-priority classes and LLQ for priority classes cannot exceed 99% of the total available link bandwidth. Additionally, Cisco recommends that the sum of all the LLQ classes bandwidth remain below 25% of the main interface bandwidth. For more information on CBWFQ, see the "Configuring Class-Based Weighted Fair Queuing" section on page 9-7.
- **OSM support**—LLQ is supported in POS mode on the 2-port OC-48c/STM-16 POS/DPT OSMs; it is not supported in DPT mode. CBWFQ is not supported on the OSM-4GE-WAN-GBIC.
- **Bandwidth granularity**—On the OSMs, the granularity of the priority rate is 1/255 of the link rate or the hierarchical shape rate. The OSMs automatically round the configured rate to the nearest multiple of 1/255. The **show policy-map interface** command displays the rounded priority rate.
- **Minimum priority rate**—As shown in Table 9-2, the **priority** command has a minimum rate. For OC-3c and above interfaces, the minimum rate is 1/255 of the physical interface speed. For T3 and lower interfaces, the minimum rate is 256,000 bps.
- **Bandwidth allocation**—When a link is not under congested conditions, the unused (or excess) bandwidth is shared among all classes. The excess bandwidth available to a class is in proportion to its guaranteed bandwidth specified by the **priority** or **bandwidth** commands. For example, if one class is guaranteed 20% of the link and a second class is guaranteed 10% of the link, then the first class receives twice as much excess bandwidth as the second class.
- **LLQ burst**—The OSMs have a fixed burst size. They do not support the **priority** command burst parameter.
- **Bandwidth command**—You cannot configure the **bandwidth** command for a priority class.
- Because the MSFC and PFC do not support LLQ, configuring LLQ on a system configured with a Supervisor Engine 720 and an OSM-1CHOC12/T1-SI, might cause the output of the show policy-map interface command to display a packet counter of 0 for a serial interface.

Table 9-2 Minimum QoS Rates for priority Command

Physical Interface	Minimum Rate for priority Command (kbps)
T3 and below	256
OC3 POS	607

Table 9-2 Minimum QoS Rates for priority Command

Physical Interface	Minimum Rate for priority Command (kbps)
OC12 POS	2439
OC48 POS	9756
GE WAN	Not supported
Enhanced GE WAN	3921
Hierarchical traffic shaping (child)	Greater of (a) 256 or (b) 1/255 of the parent rate.

Note Starting with Cisco IOS Release 12.2(18)SXE, the Low Latency Queuing feature is change for the POS OSMs and the OSM-2+4GE-WAN OSM. For further information see Configuring Strict Priority LLQ Support on POS Optical Service Modules, page 1-29, and Configuring Strict Priority Low Latency Queuing (LLQ) Support on the OSM-2+4GE-WAN+, page 1-4.

Configuration Tasks

To configure LLQ, use the Modular QoS CLI. Define the class of traffic with the **class-map** command, create a policy map that contains the **priority** command, and apply the policy to the appropriate interface with the **service-policy** command.

To configure the class of traffic, see the "Configuring Classification" section on page 3. To configure a policy with LLQ and to assign the policy to an interface, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-name`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a predefined class included in the service policy.
Step 3	`Router(config-pmap-c)# priority bandwidth-kbps | percent % of available bandwidth` [1] [2]	Reserves a priority queue for CBWFQ traffic.
Step 4	`Router(config)# interface interface-name`	Specifies the interface to which the policy map will be applied.
Step 5	`Router(config-if)# service-policy [output policy-name]`	Attaches the specified policy map to the interface.

1. Only the parameters shown are supported.
2. See Table 9-1 on page 9-5 and Table 9-2 on page 9-12.

Configuration Example

This example shows how to configure a priority queue (with a guaranteed allowed bandwidth of 50 Mbps) reserved for traffic with an IP DSCP value of 40:

Step 1 Create traffic classes and specify match criteria by defining class maps.

```
Router(config)# class-map gold-data
Router(config-cmap)# match-any ip dscp 40
Router(config-cmap)# exit
Router(config)# class-map match bar
Router(config-cmap)# match-any ip dscp 8
Router(config-cmap)# exit
Router(config)#
```

Step 2 Create the policy map. In the example, a priority queue for the class gold-data is reserved with a guaranteed allowed bandwidth of 50 Mbps, and a bandwidth of 20 Mbps is configured for the class bar. The **service-policy** command attaches the policy map to interface pos 4/1.

```
Router(config)# policy-map policy1
Router(config-pmap)# class gold-data
Router(config-pmap-c)# priority 50000
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface pos 4/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

Configuring Weighted Random Early Detection

Weighted Random Early Detection (WRED) is a congestion-avoidance mechanism that takes advantage of the congestion-control mechanism of TCP. By selectively dropping packets based on IP precedence prior to periods of high congestion, WRED tells the packet source to decrease its transmission rate. Edge routers assign IP precedence to packets as they enter the network. WRED is useful on any output interface where you expect to have congestion. However, WRED is usually used in the core routers of a network rather than at the edge. WRED uses these precedences to determine how it treats different types of traffic.

When a packet arrives, the average queue size is calculated and one of the following events occurs:

- If the average queue size is less than the minimum queue threshold, the arriving packet is queued.
- If the average queue size is between the minimum queue threshold for that type of traffic and the maximum threshold for the interface, the packet is either dropped or queued, depending on the packet drop probability for that type of traffic.
- If the average queue size is greater than the maximum threshold, the packet is dropped.

Restrictions and Usage Guidelines

The WRED restrictions and usage guidelines are as follows:

- **OSM support**—WRED is only supported on Enhanced OSMs.
- WRED is not supported in the input direction.
- In the case of a user-defined class, random-detect is supported in association with shaping or bandwidth only.
- WRED is supported for DSCP and EXP. For DSCP, you configure WRED with the **random-detect dscp-based** command.
- IPv6 traffic for WRED configuration on OSMs are accounted in class-default within WRED as OSMs do not support WRED states for IPv6.

For more details on the queue calculations and how WRED works, refer to the section “About WRED” in the chapter “Congestion Avoidance Overview” in the *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/index.htm

Configuration Tasks

To configure WRED, use the Modular QoS CLI. Define the class of traffic with the **class-map** command, create a policy map that contains the **random-detect** command, and apply the policy to the appropriate interface with the **service-policy** command.

To configure the class of traffic, see the "Configuring Classification" section on page 9-3. To configure a policy with WRED and to assign the policy to an interface, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map` *policy-name*	Specifies the name of the policy map to configure.
Step 2	`Router(config-pmap)# class` *class-name*	Specifies the name of a predefined class included in the service policy.
Step 3	`Router(config-pmap-c)# random-detect`	Enables WRED.
Step 4	`Router(config)# interface` *interface-name*	Specifies the interface to which the policy map will be applied.
Step 5	`Router(config-if)# service-policy [output` *policy-name*`]`	Attaches the specified policy map to the interface.

Configuration Example

This example shows how to configure WRED.

```
Router# config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# policy-map wred_test
Router(config-pmap)# class class-default
Router(config-pmap-c)# random-detect
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface pos 7/1
Router(config-if)# service-policy output wred_test
Router(config-if)# end
Router# show policy-map interface pos 7/1
 POS7/1

  Service-policy output: wred_test

    Class-map: class-default (match-any)
      16634097 packets, 8217243918 bytes
      30 second offered rate 482198000 bps, drop rate 0 bps
      Match: any
      queue size 0, queue limit 128
      packets output 16634097, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0
      Random-detect:
        Exp-weight-constant: 3 (1/8)
        Mean queue depth: 0
        Class Random       Tail   Minimum   Maximum     Mark       Output
                drop       drop threshold threshold  probability  packets
        0     104806          0        32        64     1/10       3026812
        1     104569          0        36        64     1/10       3027050
        2     104732          0        40        64     1/10       3026884
        3     104169          0        44        64     1/10       3027449
        4     103047          0        48        64     1/10       3028569
        5     103156          0        52        64     1/10       3028460
```

```
6           0           0         56        64      1/10            0
7           0           0         60        64      1/10            0
```

Configuring Hierarchical Traffic Shaping

Hierarchical traffic shaping allows multiple classes of traffic to be shaped at a single rate. A hierarchical traffic shaping policy consists of a child policy that identifies one or more classes of traffic, and a parent policy that shapes the output of the traffic classes into a single shape rate. You can apply a nested policy to an interface or subinterface.

Restrictions and Usage Guidelines

The hierarchical traffic shaping restrictions and usage guidelines are as follows:

- Hierarchical traffic shaping is supported as an output policy only.
- Hierarchical traffic shaping is supported on the interfaces and subinterfaces for the OSM-2+4GE-WAN OSM.

Note Shape average is supported on the OSMs; shape peak is not supported on the OSMs.

- For the OSM-2+4GE-WAN OSM and enhanced POS OSMs starting with Cisco IOS release 12.2(18)SXE, the **police** command is not supported in a child class unless it is coupled with the **priority** command in the same child class.
- Hierarchical traffic shaping is supported on Frame Relay encapsulated subinterfaces on the original and enhanced POS OSMs and in POS mode on the DPT OSM.
- Hierarchical traffic shaping is supported on Frame Relay on encapsulated main interfaces using map class on enhanced POS OSMs.
- Hierarchical traffic shapping is supported on PPP encapsulated interfaces on the enhanced POS OSMs and in POS mode on the DPT OSM.

Note If a WAN interface on an OSM already has a hierarchical MQC policy map attached, do not attempt to convert it to a flat policy map. Conversely, do not attempt to convert a flat policy map to a hierarchical MQC policy map unless you first remove the policy from the attached interfaces before you convert the policy.

Configuration Task

To configure hierarchical traffic shaping, use the Modular QoS CLI to define the parent and child polices. For the child policy, define the classes of traffic with the **class-map** command and create a policy with the **policy-map** command. For the parent policy, create a policy with the **policy-map** command, apply the child policy to the default class with the **service-policy** command, and apply the parent policy to the appropriate interface with the **service-policy** command.

To configure the classes of traffic, see the "Configuring Classification" section on page 9-3. To configure the child and parent policies for hierarchical traffic shaping, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map child-policy-name`	Specifies the name of the child policy map to configure.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a predefined class included in the service policy.
Step 3	`Router(config-pmap-c)# priority bandwidth-kbps | percent % of available bandwidth`[1]	Gives priority to a class of traffic belonging to the policy map.
Step 4	`Router(config-pmap)# class class-name`	Specifies the name of a predefined class included in the service policy.
Step 5	`Router(config-pmap-c)# bandwidth bandwidth-kbps | percent % of available bandwidth`[2]	Specifies the percentage of available bandwidth in kilobits per second to be assigned to packets that meet the match criteria of the associated traffic class.
Step 6	`Router(config)# policy-map parent-policy-name`	Specifies the name of the parent policy map to configure.
Step 7	`Router(config-pmap)# class class-name`	Specifies the name of a predefined class included in the service policy.
Step 8	`Router(config-pmap-c)# shape average cir`	Shapes traffic to the indicated bit rate for the specified class.
Step 9	`Router(config-pmap-c)# service-policy child-policy-name`	Links the parent and child policy and class.
Step 10	`Router(config)# interface interface-name`	Specifies the interface to which the policy map will be applied.
Step 11	`Router(config-if)# service-policy [output parent-policy-name]`	Attaches the specified nested parent and child policies to the interface.

1. Only the parameters shown are supported.
2. Only the parameters shown are supported.

Configuration Example

This example shows a nested traffic policy configuration where traffic matching the class called voice will be guaranteed 3200 Kbps, or 10% of parent_policy's shape average:

```
Router(config)# class-map match-all voice
Router(config-cmap)# match ip dscp 5
Router(config-cmap)# exit
```

```
Router(config)# policy-map child_policy
Router(config-pmap)# class voice
Router(config-pmap-c)# priority percent 10
Router(config-pmap-c)# exit
Router(config-pmap)# exit

Router(config)# policy-map parent_policy
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 32000000
Router(config-pmap-c)# service-policy child_policy
Router(config-pmap-c)# exit
Router(config-pmap)# exit

Router(config)# interface Serial6/1:1.1 point-to-point
Router(config-subif)# service-policy parent_policy
```

Configuring Queue Limit

For the class-based traffic shaping and CBWFQ features, you can specify or modify the maximum number of packets the queue can hold for a class policy configured in a policy map using the **queue-limit** command from policy-map class configuration mode.

Restrictions and Usage Guidelines

The queue limit restrictions and usage guidelines are as follows:

- **OSM queue-limit values**—The default queue-limit value is chosen as a function of the OSM card type and the link encapsulation configured for an interface. It is not chosen based on the bandwidth assigned to the traffic class or the amount of buffer memory. You can change the default queue limit to one of the following values: 18, 25, 42, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, or 32768.

Note If you use queue-limit values other than the default values, the value is rounded down. For example, a queue-limit value of 3000 is rounded to 2048.

- In the case of a user-defined class, queue-limit is supported in association with shaping and bandwidth only.
- Flat policy maps with CBWFQ classes on subinterfaces with smaller packet sizes (less than 128 packets) may experience tail drops because of the default queue size (128 packets) (CSCeg73678). For these classes, if the tail drop is unacceptable, use the queue-limit to adjust the queue size to a larger value.

Configuring Tasks

To configure the queue-limit, use the Modular QoS CLI. Define the class of traffic with the **class-map** command, create a policy map that contains the **queue-limit** command, and apply the policy to the appropriate interface with the **service-policy** command.

To configure the class of traffic, see the "Configuring Classification" section on page 9-3. To configure a policy with queue-limit and to assign the policy to an interface, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-name`	Specifies the name of the policy map to configure.
Step 2	`Router(config-pmap)# class class-name`	Specifies the name of a predefined class included in the service policy.
Step 3	`Router(config-if)# queue-limit number-of-packets`	Specifies the maximum number of packets the queue can hold for a class policy configured in a policy map.
Step 4	`Router(config)# interface interface-name`	Specifies the interface to which the policy map will be applied.
Step 5	`Router(config-if)# service-policy [input | output policy-name]`	Attaches the specified policy map to the interface.

Note To remove the queue packet limit from a class, use the **no queue-limit** form of this command.

Configuration Example

In the following example, a policy map called policy11 is configured to contain policy for a class called cls203, and the queue limit is set to 42 packets.

```
Router(config)# policy-map policy11
Router(config-pmap)# class cls203
Router(config-pmap-c)# queue-limit 42
```

Configuring QoS: Match VLAN

The QoS: Match VLAN feature provides trunk VLAN match and classification in Modular QoS Command-Line Interface (MQC) class maps for the OSM-2+4GE-WAN+ interface.

Note This feature was first supported in Cisco IOS Release 12.2(18)SXE.

Classification based on the **match vlan** command is supported by the following QoS features:

- Ingress/egress shaping
- Egress class based weighted fair queuing (CBWFQ)
- Egress low latency queuing (LLQ Strict Priority)
- Egress weighted random early detection (WRED)

- Hierarchical QoS on non-default class

Restrictions and Usage Guidelines

- In a hierarchical policy, match VLAN classes in a child policy are not supported. For example, the following configuration is not supported:

```
!
policy not-supported-policy
   class class-default
    shape average 100000000
    service-policy child-m-vlan

Policy Map child-m-vlan
class vlan150
bandwidth 20 (%)
!
```

- This feature supports the **class-map match-all** and **match-any** parameters.
- A match VLAN policy is always applied to a main interface.

Configuration Tasks

To configure QoS: Match VLAN support, perform the following tasks, beginning in global configuration mode:

	Command or Action	Purpose
Step 1	`Router(config)# class-map` *class-map-name* **Example:** `Router(config)# class-map vlan100`	Specifies the name of the class map to be created or modified.
Step 2	`Router(config-cmap)# match vlan` *vlan-id* **Example:** `Router(config)# match vlan 100`	Specifies the VLAN ID used as match criteria. The valid value range is 1 to 4095. Alternately, a range of VLAN IDs can be used as match criteria. Use a hyphen to separate the VLAN IDs in each range. Use a space to separate each range of VLANs.

Configuration Examples

The following example shows a sample hierarchical match VLAN policy:

```
Router# configure terminal
Router(config)# policy-map vlan-pol-example
Router(config-pmap)# class vlan150
Router(config-pmap-c)# shape average 200000000 800000 800000
Router(config-pmap-c)# service-policy child
Router(config-pmap-c)# exit
Router(config-pmap)# class vlan170
Router(config-pmap-c)# shape average 200000000 800000 800000
Router(config-pmap-c)# service-policy child
!
Router(config)# policy-map child
Router(config-pmap)# Class prec2
```

```
Router(config-pmap-c)# bandwidth 20 (%)
Router(config-pmap)# Class prec3
Router(config-pmap-c)# bandwidth 30 (%)
Router(config-pmap)# Class prec1
Router(config-pmap-c)# bandwidth 10 (%)
```

The following example creates a class map for multiple VLANs, creates a policy map for shaping, and attaches an egress interface:

```
Router# configure terminal
Router(config)# class-map vlan_multi
Router(config-cmap)# match vlan 100-102 200 300
Router(config-cmap# exit
Router(config)# policy-map pol_multi_vlan
Router(config-pmap)# class vlan_multi
Router(config-pmap-c)# shape average 100000000
Router(config-pmap-c)# exit
Router(config)# interface ge7/1.100
Router(config-subif)# encapsulation dot1q 100
Router(config-subif)# no shut
Router(config-int)# interface ge7/1.101
Router(config-subif)# encapsulation dot1q 101
Router(config-subif)# no shut
Router(config-int)# interface ge7/1.102
Router(config-subif)# encapsulation dot1q 102
Router(config-subif)# no shut
Router(config-int)# interface ge7/1.200
Router(config-subif)# encapsulation dot1q 200
Router(config-subif)# no shut
Router(config-int)# interface ge7/1.300
Router(config-subif)# encapsulation dot1q 300
Router(config-subif)# no shut
Router(config-subif)# exit
Router(config)# interface ge7/1
Router(config-int)# service-policy output pol_multi_vlan
Router(config-int)# no shut
```

Distribution of Remaining Bandwidth

You can use MQC to specify how the "remaining" bandwidth is distributed among the output queues on a Cisco 7600 router interface or subinterface. "Remaining" bandwidth is the available bandwidth left on an interface or subinterface after all guaranteed traffic is accounted for. The amount of remaining bandwidth available for use is determined by the excess information rate (EIR) configured for the queue.

In MQC, the **bandwidth remaining percent** command allows you to configure the remaining bandwidth for output queues. See the "bandwidth remaining percent" section for more information.

The following example shows how to use the **bandwidth remaining percent** command to distribute percentages of remaining bandwidth to various traffic classes in a policy map.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# policy-map myPolicy
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining percent 20
Router(config-pmap-c)# class prec1
Router(config-pmap-c)# bandwidth remaining percent 30
Router(config-pmap-c)# class prec2
Router(config-pmap-c)# bandwidth remaining percent 10
Router(config-pmap-c)# bandwidth percent 50
Router(config-pmap-c)# ^Z
```

```
Router#
20:44:36: %SYS-5-CONFIG_I: Configured from console by console
Router# show policy-map myPolicy
  Policy Map myPolicy
    Class prec1
         bandwidth remaining percent 30
    Class prec2
         bandwidth percent 50
         bandwidth remaining percent 10
    Class class-default
         bandwidth remaining percent 20
```

Command Reference

To specify how the "remaining" bandwidth is distributed among the output queues on a Cisco 7600 series router interface or subinterface, use the MQC **bandwidth remaining percent** command in policy-map class configuration mode. To remove the percentage of remaining bandwidth specified for a traffic class, use the **no** form of this command.

bandwidth remaining percent *percentage*

no bandwidth remaining percent *percentage*

Syntax Description

percentage	Specifies a percentage value for the amount of guaranteed bandwidth, based on a relative percent of available bandwidth, to be assigned to the class. The percentage can be a number between 1 and 99.

Defaults

This command has no default behavior or values.

Command Modes

Modular QoS policy-map class configuration

Command History

12.2(18)SXE	This command was introduced on the Cisco 7600 series router.

Usage Guidelines

Use the MQC bandwidth remaining percent command to specify how the "remaining" bandwidth is distributed among the output queues on a Cisco 7600 series router interface or subinterface. "Remaining" bandwidth is the available bandwidth left on an interface or subinterface after all guaranteed traffic is accounted for.

The bandwidth remaining percent command allows you to configure the remaining bandwidth for output queues. The percentage parameter specified with the bandwidth remaining percent command is translated into an internal excess information rate (EIR) value between 0 and 255. The aggregate of all user-configured EIR bandwidth percentages cannot exceed 100 percent.

If the aggregate of all remaining bandwidth is less than 100 percent, the remainder is evenly split among user queues (including the default queue) that do not have a remaining bandwidth percentage configured. The minimum EIR value of each output queue is 1.

The EIR parameter for the network control queue is fixed at 128 and is not configurable.

If you have not configured a committed information rate (CIR) value for the default queue and it is the only user queue, the default queue receives half of the remaining bandwidth percentage of the network control queue.

Unsupported Frame Relay-Specific QoS Features

The following Frame-Relay specific QoS features are not supported:

- Adaptive traffic shaping
- Adaptive policing
- DLCI priority levels
- DE bit support
- The **fair-queue** command and **random-detect** command on the main interface

Cisco IPv6 QoS on the OSMs

Cisco IPv6 QoS on the OSMs supports all QoS features supported with IPv4 QoS. For general information on Cisco IPv6 QoS, see *Implementing QoS for IPv6 for Cisco IOS Software* at: http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_c/sa_qosv6.htm

Restrictions and Usage Guidelines

The Cisco IPv6 QoS restrictions and usage guidelines are as follows:

- The **match precedence** or **match dscp** command match both IPv4 and IPv6 traffic based upon the QoS marking in the packet Layer 3 header.
- In conjunction with the **match precedence** or **match dscp** command, you can specify the **match protocol ipv6** command in the class map to match only IPv6 precedence or DSCP.
- In conjunction with the **match precedence** or **match dscp** command, you can specify the **match protocol ip** command in the class map to match only IPv4 precedence or DSCP.
- IPv6 QOS is supported only on Enhanced OSMs.
- There can only be one **match protocol** command per class map.

CHAPTER 1

Configuring Destination Sensitive Services on the Optical Services Modules

This chapter describes how to configure Destination Sensitive Services (DSS) on the Optical Services Modules (OSMs).

This chapter consists of these sections:

Understanding Destination Sensitive Services

DSS allows traffic accounting and traffic shaping to known autonomous system numbers in order to engineer and plan network circuit peering and transit agreements. DSS is supported on ingress WAN ports on the following OSMs:

- OC-3 POS:
  - OSM-4OC3-POS-SI
  - OSM-8OC3-POS-SI, SL
  - OSM-16OC3-POS-SI, SL
  - OSM-4OC3-POS-SI+
  - OSM-8OC3-POS-SI+, SL+
  - OSM-16OC3-POS-SI+, SL+
- OC-12 POS:
  - OSM-2OC12-POS-MM, SI, SL
  - OSM-4OC12-POS-MM, SI, SL
  - OSM-2OC12-POS-MM+, SI+, SL+
  - OSM-4OC12-POS-MM+, SI+, SL+
- OC-48 POS:
  - OSM-1OC48-POS-SS, SI, SL
  - OSM-1OC48-POS-SS+, SI+, SL+
- Channelized:

  - OSM-1CHOC48/T3-SS
  - OSM-1CHOC12/T3-SI
  - OSM-1CHOC12/T1-SI
  - OSM-12CT3/T1
- OC-48 POS/DPT:
  - OSM-2OC48/1DPT-SS, SI, SL
- Gigabit Ethernet
  - OSM-4GE-WAN-GBIC
  - OSM-2+4GE-WAN+

DSS supports two separate services:

- Destination Sensitive Billing (DSB)

  DSB allows accounting based on destination traffic indexes and provides a means of classifying customer traffic according to the route that the traffic travels. Trans-Pacific, Trans-Atlantic, satellite, domestic, and other provider traffic can be identified and accounted for on a destination network basis when the customer traffic is on a unique software interface. DSB provides packet and byte counters, which represent counts for IP packets per destination network. DSB is implemented using route-maps to classify the traffic into one of seven possible indexes, which represent a traffic classification.
- Destination Sensitive Traffic Shaping (DSTS)

  DSTS performs inbound traffic shaping based on the destination traffic index configuration.

  See the "Configuring Destination Sensitive Services" section on page 10-2 for configuration information.

Note Although the CLI allows it, DSB and DSTS should not be applied to the same interface.

Configuring Destination Sensitive Services

When you configure ingress DSS and DSB on a WAN interface, the traffic coming in over the WAN interface is classified and accounted for.

These sections describe how to configure ingress DSS and DSB:

Configuring Ingress DSS

To configure ingress DSS on your OSM, perform this task:

	Command	Purpose
Step 1	`Router(config)# route-map map-tag [permit | deny] [sequence-number]`	Defines the conditions for redistributing routes from one routing protocol into another.
Step 2	`Router(config-route-map)# match community-list community-list-number`	Matches a BGP community.
Step 3	`Router(config-route-map)# set traffic-index 1-8`	Creates the BGP traffic index.
Step 4	`Router(config)# router bgp autonomous-system`	Configures BGP routing.
Step 5	`Router(config-router)# table-map route-map-name`	Modifies metric and tag values when the IP routing table is updated with BGP learned routes.
Step 6	`Router(config)# ip bgp-community new-format`	Displays BGP communities in the format AA:NN (autonomous system-community number/2-byte number).
Step 7	`Router(config)# ip community-list 40 100:120`	Creates a community list for BGP and controls access to it.
Step 8	`Router(config-pmap)# class-map [match-all | match-any]`	Creates a class map to be used for matching packets to a class you define and specifies the criteria to match on.
Step 9	`Router(config-pmap-c)# match bgp-index bgp-index`	Identifies a specific BGP index to match on.
Step 10	`Router(config)# policy-map policy_map`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
Step 11	`Router(config-pmap)# class class-name`	Defines the classes you want the service policy to contain.
Step 12	`Router(config-pmap-c)# shape [average | peak] mean-rate [[burst-size] [excess-burst-size]]`	Shapes traffic to the indicated bit rate.
Step 13	`Router(config)# interface`	Specifies the WAN interface to which the policy map will be applied.
Step 14	`Router(config-if)# service-policy input policy-map`	Attaches the specified policy map to the input interface.

To configure ingress DSS, perform this task:

Step 1 Classify the BGP routes by creating BGP traffic indexes:

```
Router(config)# route-map dss-map1 permit 1
Router(config-route-map)# match community 40
Router(config-route-map)#s et traffic-index 1
Router(config-route-map)# exit
Router(config)# route-map dss-map3
Router(config-route-map)# match community 50
Router(config-route-map)# set traffic-index 3
Router(config-route-map)# exit
Router(config)# route-map dss-map7
Router(config-route-map)# match community 60
Router(config-route-map)# set traffic-index 7
Router(config-route-map)# exit
```

Step 2 Enable BGP routing for the autonomous system and apply the route-maps:

```
Router(config)# router bgp 100
Router(config-router)# table-map dss-map1
Router(config)# ip bgp-community new-format
Router(config)# ip community-list 10 100:120
Router(config-router)# exit
```

Step 3 Map the BGP index to the traffic index classes:

```
Router(config)# class-map dss1
Router(config-cmap)# match bgp-index 1
Router(config-cmap)# exit
Router(config)# class-map dss3
Router(config-cmap)# match bgp-index 3
Router(config-cmap)# exit
Router(config)# class-map dss7
Router(config-cmap)# match bgp-index 7
Router(config-cmap)# exit
```

Step 4 Configure the policy-maps and traffic shaping parameters:

```
Router(config)# policy-map dss-policy
Router(config-pmap)# class dss1
Router(config-pmap-c)#s hape average 64000
Router(config-pmap-c)# exit
Router(config-pmap)# class dss3
Router(config-pmap-c)# shape average 1024000
Router(config-pmap-c)# exit
Router(config-pmap)# class dss7
Router(config-pmap-c)# shape average 384000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)#
```

Step 5 Apply the policy configuration to the appropriate WAN interface:

```
Router(config)# interface ge-wan 4/1
Router(config-if)# service-policy input dss-policy
Router(config-if)# exit
Router(config)
```

Step 6 Display DSS configuration information:

```
Router# show ip cef 8.1.1.0
8.1.1.0/24, version 340, cached adjacency 19.1.1.3
0 packets, 0 bytes, Precedence critical (5), traffic_index 1
  via 19.1.1.3, 0 dependencies, recursive
    next hop 19.1.1.3, Vlan19 via 19.1.1.3/32
    valid cached adjacency

Router# show ip bgp 8.1.1.0
BGP routing table entry for 8.1.1.0/24, version 14
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  12
    19.1.1.3 from 19.1.1.3 (19.1.1.3)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 100:120

Router# show class-map dss1
 Class Map match-all dss1 (id 8)
   Match bgp-index 1

Router# show policy-map interface ge-wan 4/1

 GigabitEthernet4/1

  service-policy input: dss-policy

    class-map: dss1 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      match: bgp-index 1
      queue size 0, queue limit 2
      packets input 0, packet drops 0
<ouput truncated>

Router(config)#
```

Configuring Ingress DSB

To configure ingress DSB on your OSM, display the BGP policy accounting configuration, and clear the ingress DSB counters, perform this task:

Step 1 Enable ingress DSB on the port:

```
Router(config)# interface GE-WAN6/2
Router(config-if)# ingress-dsb
```

Step 2 Enable BGP policy accounting on the interface:

```
Router(config-if)# bgp-policy accounting
Router(config-if)# end
```

Step 3 Display the BGP policy accounting configuration on the input interface:

```
Router# show cef interface GE-WAN6/2 policy
GE_WAN6/2 is up (if_number 18)
  Corresponding hwidb fast_if_number 18
  Corresponding hwidb firstsw->if_number 18
BGP based Policy accounting is enabled
 Index         Packets            Bytes
     1         5259200       315552000
     2         5259300       315558000
     3         5259300       315558000
     4         5258900       315534000
     5         5258916       315534960
     6         5259000       315540000
     7         5259100       315546000
     8               0               0
Router#
```

Step 4 Clear all ingress DSB counters:

```
Router# clear cef interface policy-statistics
```

Step 5 Clear the ingress DSB counter for a specific interface:

```
Router# clear cef interface GE-WAN6/2 policy-statistics
```

CHAPTER 1

Configuring Multiprotocol Label Switching on the Optical Services Modules

This chapter describes how to configure Multiprotocol Label Switching (MPLS) and Any Transport over Multiprotocol Label Switching (AToM) on the Optical Services Modules (OSMs).

This chapter consists of these sections:

Configuring MPLS

These sections describe MPLS and provides configuration information:

- Configuring MPLS, page 11-6

Understanding MPLS

MPLS uses label switching to forward packets over various link-level technologies such as Packet-over-SONET, Frame Relay, ATM, and Ethernet. Labels are assigned to packets based on groupings or forwarding equivalence classes (FECs). Packets belonging to the same FEC get similar treatment. The label is added between the Layer 2 and the Layer 3 header (in a packet environment) or in the virtual path identifier/virtual channel identifier (VPI/VCI) field (in ATM networks).

In an MPLS network, the edge router performs a label lookup of the incoming label, swaps the incoming label with an outgoing label, and sends the packet to the next hop. Labels are imposed on packets only at the ingress edge of the MPLS network and are removed at the egress edge. The core network reads the labels, applies the appropriate services, and forwards the packets based on the labels.

MPLS Support on OSMs

MPLS is supported on the following Catalyst 6000 family and Cisco 7600 series OSMs:

- OC-3 POS:
  - OSM-4OC3-POS-SI
  - OSM-8OC3-POS-SI, SL
  - OSM-16OC3-POS-SI, SL
  - OSM-4OC3-POS-SI+
  - OSM-8OC3-POS-SI+, SL+
- OC-12 POS:
  - OSM-2OC12-POS-MM, SI, SL
  - OSM-4OC12-POS-MM, SI, SL
  - OSM-2OC12-POS-MM+, SI+, SL+
  - OSM-4OC12-POS-MM+, SI+, SL+
- OC-12 ATM:
  - OSM-2OC12-ATM-MM
  - OSM-2OC12-ATM-SI
  - OSM-2OC12-ATM-MM+
  - OSM-2OC12-ATM-SI+
- OC-48 POS:
  - OSM-1OC48-POS-SS, SI, SL
  - OSM-1OC48-POS-SS+, SI+, SL+
- Channelized:
  - OSM-1CHOC48/T3-SS
  - OSM-1CHOC12/T3-SI
  - OSM-1CHOC12/T1-SI

  - OSM-12CT3/T1

Note You cannot use channelized OSMs as MPLS core-facing interfaces.

- OC-48 POS/DPT:
  - OSM-2OC48/1DPT-SS, SI, SL

Note OSM-2OC48/1DPT-SS, SI, SL support MPLS in POS mode; OSM-2OC48/1DPT-SS, SI, SL also support MPLS in DPT mode with SUP720-3BXL-based systems.

- Gigabit Ethernet
  - OSM-4GE-WAN-GBIC
  - OSM-2+4GE-WAN+
- WS-X6182-2PA FlexWAN
- WS-X6582-2PA Enhanced FlexWAN

Supported Features

The following features are supported with the SUP720-3BXL and the supervisor engine 2:

Note Features in the Cisco IOS 12.2SX releases that are also supported in the Cisco IOS 12.2 mainline, 12.2T and 12.2S releases are documented in the corresponding publications for those releases. When applicable, this section refers to those publications for platform-independent features supported in the Cisco IOS 12.2SX releases. The Cisco IOS 12.2S releases do not support software images for the Cisco 7600 series routers, and the Cisco IOS 12.2S publications do not list support for the Cisco 7600 series routers.

- Multi-VRF for CE Routers (VRF Lite)—VRF-lite is a feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. See http://www.cisco.com/en/US/products/hw/routers/ps259/prod_bulletin09186a00800921d7.html.

  Note Multi-VRF for CE Routers (VRF Lite) is supported with the following features: IPv4 forwarding between VRFs interfaces, IPv4 ACLs, and IPv4 HSRP. Starting with Cisco IOS Release 12.2(18)SXE, Multi-VRF for CE Routers (VRF Lite) is supported with IPv4 multicast.

  Note Multi-VRF for CE Routers (VRF Lite) is also supported with the Supervisor Engine 720 with PFC3A.

- MPLS Label Distribution Protocol (LDP)—MPLS label distribution protocol (LDP), as standardized by the Internet Engineering Task Force (IETF) and as enabled by Cisco IOS software, allows the construction of highly scalable and flexible IP Virtual Private Networks (VPNs) that

support multiple levels of services. See http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s14/fs2sldp.htm.

- Multiprotocol Label Switching (MPLS) on Cisco Routers—This feature provides basic MPLS support for imposing and removing labels on IP packets at label edge routers (LERs) and switching labels at label switch routers (LSR). See http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120st/120st21/fs_rtr.htm.
- MPLS Traffic Engineering–DiffServ Aware (DS-TE)—This feature provides extensions made to Multiprotocol Label Switching Traffic Engineering (MPLS TE) to make it DiffServ aware, allowing constraint-based routing of guaranteed traffic. See http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s18/fsdserv3.htm.
- MPLS Traffic Engineering Forwarding Adjacency—This feature allows a network administrator to handle a traffic engineering, label-switched path (LSP) tunnel as a link in an Interior Gateway Protocol (IGP) network based on the Shortest Path First (SPF) algorithm. For information on forwarding adjacency with Intermediate System-to-Intermediate System (IS-IS) routing, see http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s18/fstefa_3.htm.

  For information on forwarding adjacency with Open Shortest Path First (OSPF) routing, see http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s24/ospffa.htm.
- MPLS Traffic Engineering (TE) Interarea Tunnels—This feature allows the router to establish MPLS TE tunnels that span multiple Interior Gateway Protocol (IGP) areas and levels, removing the restriction that had required the tunnel head-end and tail-end routers to be in the same area. See http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s18/fsiarea3.htm.
- MPLS Virtual Private Networks (VPNs)—This feature allows you to deploy scalable IPv4 Layer 3 VPN backbone services over a Cisco IOS network. See http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120st/120st21/fs_vpn.htm.
- MPLS VPN Carrier Supporting Carrier (CSC)—The feature enables one MPLS VPN-based service provider to allow other service providers to use a segment of its backbone network. See http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/ftcsc8.htm.
- MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution—This feature enables you to configure your carrier supporting carrier network to enable Border Gateway Protocol (BGP) to transport routes and Multiprotocol Label Switching (MPLS) labels between the backbone carrier provider edge (PE) routers and the customer carrier customer edge (CE) routers. See http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ftcscl13.htm.
- MPLS VPN—Interautonomous System Support—This feature allows an MPLS VPN to span service providers and autonomous systems. See http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s24/fsias24.htm.
- MPLS VPN—Inter-AS—IPv4 BGP Label Distribution: This feature enables you to set up a Virtual Private Network (VPN) service provider network so that the autonomous system boundary routers (ASBRs) exchange IPv4 routes with Multiprotocol Label Switching (MPLS) labels of the provider

edge (PE) routers.See http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ftiasl13.htm.

- Hot Standby Router Protocol (HSRP) Support for Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs)—This feature ensures that the HSRP virtual IP address is added to the correct IP routing table and not to the default routing table. See http://www.cisco.com/en/US/products/sw/iosswrel/ps1834/products_feature_guide09186a008008021e.html.
- OSPF Sham Link: OSPF Sham-Link Support for MPLS VPN—This feature allows you to use a sham-link to connect Virtual Private Network (VPN) client sites that run the Open Shortest Path First (OSPF) protocol and share backdoor OSPF links in a Multiprotocol Label Switching (MPLS) VPN configuration. See http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/ospfshmk.htm.
- BGP Multipath Load Sharing for eBGP and iBGP—This feature allows you to configure multipath load balancing with both external BGP (eBGP) and internal BGP (iBGP) paths in Border Gateway Protocol (BGP) networks that are configured to use Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). See http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t4/fteibmpl.htm.
- Any Transport over MPLS (AToM). Transports Layer 2 packets over a Multiprotocol Label Switching (MPLS) backbone. See the "Any Transport over MPLS" section on page 11-23.

MPLS Limitations and Restrictions

The following platform-specific limitations and restrictions apply to the MPLS support on the OSM modules:

- MPLS Limitations, page 11-5
- MPLS Traffic Engineering with Fast ReRoute (FRR) protection—this feature is not yet supported.

MPLS Limitations

The following MPLS limitations apply:

- MTU checking and fragmentation is not supported on the OSMs except that checking is supported on the OSM-2+4GE-WAN+ on the receive path.
- With supervisor engine 2, MPLS Provider (P) functionality is not supported on Ethernet interfaces that also support Layer 2 switching. The only way to support P functionality on these interfaces is to create a trunk from a Gigabit Ethernet interface on, for example, a WS-6516-GBIC module to an interface on the OSM-4GE-WAN module that is configured to allow P switching. The interface on the WS-6516-GBIC module should be placed in trunking mode, and appropriate subinterfaces should be created on the OSM-4GE-WAN module interface. With SUP720-3BXL-based systems, MPLS Provider (P) functionality is supported.
- With supervisor engine 2, load sharing is supported on PE paths only and not on the P device.
- Encapsulation on the 2-Port OC-12 ATM OSM—MPLS is supported only when the interface is configured for AAL5SNAP (cell mode) encapsulation (default).

Note For information on other limitations and restrictions, see "MPLS VPN Limitations and Restrictions" section on page 11-20, "Ethernet over MPLS Restrictions" section on page 11-24, "ATM AAL5 over MPLS Restrictions" section on page 11-25, "ATM Cell Relay over MPLS Restrictions" section on page 11-25, "Frame Relay over MPLS Restrictions" section on page 11-25, and "Restrictions for VPLS" section on page 11-115.

Configuring MPLS

For information on configuring MPLS, refer to the *Multiprotocol Label Switching on Cisco Routers* feature module at the following URLs:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/mpls4t.htm

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt3/xcftagov.htm#1021991

HDLC Over MPLS

HDLC over MPLS encapsulates HDLC protocol data units (PDUs) in MPLS packets and forwards them across the MPLS network. The PE routers do not participate in any protocol negotiation or authentication.

HDLC Over MPLS Restrictions

The following restrictions pertain to the HDLC over MPLS feature:

- **Synchronous interfaces**: The connections between the CE and PE routers on both ends of the backbone must have similar link layer characteristics. The connections between the CE and PE routers must both be synchronous.
- **Interface configuration**: You must configure HDLC over MPLS on router interfaces only. You cannot configure HDLC over MPLS on subinterfaces.

Note For HDLCoMPLS, SUP720-PFC3B-based systems and SUP720-PFC3BXL-based systems require that the core-facing cards must be WAN cards (enhanced OSMs, FlexWAN and Enhanced FlexWAN modules, and Shared Port Adapter [SPA] Interface Processors [SIPs]).

Supported OSMs

The following OSMs support HDLC over MPLS:

- OC-3 POS:
  - OSM-4OC3-POS-SI+
  - OSM-8OC3-POS-SI+, SL+
- OC-12 POS:
  - OSM-2OC12-POS-MM+, SI+, SL+

  - OSM-4OC12-POS-MM+, SI+, SL+
- OC-48 POS:
  - OSM-2OC48/1DPT-SS, SI, SL

Note HDLCoMPLS is supported for POS mode only for the OSM-2OC48/1DPT-SS, SI, SL.

  - OSM-1OC48-POS-SS+, SI+, SL+

Configuring HDLC Over MPLS

With HDLC over MPLS, the whole HDLC packet is transported. The ingress PE router removes only the HDLC flags and frame check sequence (FCS) bits. The contents of the packet are not used or changed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial***slot/port*
4. **encapsulation** *encapsulation-type*
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial***slot/port* **Example:** `Router(config)# interface serial5/0`	Specifies a serial interface and enters interface configuration mode. You must configure HDLC over MPLS on router interfaces only. You cannot configure HDLC over MPLS on subinterfaces.

	Command or Action	Purpose
Step 4	**encapsulation** *encapsulation-type* **Example:** `Router(config-if)# encapsulation hdlc`	Specifies HDLC encapsulation.
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-fr-pw-switching)# xconnect 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

This example shows an HDLC over MPLS configuration and verification:

```
PE1# show run int pos1/8
Building configuration...

Current configuration : 137 bytes
!
interface POS1/8
 mtu 5000
 no ip address
 mls qos trust dscp
 clock source internal
 xconnect 33.33.33.33 101 encapsulation mpls
end

PE1# sh mpls l2 vc 101

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
PO1/8          HDLC                 33.33.33.33     101        UP
PE1#

PE1# sh mpls l2 vc 101 detail
Local interface: PO1/8 up, line protocol up, HDLC up
  Destination address: 33.33.33.33, VC ID: 101, VC status: up
    Tunnel label: imp-null, next hop point2point
    Output interface: PO4/4.1, imposed label stack {1396}
  Create time: 00:17:49, last status change time: 00:03:33
  Signaling protocol: LDP, peer 33.33.33.33:0 up
    MPLS VC labels: local 25, remote 1396
    Group ID: local 0, remote 0
    MTU: local 5000, remote 5000
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 1011, send 1010
    byte totals:   receive 104898, send 104562
    packet drops:  receive 0, send 0

PE1# sh mpls for | inc PO1/8
25     Untagged    l2ckt(101)        114705     PO1/8      point2point
PE1#

PE2#sh run int pos8/1
Building configuration...

Current configuration : 137 bytes
!
```

```
interface POS8/1
 mtu 5000
 no ip address
 mls qos trust dscp
 clock source internal
 xconnect 11.11.11.11 101 encapsulation mpls
end

PE2# sh mpls l2 vc 101

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
PO8/1          HDLC                 11.11.11.11     101        UP
PE2#sh mpls l2 vc 101 detail
Local interface: PO8/1 up, line protocol up, HDLC up
  Destination address: 11.11.11.11, VC ID: 101, VC status: up
    Tunnel label: imp-null, next hop point2point
    Output interface: PO8/4.1, imposed label stack {25}
  Create time: 00:12:37, last status change time: 00:06:19
  Signaling protocol: LDP, peer 11.11.11.11:0 up
    MPLS VC labels: local 1396, remote 25
    Group ID: local 0, remote 0
    MTU: local 5000, remote 5000
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 1028, send 1028
    byte totals:   receive 105960, send 105940
    packet drops:  receive 0, send 0

PE2# sh mpls for | inc PO8/1
1396   Untagged    l2ckt(101)        114634     PO8/1      point2point
PE2#
CE1#sh run int pos3/0/0
Building configuration...

Current configuration : 127 bytes
!
interface POS3/0/0
 ip address 130.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 clock source internal
end

CE2# sh run int pos3/0
Building configuration...

Current configuration : 123 bytes
!
interface POS3/0
 mtu 5000
 ip address 130.0.0.2 255.0.0.0
 no ip directed-broadcast
 crc 16
 clock source internal
end

CE1# ping
Protocol [ip]:
Target IP address: 130.0.0.2
Repeat count [5]: 1000
Datagram size [100]:
Timeout in seconds [2]:
```

```
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 1000, 100-byte ICMP Echos to 130.0.0.2, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (1000/1000), round-trip min/avg/max = 1/1/4 ms
```

Note Keepalives are end to end, that is, from CE to CE.

PPP Over MPLS

PPP over MPLS encapsulates PPP PDUs in MPLS packets and forwards them across the MPLS network. The PE routers do not participate in any protocol negotiation or authentication.

Note For PPPoMPLS, all PPP negotiations (for example, link control protocol [LCP]) are end to end, that is, from CE to CE. The PE routers do not participate in the PPP negotiations.

Supported OSMs

The following OSMs support HDLC over MPLS:

- OC-3 POS:
  - OSM-4OC3-POS-SI+
  - OSM-8OC3-POS-SI+, SL+
- OC-12 POS:
  - OSM-2OC12-POS-MM+, SI+, SL+
  - OSM-4OC12-POS-MM+, SI+, SL+
- OC-48 POS:
  - OSM-2OC48/1DPT-SS, SI, SL
  - OSM-1OC48-POS-SS+, SI+, SL+

PPP Over MPLS Restrictions

The following restrictions pertain to the PPP over MPLS feature:

- **Zero hops on a PE router**: Zero hops on one router is not supported. However, you can have back-to-back PE routers.
- **Synchronous interfaces**: The connections between the CE and PE routers on both ends of the backbone must have similar link layer characteristics. The connections between the CE and PE routers must both be synchronous.
- **Multilink PPP**: Multilink PPP (MLPPP) is not supported. You cannot configure a PPPoMPLS VC on a MLPPP interface on the PE router.

Note While MLPPPoMPLS is not supported, it can be emulated. To achieve this, each member link of the MLPPP bundle on a CE requires a corresponding PPPoMPLS tunnel on the PE router that it directly connects to. For example, if an MLPPP bundle is comprised of three member links, you must configure three PPPoMPLS tunnels on each PE with each tunnel corresponding to a member link.

Configuring PPP Over MPLS

With PPP over MPLS, the ingress PE router removes the flags, address, control field, and the frame check sequence (FCS).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial***slot/port*
4. **encapsulation** *encapsulation-type*
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**

Detailed Steps

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial***slot/port* **Example:** `Router(config)# interface serial5/0`	Specifies a serial interface and enters interface configuration mode. You must configure PPP over MPLS on router interfaces only. You cannot configure HDLC over MPLS on subinterfaces.

	Command or Action	Purpose
Step 4	**encapsulation** *encapsulation-type* **Example:** `Router(config-if)# encapsulation ppp`	Specifies PPP encapsulation.
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-fr-pw-switching)# xconnect 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

This example shows configuration and verification:

```
PE1# sh run int pos1/8
Building configuration...

Current configuration : 156 bytes
!
interface POS1/8
 mtu 5000
 no ip address
 encapsulation ppp
 mls qos trust dscp
 clock source internal
 xconnect 33.33.33.33 101 encapsulation mpls
end

PE2# sh run int pos8/1
Building configuration...

Current configuration : 156 bytes
!
interface POS8/1
 mtu 5000
 no ip address
 encapsulation ppp
 mls qos trust dscp
 clock source internal
 xconnect 11.11.11.11 101 encapsulation mpls
end
```

This example show how to verify the configuration:

```
PE1#
PE1# sh mpls l2 vc 101

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
PO1/8          PPP                  33.33.33.33     101        UP
PE1#

PE2# sh mpls l2 vc 101

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
PO8/1          PPP                  11.11.11.11     101        UP
PE2#

PE1# sh mpls l2 vc 101 detail
Local interface: PO1/8 up, line protocol up, PPP up
```

```
  Destination address: 33.33.33.33, VC ID: 101, VC status: up
    Tunnel label: imp-null, next hop point2point
    Output interface: PO4/4.1, imposed label stack {2530}
  Create time: 00:02:02, last status change time: 00:01:16
  Signaling protocol: LDP, peer 33.33.33.33:0 up
    MPLS VC labels: local 413, remote 2530
    Group ID: local 0, remote 0
    MTU: local 5000, remote 5000
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 19, send 18
    byte totals:   receive 1394, send 1058
    packet drops:  receive 0, send 0

PE2# sh mpls l2 vc 101 detail
Local interface: PO8/1 up, line protocol up, PPP up
  Destination address: 11.11.11.11, VC ID: 101, VC status: up
    Tunnel label: imp-null, next hop point2point
    Output interface: PO8/4.1, imposed label stack {413}
  Create time: 00:01:49, last status change time: 00:01:15
  Signaling protocol: LDP, peer 11.11.11.11:0 up
    MPLS VC labels: local 2530, remote 413
    Group ID: local 0, remote 0
    MTU: local 5000, remote 5000
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 19, send 19
    byte totals:   receive 1074, send 1069
    packet drops:  receive 0, send 0
```

Note Keepalives are end to end, that is, from CE toCE.

Configuring MPLS QoS

These sections provide configuration information for MPLS QoS:

Supported MPLS QoS Features

The OSMs support the following MPLS QoS features:

- OSM QoS features using MPLS EXP classification. See "Configuring QoS on the OSMs" section on page 1-2.
- MPLS EXP marking done by the OSMs when they are used with a Supervisor Engine 2. See "Configuring Class-Based Marking for MPLS (Supervisor Engine 2)" section on page 11-14.

- MPLS EXP policing and marking done by PFC3BXL when the OSMs are used with a Sup720-3BXL. For PFC3BXL policing and marking, refer to http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/swcg/qos.htm.

Note For AToM QoS features, see "How to Configure QoS with AToM" section on page 11-79.

Understanding the MPLS Experimental Field

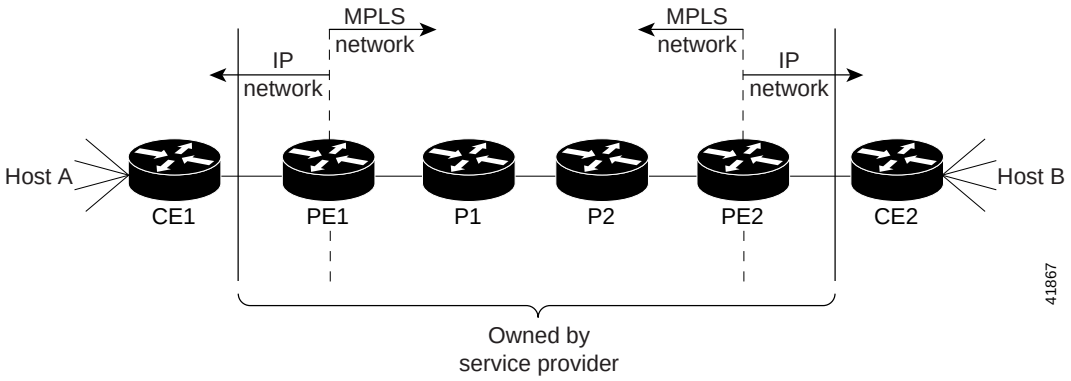
Setting the MPLS experimental field value satisfies the requirement of service providers that do not want the value of the IP precedence field modified within IP packets transported through their networks.

By choosing different values for the MPLS experimental field, you can mark packets so that packets have the priority that they require during periods of congestion.

By default, the IP precedence value is copied into the MPLS experimental field during imposition.You can mark the MPLS EXP bits with a PFC3BXL policy.

Figure 11-1 shows a service provider's MPLS network that connects two sites of a customer's network.

Figure 1-1 MPLS Network Connecting Two Sites of a Customer's IP Network



Configuring Class-Based Marking for MPLS (Supervisor Engine 2)

To configure Class-based Marking for MPLS (Supervisor Engine 2), perform the tasks described in the following sections:

- Configuring a Class Map to Classify MPLS Packets, page 11-15
- Configuring a Policy Map to Set the MPLS Experimental Field, page 11-15.

Note Class-based marking for MPLS (supervisor engine 2) is supported only on the P-facing interface of the ingress PE.

Configuring a Class Map to Classify MPLS Packets

To configure a class map, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# class-map` *class-name*	Specifies the class map to which packets will be matched.
Step 2	`Router(config-cmap)# match mpls experimental` *value*	Specifies the packet characteristics that will be matched to the class.
Step 3	`Router(config-cmap)# exit`	Exits class-map configuration mode.

This example shows that all packets that contain MPLS experimental value 4 are matched by the traffic class exp4:

```
Router(config)# class-map exp4
Router(config-cmap)# match mpls experimental 4
Router(config-cmap)# exit
```

Configuring a Policy Map to Set the MPLS Experimental Field

To configure a policy map, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map` *policy-name*	Creates a policy map that can be attached to one or more interfaces to specify a service policy.
Step 2	`Router(config-pmap)# class` *class-name*	Specifies the name of the class map previously designated in the **class-map** command.
Step 3	`Router(config-pmap-c)# set mpls experimental` *value*[1]	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Step 4	`Router(config-pmap-c)# exit`	Exits policy-map configuration mode.

1. You can also configure additional supported features, such as shaping.

This example shows that the value in the MPLS experimental field of each packet that is matched by the class-map exp4 is set to 5:

```
Router(config)# policy-map set_experimental_5
Router(config-pmap)# class exp4
Router(config-pmap-c)# set mpls experimental 5
Router(config-pmap-c)# exit
Router(config-pmap)# exit
```

Attaching the Service Policy

To attach the service policy to an interface, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface name`	Designates the output interface.
Step 2	`Router(config-if)# service-policy {input | output} policy-name`	Attaches the specified policy map to the interface.
Step 3	`Router(config-if)# exit`	Exits interface configuration mode.

This example shows that the service policy set_experimental_5 is attached to an POS output interface:

```
Router(config)# interface POS6/1
Router(config-if)# service-policy output set_experimental_5
Router(config-if)# exit
```

Verifying QoS Operation

To verify the operation of MPLS QoS, perform this task:

Command	Purpose
`Router# show policy-map interface [interface-name]`	Displays detailed information about QoS.

Configuration Examples

Sample configurations provided in this section can be applied to either OSMs or FlexWAN modules supported on the Cisco 7600 series routers.

Ingress PE Router Configuration

In the following example, IP packets with IP precedence 1 entering an MPLS network are shaped to 2000000 bits per second and set to MPLS experimental field 5. When IP packets with IP precedence 0 enter the MPLS network, they are shaped to 3000000 bits per second and set to MPLS experimental field 3. When IP packets with any other IP precedence value enter the MPLS network, they are shaped to 5000000 bits per second.

Step 1 Define two traffic classes:

```
Router(config)# class-map gold
Router(config-cmap)# match mpls experimental 1
Router(config-cmap)# exit
Router(config)# class-map silver
Router(config-cmap)# match mpls experimental 0
Router(config-cmap)# exit
```

Note Traffic classes should be defined to match on MPLS experimental values instead of IP precedence values.

Step 2 Define a policy to take different actions on different traffic classes:

```
Router(config)# policy-map policy1
Router(config-pmap)# class gold
Router(config-pmap-c)# set mpls experimental 5
Router(config-pmap-c)# shape average 2000000
Router(config-pmap-c)# exit
Router(config-pmap)# class silver
Router(config-pmap-c)# set mpls experimental 3
Router(config-pmap-c)# shape average 3000000
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 5000000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
```

Step 3 Apply the policy to the output interface of a PE router:

```
Router(config)# interface GE-WAN7/1
Router(config-if)# service-policy output policy1
```

Step 4 Verify the QoS configuration:

```
Router# show policy-map interface POS6/2

 POS6/2

  service-policy output:policy1

    class-map:gold (match-all)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      match:mpls experimental  1
      queue size 0, queue limit 500
      packets output 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0
      set:
       mpls experimental 5
      shape:cir 2000000,  Bc 8000,  Be 8000
        output bytes 0, shape rate 0 bps

    class-map:silver (match-all)
      9521 packets, 9425790 bytes
      30 second offered rate 3681000 bps, drop rate 1505000 bps
      match:mpls experimental  0
      queue size 0, queue limit 128
      packets output 2845, packet drops 6676
      tail/random drops 6676, no buffer drops 0, other drops 0
      set:
       mpls experimental 3
      shape:cir 3000000,  Bc 12000,  Be 12000
        output bytes 2816550, shape rate 642000 bps

    class-map:class-default (match-any)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      match:any
        0 packets, 0 bytes
        30 second rate 0 bps
      queue size 0, queue limit 128
      packets output 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0
      shape:cir 5000000,  Bc 20000,  Be 20000
        output bytes 0, shape rate 0 bps
Router#
```

Configuring MPLS VPN

These sections describe how to configure MPLS VPN:

MPLS VPN Support on OSMs

MPLS VPN is supported on the following OSMs:

- OC-3 POS:
  - OSM-4OC3-POS-SI
  - OSM-8OC3-POS-SI, SL
  - OSM-16OC3-POS-SI, SL
  - OSM-4OC3-POS-SI+
  - OSM-8OC3-POS-SI+, SL+
- OC-12 POS:
  - OSM-2OC12-POS-MM, SI, SL
  - OSM-4OC12-POS-MM, SI, SL
  - OSM-2OC12-POS-MM+, SI+, SL+
  - OSM-4OC12-POS-MM+, SI+, SL+
- OC-12 ATM:
  - OSM-2OC12-ATM-MM
  - OSM-2OC12-ATM-SI
  - OSM-2OC12-ATM-MM+
  - OSM-2OC12-ATM-SI+
- OC-48 POS:
  - OSM-1OC48-POS-SS, SI, SL
  - OSM-1OC48-POS-SS+, SI+, SL+
- Channelized:
  - OSM-1CHOC12/T3-SI
  - OSM-1CHOC12/T1-SI
  - OSM-12CT3/T1
- OC-48 POS/DPT[1]:
  - OSM-2OC48/1DPT-SS, SI, SL
- Gigabit Ethernet:
  - OSM-4GE-WAN-GBIC
  - OSM-2+4GE-WAN+
- WS-X6182-2PA FlexWAN
- WS-X6582-2PA Enhanced FlexWAN

1. MPLS/VPN is supported in POS mode only on the 2-port OC48/1DPT OSMs.

MPLS VPN Limitations and Restrictions

The following MPLS VPN limitations apply:

- With supervisor engine 2-based systems, load sharing is supported on PE in ip2tag and tag2ip paths; load balancing in tag2tag paths is not supported without a unique configuration ("MPLS Per-Label Load Balancing" section on page 11-21). With SUP720-3BXL-based systems, load sharing is supported.
- With supervisor engine 2-based systems, MTU checking and fragmentation is not supported. With SUP720-3BXL-based systems, MTU checking and fragmentation is supported.
- For supervisor engine 2-based systems, a total of 511 VPN routing/forwarding instance routes (VRFs) are supported per system if OSMs are non-enhanced.
- For supervisor engine 2-based systems, a total of 1000 VRFs per chassis are supported if all OSMs are enhanced.
- For SUP720-3BXL-based systems, a total of 1000 VRFs per chassis are supported with enhanced OSMs; using a non-enhanced OSM causes the system to default to 511 VRFs.
- For a supervisor engine 2 or a SUP720-3BXL system, a total of 1000 VRFs per chassis with Flexwan modules only.
- With supervisor engine 2, MPLS Provider (P) functionality is not supported on Ethernet interfaces that also support Layer 2 switching. The only way to support P functionality on these interfaces is to create a trunk from a Gigabit Ethernet interface on, for example, a WS-6516-GBIC module to an interface on the OSM-4GE-WAN module that is configured to allow P switching. The interface on the WS-6516-GBIC module should be placed in trunking mode, and appropriate subinterfaces should be created on the OSM-4GE-WAN module interface. With SUP720-3BXL-based systems, MPLS Provider (P) functionality is supported.

MPLS VPN Memory Requirements and Recommendations

When a Cisco 7600 series router or a Catalyst 6500 series switch functions as a PE router in an MPLS VPN environment, the memory requirements that are listed in Table 11-1apply:

Table 1-1 MPLS VPN Memory Requirements and Recommendations

MSFC2 Memory Configuration	**Maximum Number of Internet Routes, eBGP sessions, and VPNv4 routes**
MSFC2 with 512 MB	100,000 Internet routes, 750 eBGP sessions, and 100,000 VPNv4 routes
Supervisor Engine 2 Memory Configuration	**Maximum Number of Internet Routes, eBGP sessions, and VPNv4 routes**
Supervisor Engine 2 with 256 MB	100,000 Internet routes, 750 eBGP sessions, and 175,000 VPNv4 routes
OSM Memory Configuration	**Maximum Number of Internet Routes, eBGP sessions, and VPNv4 routes**
OSM with 256 MB	100,000 Internet routes, 750 eBGP sessions, and 175,000 VPNv4 routes
FlexWAN Memory Configuration	**Maximum Number of Internet Routes, eBGP sessions, and VPNv4 routes**
FlexWAN with 2x128 MB	100,000 Internet routes, 750 eBGP sessions, and 100, 000 VPNv4 routes

If the number of Internet routes, eBGP sessions, and VPNv4 routes exceed those listed in Table 11-1, upgrade to the next memory option. If you have a FlexWAN module installed in the system, the number of Internet routes, eBGP sessions, and VPNv4 routes in the configuration file must not exceed the requirement listed in the table for FlexWAN.

MPLS Per-Label Load Balancing

Note MPLS Per-Label Load Balancing is supported on supervisor engine 2-based systems.

When the Cisco 7600 router is configured as a P router, you can ensure traffic is distributed among equal cost paths by using the **mpls load-balance per-label** command to enable or disable the load balancing for tag-to-tag traffic.

When enabled, MPLS per-label load balancing ensures that traffic is balanced based on the incoming labels (per prefix) among MPLS interfaces; each interface supports an equal number of incoming labels.

```
mpls load-balance per-label
[no] mpls load-balance per-label
```

The default is disabled.

Use the no form of the command to return to the default setting.

This example shows how to enable load balancing for tag-to-tag traffic:

```
Router(config)# mpls load-balance per-label
Router(config)#
```

Note The **mpls load-balance per-label** command is only available for supervisor engine 2-based systems.

You can use the **show mpls ttfib** command to view the incoming label (indicated by an asterisk*) that is included in the load balancer. The following shows the output of the **show mpls ttfib** command:

```
Router# show mpls ttfib
Local  Outgoing    Packets Tag          LTL   Dest.   Destination    Outgoing
Tag    Tag or VC   Switched             Index Vlanid  Mac Address    Interface
4116   21          0                    0xE0  1020    0000.0400.0000 PO4/1*
       34          0                    0x132 1019    00d0.040d.380a GE5/3
       45          0                    0xE3  4031    0000.0430.0000 PO4/4
4117   16         0                    0x132 1019    00d0.040d.380a GE5/3*
       17          0                    0xE0  1020    0000.0400.0000 PO4/1
       18          0                    0xE3  4031    0000.0430.0000 PO4/4
4118   21          0                    0xE0  1020    0000.0400.0000 PO4/1*
       56          0                    0xE3  4031    0000.0430.0000 PO4/4
4119   35          0                    0xE3  4031    0000.0430.0000 PO4/4*
       47          0                    0xE0  1020    0000.0400.0000 PO4/1
```

Note The SUP720-3BXL handles MPLS labeled packets without commands. If the packet has three labels or less and the underlying packet is IPv4 then the SUP720-3BXL uses the source and destination IPv4 address. If the underlying packet is not IPv4 or more then three labels are present, then the SUP720-3BXL parses down as deep as the fifth or lowest label and uses it for hashing.

For information on configuring MPLS VPN, refer to the *MPLS Virtual Private Networks* feature module at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t7/vpn_en.htm.

Configuring MPLS VPN QoS

The OSMs support the following MPLS VPN QoS features:

- OSM QoS features using MPLS EXP classification. See "Configuring QoS on the OSMs" section on page 1-2.
- MPLS EXP marking done by the OSMs when they are used with a Supervisor Engine 2. See "Configuring Class-Based Marking for MPLS (Supervisor Engine 2)" section on page 11-14.
- MPLS EXP policing and marking done by PFC3BXL when the OSMs are used with a Sup720-3BXL. For PFC3BXL policing and marking, refer to http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/swcg/qos.htm.

In addition to these features, for Supervisor Engine 2-based systems, MPLS VPN also supports the **set ip precedence** command on the input WAN interfaces on the OSMs.

The following restrictions apply to the support for MPLS VPN QoS on the OSMs:

- PFC2 QoS features are not supported with MPLS VPN.
- MPLS VPN QoS is supported on the VPN interfaces only.

Match IP precedence and SET IP precedence and MPLS Experimental values are supported on the input interface only.

Configuration Example

The following example shows how to configure QoS on an MPLS VPN:

```
Router# configure terminal
Router(config)# class-map match-any vpn-class
Router(config-cmap)# match ip precedence 3
Router(config-cmap)# exit
Router(config)# policy-map VPN-MARKING
Router(config-pmap)# class vpn-class
Router(config-pmap-c)# set ip precedence 5
Router(config-pmap-c)# set mpls exp 5
Router(config-pmap-c)# ^Z
Router# configure terminal
Router(config)# interface ge-WAN 5/4
Router(config-if)# service-policy input VPN-MARKING
Router(config-if)# ^Z
Router# show running-config interface g5/4
Building configuration...

Current configuration :175 bytes
!
interface GE-WAN5/4
 ip vrf forwarding TEST
 ip address 194.3.1.3 255.255.255.0
 negotiation auto
 service-policy input VPN-MARKING
 mls qos trust dscp
end
Router#
```

Any Transport over MPLS

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 routers for setting up and maintaining connections. Forwarding occurs through the use of two level labels, switching between the edge routers. 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 VPI/VCI value for the AAL5 PDU, the DLCI value for Frame Relay PDU, or the VLAN identifier for an Ethernet frame).

AToM supports the following like-to-like transport types for SUP720-3BXL-based systems and for supervisor engine 2-based systems:

- Ethernet over MPLS (VLAN mode and port mode)

Note SUP720-3BXL-based systems support both hardware-based WAN as well as OSM-, FlexWAN, or FlexWAN2-based WAN.

- ATM AAL5 over MPLS
- ATM Cell Relay over MPLS
- Frame Relay over MPLS

Note SUP720-PFC3B-based systems and SUP720-PFC3BXL-based systems require that the core-facing cards must be WAN cards (enhanced OSMs, FlexWAN and Enhanced FlexWAN modules, and Shared Port Adapter [SPA] Interface Processors [SIPs]). This applies to Ethernet over MPLS, ATM AAL5 over MPLS, ATM Cell Relay over MPLS, and Frame Relay over MPLS.

Also, the specific MPLS core-facing line card may not be supported for a specific AToM technology; view specific AToM configurations in this chapter, in the *FlexWAN and Enhanced FlexWAN Installation and Configuration Note*, and in the *Cisco 7600 Series Router SIP, SSC, and SPA Software Configuration Guide* for more details.

Restrictions for Any Transport over MPLS

The following general restrictions pertain to all transport types under AToM:

- **Sequencing**: AToM does not support detecting of out-of-order packets.
- **Address format**: Configure the LDP router ID on all PE routers to be a loopback address with a /32 mask. Otherwise, some configurations might not properly function.
- **Fragmentation and Reassembly**: Ensure that the maximum transmission unit (MTU) of all intermediate links between endpoints is sufficient to carry the largest Layer 2 packet received.
- **Control word**: You cannot use CLI to enable or disable control word. Control word is mandatory for FRoMPLS and ATM AAL5 over MPLS. Control word is optional for ATM Cell Relay over MPLS; however, because there is no CLI option at this time, it is not imposed and is not expected to be present in the disposition packets.

Ethernet over MPLS Restrictions

The following restrictions pertain to the Ethernet over MPLS feature:

- **Fragmentation and Reassembly**: Ensure that the maximum transmission unit (MTU) of all intermediate links between endpoints is sufficient to carry the largest Layer 2 packet received.
- **Packet Format**: EoMPLS supports VLAN packets that conform to the IEEE's 802.1Q standard. The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames.
- **Preserving 802.1 P bits and IP precedence bits**: If QoS is disabled globally, both the 802.1p and IP precedence bits are preserved. When the QoS is enabled on a Layer 2 port, either 802.1q P bits or IP precedence bits can be preserved with the trusted configuration. However, by default the unpreserved bits are overwritten by the value of preserved bits. For instance, if you preserve the P bits, the IP precedence bits are overwritten with the value of the P bits. PFC3BXL provides a new command that allows you to trust the P bits while preserving the IP precedence bits. To preserve the IP precedence bits, use the **no mls qos rewrite ip dscp** command.

Note The **no mls qos rewrite ip dscp** command is not compatible with the MPLS and MPLS VPN features. See http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/swcg/qos.htm.

- **Private VLANs**: EoMPLS is not supported with private VLANs.
- **Layer 2 Connections**: The following restrictions apply to using Layer 2 connection with Ethernet over MPLS:
  - You cannot have a direct Layer 2 connection between PEs with Ethernet over MPLS.
  - You cannot have more than one Layer 2 connection between routers if those routers are configured to transport Ethernet VLAN packets over the MPLS backbone. Adding a second Layer 2 connection causes the spanning tree state to constantly toggle if you disable spanning tree on the peer router.
- **Ethernet over MPLS and Trunks**: The following restrictions apply to using trunks with Ethernet over MPLS. For more information, see the Cisco 7600 Series Router software documentation.
  - **Spanning Tree**: To support Ethernet spanning tree bridge protocol data units (BPDUs) across an EoMPLS cloud, you must disable the supervisor engine spanning tree for the Ethernet over MPLS VLAN. This ensures that the EoMPLS VLANs are carried only on the trunk to the customer switch. Otherwise, the BPDUs are directed to the supervisor engine and not to the EoMPLS cloud.
  - **Native VLAN**: The native VLAN of a trunk must not be configured as an EoMPLS VLAN.
- **Layer 2 Protocol Tunneling**: With PFC3BXL-based systems, there is a configuration choice for user to decide which specific protocols (for example, CDP, VTP, BPDUs). get tunneled across the MPLS cloud and which ones terminate locally. This is supported in software switching path.
- ISL encapsulation is not supported for the interface that receives EoMPLS packets.
- Unique VLANs are required across interfaces. You cannot use the same VLAN ID on different interfaces.
- EoMPLS tunnel destination route in routing and CEF table must be with a /32 mask to insure that there is an LSP from PE to PE.
- For a particular EoMPLS connection, both the ingress EoMPLS interface on the ingress PE and the egress EoMPLS interface on the egress PE have to be sub-interfaces with dot1Q encapsulation or neither is a sub-interface.

- 802.1Q in 802.1Q over EoMPLS is supported if outgoing interface connecting to MPLS network is a port on an Layer 2 card.
- Shaping of EoMPLS traffic is not supported if egress interface connecting to MPLS network is Layer 2 card.
- EoMPLS based on PFC3BXL does not perform any Layer 2 look up to determine if the destination MAC address resides on the local or remote segment and does not perform any Layer 2 address learning (as traditional LAN bridging does). This functionality (local switching or hair pinning) is available only when using OSM/FlexWAN-based modules as uplinks.

ATM AAL5 over MPLS Restrictions

The following restrictionapplies to the ATM AAL5 over MPLS feature.

- **Fragmentation and Reassembly**: Ensure that the maximum transmission unit (MTU) of all intermediate links between endpoints is sufficient to carry the largest Layer 2 packet received.
- Both CE-facing and core-facing cards must be WAN cards (enhanced OSMs, FLexWAN and Enhanced FlexWAN modules) and not LAN cards.

ATM Cell Relay over MPLS Restrictions

The following restrictions pertain to the ATM Cell Relay over MPLS feature:

- **PVC configuration**: You can configure ATM Cell Relay over MPLS on PVCs only.
- Single cell relay over MPLS (SCRoMPLS): In this release, each MPLS packet contains one ATM cell. In other words, each ATM cell is transported as a single packet.

Note Cell packing is not supported.

- For SCRoMPLS, if one end of the VC has a WS-X6182-2PA FlexWAN or a WS-X6582-2PA Enhanced FlexWAN with an ATM port adapter (PA) interface, then the VPIs/VCIs must match.

Note For SCRoMPLS, if there is a WS-X6182-2PA FlexWAN or a WS-X6582-2PA Enhanced FlexWAN with an ATM port adapter (PA) interface on one end of the VC and the VPIs/VCIs do not match then a VC does come up but does not switch traffic.

- **Control word**: The use of the control word is not supported.
- **VCC mode**: ATM Cell Relay over MPLS supports only virtual channel connection (VCC) mode.
- **Fragmentation and Reassembly**: Ensure that the maximum transmission unit (MTU) of all intermediate links between endpoints is sufficient to carry the largest Layer 2 packet received.
- Both CE-facing and core-facing cards must be WAN cards (enhanced OSMs, FLexWAN and Enhanced FlexWAN modules) and not LAN cards.

Frame Relay over MPLS Restrictions

The Frame Relay over MPLS feature has the following restriction:

- **BECN, FECN, and DE Bits**: OSMs do not update backward explicit congestion notification (BECN), forward explicit congestion notification (FECN), and discard eligibility (DE) bit counters; the bit counters remain at zero.
- Port-based mode (many-to-one): All the DLCIs coming in on a given interface/port are mapped to one MPLS LSP. This mode is not supported.
- FRF.12 is not supported on PE-CE link.
- LFI/MLPPP over FR DLCI that is transported over MPLS LSPs is not supported.
- Mapping DE bit on to MPLS EXP based on configured EXP value is not supported.
- Both CE-facing and core-facing cards must be WAN cards (enhanced OSMs, FLexWAN and Enhanced FlexWAN modules) and not LAN cards.

Information About Any Transport over MPLS

To configure AToM, you must understand the following concepts:

- How AToM Transports Layer 2 Packets, page 11-26
- Compatibility with Previous Releases of AToM, page 11-27
- Benefits of AToM, page 11-27

How AToM Transports Layer 2 Packets

AToM encapsulates Layer 2 frames at the ingress PE and sends them to a corresponding PE at the other end of a pseudowire, which is a connection between the two PE routers. The egress PE removes the encapsulation and sends out the Layer 2 frame.

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You set up the connection, called a pseudowire, between the routers. You specify the following information on each PE router:

- The type of Layer 2 data that will be transported across the pseudowire, such as Ethernet, Frame Relay, or ATM
- The IP address of the loopback interface of the peer PE router, which enables the PE routers to communicate
- A VC ID that uniquely identifies the pseudowire

The following example shows the basic configuration steps on a PE router that enable the transport of Layer 2 packets. Each transport type (EoMPLS, ATMoMPLS, FRoMPLS) has slightly different steps.

First define the interface or subinterface on the PE router.

```
Router# interface interface-type interface-number
```

Then specify the encapsulation type for the interface, such as dot1q.

```
Router(config-if)# encapsulation encapsulation-type
```

The last step does the following:

- Makes a connection to the peer PE router by specifying the LDP router ID of the peer PE router.
- Identifies a unique identifier that is shared between the two PE routers. The *vcid* is a 32-bit identifier.

  The combination of the peer-router-id and the VC ID must be a unique combination on the router. Two circuits cannot use the same combination of peer-router-id and VC ID.

- Specifies the tunneling method used to encapsulate data in the pseudowire. For AToM, the tunneling method used to encapsulate data is **mpls.**

```
Router(config-if)# xconnect peer-router-id vcid encapsulation mpls
```

Note The **xconnect** command as shown above is only applicable for some transports and not for FRoMPLS.

Compatibility with Previous Releases of AToM

In previous releases of AToM, the command used to configure AToM circuits was **mpls l2 transport route.** This command has been replaced with the **xconnect** command. You can use the **xconnect** command to configure FRoMPLS and EoMPLS circuits.

Note You must use the **mpls l2 transport route** command to configure ATM AAL5 over MPLS and ATM Cell Relay over MPLS circuits.

Benefits of AToM

The following list explains some of the benefits of enabling Layer 2 packets to be sent in the MPLS network:

- The AToM product set accommodates many types of Layer 2 packets, including Ethernet and Frame Relay, across multiple Cisco router platforms, including the Cisco 7600 series routers. This enables the service provider to transport all types of traffic over the backbone and accommodate all types of customers.
- AToM adheres to the standards developed for transporting Layer 2 packets over MPLS. (See the "Ethernet over MPLS" section on page 11-28 for the specific standards that AToM follows.) This benefits the service provider who wants to incorporate industry-standard methodologies in the network. Other Layer 2 solutions are proprietary, which can limit the service provider's ability to expand the network and can force the service provider to use only one vendor's equipment.
- Upgrading to AToM is transparent to the customer. Because the service provider network is separate from the customer network, the service provider can upgrade to AToM without disruption of service to the customer. The customers assume that they are using a traditional Layer 2 backbone.

Prerequisites

Before configuring AToM, ensure that the network is configured as follows:

- Configure IP routing in the core so that the PE routers can reach each other via IP.
- Configure MPLS in the core so that a label switched path (LSP) exists between the PE routers.

AToM and QoS

MPLS AToM uses the three experimental bits in a label to determine the queue of packets. You statically set the experimental bits in both the VC label and the LSP tunnel label, because the LSP tunnel label might be removed at the penultimate router. See "How to Configure QoS with AToM" section on page 11-79 and "HQoS for EoMPLS Virtual Circuits" section on page 11-90 for more information.

Ethernet over MPLS

Ethernet over MPLS works by encapsulating Ethernet PDUs in MPLS packets and forwarding them across the MPLS network. Each PDU is transported as a single packet. There are various ways to configure Ethernet over MPLS:

- VLAN mode—transports Ethernet traffic from a source 802.1Q VLAN to a destination 802.1Q VLAN through a single VC over an MPLS network.
- Port mode—allows all traffic on a port to share a single VC across an MPLS network.

There are two methods to configure EoMPLS on a SUP720-3BXL-based system and one method for a supervisor engine 2-based system.

SUP720-3BXL-Based EoMPLS

With SUP720-3BXL-based systems the supervisor engine 720 supports the MPLS functionality. The supervisor engine 720 can receive Layer 2 traffic, impose labels, and switch the frames into the MPLS core without using an OSM or FlexWAN module.

You can also equip a SUP720-3BXL-based system with an OSM or a Flexwan module facing the core of MPLS network. In this case, you can use either OSM/FlexWAN-based configuration or the SUP720-3BXL-based configuration.

Note A system can have both an OSM/FlexWAN-based configuration and a SUP720-3BXL-based configuration enabled at the same time. Cisco supports this configuration but does not recommend it. Unless the uplinks to the MPLS core are through OSM/FlexWAN-enabled interfaces then OSM/FlexWAN-based EoMPLS connections are not active; this causes packets for OSM/FlexWAN-based EoMPLS arriving on non-WAN interfaces to be dropped.

Supervisor Engine 2-Based EoMPLS

You must equip a supervisor engine 2-based system with an OSM or a FlexWAN module facing the core of MPLS network.

Supported OSMs

Table 11-2 lists the POS/SDH OSMs that support EoMPLS.

Table 1-2 POS/SDH OSMs That Support EoMPLS

OC-3c OSMs	OC-12c OSMs	OC-48c OSMs	Gigabit Ethernet OSMs
OSM-4OC3-POS-SI	OSM-2OC12-POS-MM	OSM-1OC48-POS-SS	OSM-4GE-WAN-GBIC
OSM-4OC3-POS-SI+	OSM-2OC12-POS-SI	OSM-1OC48-POS-SI	OSM-2+4GE-WAN+
OSM-8OC3-POS-SI	OSM-2OC12-POS-SL	OSM-1OC48-POS-SL	
OSM-8OC3-POS-SL	OSM-2OC12-POS-MM+	OSM-1OC48-POS-SS+	
OSM-8OC3-POS-SI+	OSM-2OC12-POS-SI+	OSM-1OC48-POS-SI+	
OSM-8OC3-POS-SL+	OSM-4OC12-POS-MM	OSM-1OC48-POS-SL+	
OSM-16OC3-POS-SI	OSM-4OC12-POS-SI		
OSM-16OC3-POS-SL	OSM-4OC12-POS-SL		
	OSM-4OC12-POS-MM+		
	OSM-4OC12-POS-SI+		

Note Though the OSM-2OC12-POS-SI+ card contains 2 POS ports and 4 GigE ports, the GigE ports do not support mqc queuing or shaping.

Configuring EoMPLS VLAN Mode for Supervisor Engine 2 or OSM-Based System

To configure MPLS to transport Layer 2 VLAN packets between two endpoints in an OSM-based system, perform the following steps on the provider edge (PE) routers.

Note When OSPF is used as the IGP, all loopback addresses on PE routers must be configured with 32-bit masks to ensure proper operation of MPLS forwarding between PE routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan**
4. **interface gigabitEthernet**
5. **switchport**
6. **switchport trunk encapsulation dot1q**
7. **switchport trunk allowed** *vlan list*
8. **switchport mode trunk**
9. **exit**
10. **interface vlan**
11. **mpls l2transport route**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vlan** {*vlan-id* | *vlan-range*} **Example:** `Router (config)# vlan 2-3`	Enter VLAN ID or range.
Step 4	**interface gigabitEthernet** **Example:** `Router(config)# interface gigabitEthernet`	Specifies the Layer 2 interface and enters interface configuration mode.
Step 5	**switchport** **Example:** `Router(config-if)# switchport`	Configures the port for switching.
Step 6	**switchport trunk encapsulation dot1** **Example:** `Router(config-if)# switchport trunk encapsulation dot1`	Set the trunk characteristics when the interface is in trunking mode.
Step 7	**switchport trunk allowed** *vlan list* **Example:** `Router(config-if)# switchport trunk allowed vlan list`	Changes the allowed list for the specified VLANs.
Step 8	**switchport mode trunk** **Example:** `Router(config-if)# switchport mode trunk`	Specifies a trunking VLAN Layer 2 interface.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

	Command	Purpose
Step 10	**interface vlan** *vlanid* **Example:** `Router(config)# interface vlan vlanid`	Creates a unique VLAN ID number.
Step 11	**mpls l2transport route** *destination vc-id* **Example:** `Router(config-if)# mpls l2transport route 9.9.11.11 3`	Specifies the VC to use to transport the Layer 2 VLAN packets. The argument *destination* specifies the loopback address of the remote router. The argument *vc-id* is a value you supply. It must be unique for each VC. The VC ID is used to connect the endpoints of the VC.

The following configuration shows a mode trunk configuration.

CE1 Configuration

```
!
interface GigabitEthernet1/0
no ip address
no ip mroute-cache
negotiation auto
no cdp enable
no shut
!
interface GigabitEthernet1/0.2
encapsulation dot1Q 2
ip address 180.8.0.1 255.255.0.0
no cdp enable
no shut
!
interface GigabitEthernet1/0.3
encapsulation dot1Q 3
ip address 180.9.0.1 255.255.0.0
no cdp enable
no shut
!
```

CE2 Configuration

```
!
interface GigabitEthernet4/0
no ip address
no ip directed-broadcast
negotiation auto
tag-switching ip
no cdp enable
no shut
!
interface GigabitEthernet4/0.2
encapsulation dot1Q 2
ip address 180.8.0.2 255.255.0.0
no ip directed-broadcast
no cdp enable
no shut
!
interface GigabitEthernet4/0.3
encapsulation dot1Q 3
ip address 180.9.0.2 255.255.0.0
no ip directed-broadcast
```

```
no cdp enable
no shut
!
```

PE1 Configuration

```
!
vlan 2-3
!
interface GigabitEthernet1/4
no ip address
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 2-3
switchport mode trunk
no shut
!
interface Vlan2
no ip address
no ip mroute-cache
mpls l2transport route 11.11.11.11 2
no shut
!
interface Vlan3
no ip address
no ip mroute-cache
mpls l2transport route 11.11.11.11 3
no shut
!
```

PE2 Configuration

```
!
vlan 2-3
!
interface GigabitEthernet7/4
no ip address
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 2-3
switchport mode trunk
no shut
!
interface Vlan2
no ip address
no ip mroute-cache
mpls l2transport route 13.13.13.13 2
no shut
!
interface Vlan3
no ip address
no ip mroute-cache
mpls l2transport route 13.13.13.13 3
no shut
!
```

Configuring EoMPLS VLAN Mode for SUP720-3BXL-Based System

To configure MPLS to transport Layer 2 VLAN packets between two endpoints in a supervisor engine 720-based system, perform the following steps on the provider edge (PE) routers.

Note You must configure Ethernet over MPLS (VLAN mode) on the subinterfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vtp mode transparent**
4. **interface gigabitethernet***slot/interface.subinterface*
5. **encapsulation dot1q** *vlan-id*
6. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vtp mode transparent** **Example:** `Router(config)#vtp mode transparent`	Disables VLAN Trunking Protocol (VTP).
Step 4	**interface gigabitethernet***slot/interface.subinterface* **Example:** `Router(config)# interface gigabitethernet4/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. Make sure the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-subif)# encapsulation dot1q 100`	Enables the subinterface to accept 802.1Q VLAN packets. The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet. All other subinterfaces and backbone routers do not.
Step 6	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-subif)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.

Recall that you can use either OSM/FlexWAN-based configuration or the SUP720-3BXL-based configuration. The following configuration shows the use of both with dot1Q tunneling on the supervisor engine 2.

Note The IP address is configured on subinterfaces of the CE devices.

CE1 Configuration

```
!
interface GigabitEthernet1/0
no ip address
no ip mroute-cache
negotiation auto
no cdp enable
no shut
!
interface GigabitEthernet1/0.2
encapsulation dot1Q 2
ip address 180.8.0.1 255.255.0.0
no cdp enable
no shut
!
interface GigabitEthernet1/0.3
encapsulation dot1Q 3
ip address 180.9.0.1 255.255.0.0
no cdp enable
no shut
!
```

CE2 Configuration

```
!
interface GigabitEthernet4/0
no ip address
no ip directed-broadcast
negotiation auto
tag-switching ip
no cdp enable
no shut
!
interface GigabitEthernet4/0.2
encapsulation dot1Q 2
ip address 180.8.0.2 255.255.0.0
no ip directed-broadcast
no cdp enable
no shut
!
interface GigabitEthernet4/0.3
encapsulation dot1Q 3
ip address 180.9.0.2 255.255.0.0
no ip directed-broadcast
no cdp enable
no shut
!
```

PE1 Configuration (supervisor engine 2)

```
!
vlan 2-3
!
interface GigabitEthernet1/4
no ip address
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 2-3
switchport mode trunk
no shut
!
```

```
interface Vlan2
no ip address
no ip mroute-cache
mpls l2transport route 11.11.11.11 2
no shut
!
interface Vlan3
no ip address
no ip mroute-cache
mpls l2transport route 11.11.11.11 3
no shut
!
```

PE2 Configuration (supervisor engine 720)

```
!
vtp mode transparent
!
interface GigabitEthernet7/4
no ip address
no shut
!
interface GigabitEthernet7/4.1
encapsulation dot1Q 2
xconnect 13.13.13.13 2 encapsulation mpls
no shut
!
interface GigabitEthernet7/4.2
encapsulation dot1Q 3
xconnect 13.13.13.13 3 encapsulation mpls
no shut
!
```

Ethernet over MPLS VLAN Mode Configuration Guidelines

When configuring Ethernet over MPLS in VLAN mode, use the following guidelines:

- The AToM control word is supported. However, if the peer PE does not support a control word, the control word is disabled. This negotiation is done by LDP label binding.
- Ethernet packets with hardware level cyclic redundancy check (CRC) errors, framing errors, and runt packets are discarded on input.

Verifying the Configuration

To verify and display the configuration of Layer 2 VLAN transport over MPLS tunnels, perform the following steps:

Step 1 To display a brief summary of IP status and configuration for all interfaces, issue the **show vlan brief** command. If the interface can provide two-way communication, the Protocol field is marked “up.” If the interface hardware is usable, the Status field is marked “up.”

```
Router# show vlan brief
osr1#sh vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------
1    default                          active
2    VLAN0002                         active
3    VLAN0003                         active
```

```
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

Step 2 To make sure the PE router endpoints have discovered each other, issue the **show mpls ldp discovery** command. When an PE router receives an LDP Hello message from another PE router, it considers that router and the specified label space to be "discovered."

```
Router# show mpls ldp discovery
osr1#show mpls ldp discovery
 Local LDP Identifier:
    13.13.13.13:0
    Discovery Sources:
    Interfaces:
        GE-WAN3/3 (ldp): xmit/recv
            LDP Id: 12.12.12.12:0
    Targeted Hellos:
        13.13.13.13 -> 11.11.11.11 (ldp): active/passive, xmit/recv
            LDP Id: 11.11.11.11:0
```

Step 3 To make sure the label distribution session has been established, issue the **show mpls ldp neighbor** command. The third line of the output shows that the state of the LDP session is operational and shows that messages are being sent and received.

```
Router# show mpls ldp neighbor
osr1#show mpls ldp neighbor
    Peer LDP Ident: 12.12.12.12:0; Local LDP Ident 13.13.13.13:0
        TCP connection: 12.12.12.12.646 - 13.13.13.13.11010
        State: Oper; Msgs sent/rcvd: 1649/1640; Downstream
        Up time: 23:42:45
        LDP discovery sources:
          GE-WAN3/3, Src IP addr: 34.0.0.2
        Addresses bound to peer LDP Ident:
          23.2.1.14       37.0.0.2        12.12.12.12     34.0.0.2
          99.0.0.1
    Peer LDP Ident: 11.11.11.11:0; Local LDP Ident 13.13.13.13:0
        TCP connection: 11.11.11.11.646 - 13.13.13.13.11013
        State: Oper; Msgs sent/rcvd: 1650/1653; Downstream
        Up time: 23:42:29
        LDP discovery sources:
          Targeted Hello 13.13.13.13 -> 11.11.11.11, active, passive
        Addresses bound to peer LDP Ident:
          11.11.11.11     37.0.0.1        23.2.1.13
```

Step 4 To make sure the label forwarding table is built correctly, issue the **show mpls forwarding-table** command. The output shows the following data:

- Local tag—Label assigned by this router.
- Outgoing tag or VC—Label assigned by next hop.
- Prefix or Tunnel Id—Address or tunnel to which packets with this label are going.
- Bytes tag switched— Number of bytes switched out with this incoming label.
- Outgoing interface—Interface through which packets with this label are sent.
- Next Hop—IP address of neighbor that assigned the outgoing label.

```
Router# show mpls forwarding-table
osr1#show mpls forwarding-table
Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Untagged    223.255.254.254/32   \
```

```
                                        0          Gi2/1      23.2.0.1
20      Untagged    l2ckt(2)            133093     Vl2        point2point
21      Untagged    l2ckt(3)            185497     Vl3        point2point
24      Pop tag     37.0.0.0/8          0          GE3/3      34.0.0.2
25      17          11.11.11.11/32      0          GE3/3      34.0.0.2
26      Pop tag     12.12.12.12/32      0          GE3/3      34.0.0.2
osr1#
```

Step 5 To view the state of the currently routed VCs issue the **show mpls l2transport vc** command.

```
Router# show mpls l2transport vc
osr1#show mpls l2transport vc

Local intf     Local circuit         Dest address    VC ID      Status
-------------  --------------------- --------------- ---------- ----------
Vl2            Eth VLAN 2            11.11.11.11     2          UP
Vl3            Eth VLAN 3            11.11.11.11     3          UP
```

Step 6 Add the keyword **detail** to see detailed information about each VC.

```
Router# show mpls l2transport vc detail
osr1#show mpls l2transport vc detail
Local interface: Vl2 up, line protocol up, Eth VLAN 2 up
  Destination address: 11.11.11.11, VC ID: 2, VC status: up
    Tunnel label: 17, next hop 34.0.0.2
    Output interface: GE3/3, imposed label stack {17 18}
  Create time: 01:24:44, last status change time: 00:10:55
  Signaling protocol: LDP, peer 11.11.11.11:0 up
    MPLS VC labels: local 20, remote 18
    Group ID: local 71, remote 89
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 1009, send 1019
    byte totals:   receive 133093, send 138089
    packet drops:  receive 0, send 0

Local interface: Vl3 up, line protocol up, Eth VLAN 3 up
  Destination address: 11.11.11.11, VC ID: 3, VC status: up
    Tunnel label: 17, next hop 34.0.0.2
    Output interface: GE3/3, imposed label stack {17 19}
  Create time: 01:24:38, last status change time: 00:10:55
  Signaling protocol: LDP, peer 11.11.11.11:0 up
    MPLS VC labels: local 21, remote 19
    Group ID: local 72, remote 90
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 1406, send 1414
    byte totals:   receive 185497, send 191917
    packet drops:  receive 0, send 0
```

Configuring EoMPLS Port Mode for Supervisor Engine 2 or OSM-Based System

To support 802.1Q-in-802.1Q traffic and native Ethernet traffic over EoMPLS in an OSM-based system, configure port-based EoMPLS by performing these tasks:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan**
4. **vlan dot1q tag native**
5. **interface gigabitEthernet**
6. **switchport**
7. **switchport mode dot1qtunnel**
8. **switchport access vlan**
9. **exit**
10. **interface vlan**
11. **mpls l2transport route**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vlan** *{vlan-id | vlan-range}* **Example:** `Router (config)# vlan 2-3`	Enter VLAN ID or range.
Step 4	**vlan dot1q tag native** **Example:** `Router(config)# vlan dot1q tag native`	Enables dot1q tagging for all VLANs in a trunk.
Step 5	**interface gigabitEthernet** `Router(config)# interface gigabitEthernet`	Specifies the Layer 2 interface and enters interface configuration mode.
Step 6	**switchport** **Example:** `Router(config-if)# switchport`	Configures the port for switching.

	Command	Purpose
Step 7	**switchport mode dot1qtunnel** **Example:** `Router(config-if)# switchport mode dot1qtunnel`	Set the trunking mode to tunneling.
Step 8	**switchport access vlan** *vlan_id* **Example:** `Router(config-if)# switchport access vlan 7`	Configures the port to accept traffic from the specified VLAN.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 10	**interface vlan** *vlanid* **Example:** `Router(config)# interface vlan vlanid`	Creates a unique VLAN ID number.
Step 11	**mpls l2transport route** *destination vc-id* **Example:** `Router(config-if)# mpls l2transport route 11.11.11.11 2`	Specifies the VC to use to transport the Layer 2 VLAN packets. The argument *destination* specifies the loopback address of the remote router. The argument *vc-id* is a value you supply. It must be unique for each VC. The VC ID is used to connect the endpoints of the VC.

This example shows a port mode access configuration for untagged packets. It requires configuring the IP addresses on the main interface of the CE devices.

CE1 Configuration

```
!
interface GigabitEthernet1/0
ip address 180.8.0.1 255.255.0.0
no ip mroute-cache
negotiation auto
no cdp enable
no shut
!
```

CE 2 Configuration

```
!
interface GigabitEthernet4/0
ip address 180.8.0.2 255.255.0.0
no ip directed-broadcast
negotiation auto
tag-switching ip
no cdp enable
no shut
!
```

PE1 Configuration

```
!
vlan 2
!
```

```
interface GigabitEthernet1/4
no ip address
switchport
switchport access vlan 2
switchport mode access
no shut
!
interface Vlan2
no ip address
no ip mroute-cache

mpls l2transport route 11.11.11.11 2
no shut
!
```

PE2 Configuration

```
!
vlan 2
!
interface GigabitEthernet7/4
no ip address
switchport
switchport access vlan 2
switchport mode access
no shut
!
interface Vlan2
no ip address
no ip mroute-cache

mpls l2transport route 13.13.13.13 2
no shut
!
```

This configuration shows a port mode dot1Q-tunneling configuration. You must configure subinterfaces on the CE devices for this configuration. There is a specific access VLAN for the packets.

CE1 Configuration

```
!
interface GigabitEthernet1/0
no ip address
no ip mroute-cache
negotiation auto
no cdp enable
no shut
!
interface GigabitEthernet1/0.2
encapsulation dot1Q 2
ip address 180.8.0.1 255.255.0.0
no cdp enable
no shut
!
interface GigabitEthernet1/0.3
encapsulation dot1Q 3
ip address 180.9.0.1 255.255.0.0
no cdp enable
no shut
!
```

CE2 Configuration

```
!
interface GigabitEthernet4/0
no ip address
no ip directed-broadcast
negotiation auto
```

```
tag-switching ip
no cdp enable
no shut
!
interface GigabitEthernet4/0.2
encapsulation dot1Q 2
ip address 180.8.0.2 255.255.0.0
no ip directed-broadcast
no cdp enable
no shut
!
interface GigabitEthernet4/0.3
encapsulation dot1Q 3
ip address 180.9.0.2 255.255.0.0
no ip directed-broadcast
no cdp enable
no shut
!
```

PE1 Configuration

Note This configuration requires vlan dot1q tag native.

```
!
vlan 2
!
vlan dot1q tag native
!
interface GigabitEthernet1/4
no ip address
switchport
switchport access vlan 2
switchport trunk encapsulation dot1q
switchport mode dot1q-tunnel
no cdp enable
spanning-tree bpdufilter enable
no shut
!
interface Vlan2
no ip address
no ip mroute-cache

mpls l2transport route 11.11.11.11 2
no shut
!
```

PE2 Configuration

Note This configuration requires vlan dot1q tag native.

```
!
vlan 2
!
vlan dot1q tag native
!
interface GigabitEthernet7/4
no ip address
switchport
switchport access vlan 2
switchport trunk encapsulation dot1q
switchport mode dot1q-tunnel
```

```
no cdp enable
spanning-tree bpdufilter enable
no shut
!
interface Vlan2
no ip address
no ip mroute-cache

mpls l2transport route 13.13.13.13 2
no shut
!
```

Configuring EoMPLS Port Mode for SUP720-3BXL-Based System

To support 802.1Q-in-802.1Q traffic and native Ethernet traffic over EoMPLS in a supervisor engine 720-based system, configure port-based EoMPLS by performing these tasks:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet***x/x*
4. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode and enter interface configuration mode.
Step 3	**interface gigabitethernet***slot/interface* **Example:** `Router(config-if)# interface gigabitethernet4/0`	Specifies the Gigabit Ethernet interface and enters subinterface configuration mode. Make sure the interface on the adjoining CE router is on the same VLAN as this PE router.
Step 4	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-subif)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.

Note When the underlying port of the VLAN is an access port or .1q in .1q tunnel, then you must use an OSM or FlexWAN module to access the MPLS core similarly to the supervisor engine 2 configurations in the example below.

The following example provides both SUP720-3BXL and supervisor engine 2 configurations. It also provides two configurations for the CE devices: one where the IP address is configured on the main interface and another where the IP address is configured on the subinterface.

CE1 Configuration (main interface)

```
!
interface GigabitEthernet1/0
ip address 180.8.0.1 255.255.0.0
no ip mroute-cache
negotiation auto
no cdp enable
no shut
!
```

CE1 Configuration (subinterface)

```
!
interface GigabitEthernet1/0
no ip address
no ip mroute-cache
negotiation auto
no cdp enable
no shut
!
interface GigabitEthernet1/0.2
encapsulation dot1Q 2
ip address 180.8.0.1 255.255.0.0
no cdp enable
no shut
!
interface GigabitEthernet1/0.3
encapsulation dot1Q 3
ip address 180.9.0.1 255.255.0.0
no cdp enable
no shut
!
!
```

CE2 Configuration (main interface)

```
!
interface GigabitEthernet4/0
ip address 180.8.0.2 255.255.0.0
no ip directed-broadcast
negotiation auto
tag-switching ip
no cdp enable
no shut
!
```

CE2 Configuration (subinterface)

```
!
interface GigabitEthernet4/0
no ip address
no ip directed-broadcast
negotiation auto
tag-switching ip
no cdp enable
no shut
```

```
!
interface GigabitEthernet4/0.2
encapsulation dot1Q 2
ip address 180.8.0.2 255.255.0.0
no ip directed-broadcast
no cdp enable
no shut
!
interface GigabitEthernet4/0.3
encapsulation dot1Q 3
ip address 180.9.0.2 255.255.0.0
no ip directed-broadcast
no cdp enable
no shut
!
```

PE1 Configuration (supervisor engine 2)

```
!
vlan 2
!
interface GigabitEthernet1/4
 no ip address
 switchport
 switchport access vlan 2
 switchport trunk encapsulation dot1q
 switchport mode dot1q-tunnel
 no cdp enable
 spanning-tree bpdufilter enable
 no shut
!
interface Vlan2
 no ip address
 no ip mroute-cache
 mpls l2transport route 11.11.11.11 2
 no shut
!
```

PE2 Configuration (SUP720-3BXL)

```
!
interface GigabitEthernet7/4
no ip address
xconnect 13.13.13.13 2 encapsulation mpls
no shut
!
```

Ethernet over MPLS Port Mode Configuration Guidelines

When configuring Ethernet over MPLS in port mode, use the following guidelines:

- The AToM control word is supported. However, if the peer PE does not support a control word, the control word is disabled. This negotiation is done by LDP label binding.
- Ethernet packets with hardware level cyclic redundancy check (CRC) errors, framing errors, and runt packets are discarded on input.
- Port mode and Ethernet VLAN mode are mutually exclusive. If you enable a main interface for port-to-port transport, you cannot also enter commands on a subinterface.

Verifying the Configuration

To verify and display the configuration of Layer 2 VLAN transport over MPLS tunnels, perform the following steps:

Step 1 To display a brief summary of IP status and configuration for all interfaces, issue the **show vlan brief** command. If the interface can provide two-way communication, the Protocol field is marked "up." If the interface hardware is usable, the Status field is marked "up."

```
Router# show vlan brief
osr1#sh vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active
2    VLAN0002                         active    Gi1/4
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

Step 2 To make sure the PE router endpoints have discovered each other, issue the **show mpls ldp discovery** command. When an PE router receives an LDP Hello message from another PE router, it considers that router and the specified label space to be "discovered."

```
Router# show mpls ldp discovery
osr1#show mpls ldp discovery
 Local LDP Identifier:
    13.13.13.13:0
    Discovery Sources:
    Interfaces:
        GE-WAN3/3 (ldp): xmit/recv
            LDP Id: 12.12.12.12:0
    Targeted Hellos:
        13.13.13.13 -> 11.11.11.11 (ldp): active/passive, xmit/recv
            LDP Id: 11.11.11.11:0
```

Step 3 To make sure the label distribution session has been established, issue the **show mpls ldp neighbor** command. The third line of the output shows that the state of the LDP session is operational and shows that messages are being sent and received.

```
Router# show mpls ldp neighbor
osr1#show mpls ldp neighbor
    Peer LDP Ident: 12.12.12.12:0; Local LDP Ident 13.13.13.13:0
        TCP connection: 12.12.12.12.646 - 13.13.13.13.11010
        State: Oper; Msgs sent/rcvd: 1715/1706; Downstream
        Up time: 1d00h
        LDP discovery sources:
          GE-WAN3/3, Src IP addr: 34.0.0.2
        Addresses bound to peer LDP Ident:
          23.2.1.14       37.0.0.2        12.12.12.12     34.0.0.2
          99.0.0.1
    Peer LDP Ident: 11.11.11.11:0; Local LDP Ident 13.13.13.13:0
        TCP connection: 11.11.11.11.646 - 13.13.13.13.11013
        State: Oper; Msgs sent/rcvd: 1724/1730; Downstream
        Up time: 1d00h
        LDP discovery sources:
          Targeted Hello 13.13.13.13 -> 11.11.11.11, active, passive
        Addresses bound to peer LDP Ident:
          11.11.11.11     37.0.0.1        23.2.1.13
```

Step 4 To make sure the label forwarding table is built correctly, issue the **show mpls forwarding-table** command. The output shows the following data:

- Local tag—Label assigned by this router.
- Outgoing tag or VC—Label assigned by next hop.

- Prefix or Tunnel Id—Address or tunnel to which packets with this label are going.
- Bytes tag switched— Number of bytes switched out with this incoming label.
- Outgoing interface—Interface through which packets with this label are sent.
- Next Hop—IP address of neighbor that assigned the outgoing label.

```
Router# show mpls forwarding-table
osr1#show mpls forwarding-table
Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Untagged    223.255.254.254/32   \
                                       0          Gi2/1      23.2.0.1
20     Untagged    l2ckt(2)            55146580   Vl2        point2point
24     Pop tag     37.0.0.0/8          0          GE3/3      34.0.0.2
25     17          11.11.11.11/32      0          GE3/3      34.0.0.2
26     Pop tag     12.12.12.12/32      0          GE3/3      34.0.0.2
```

Step 5 To view the state of the currently routed VCs issue the **show mpls l2transport vc** command.

```
Router# show mpls l2transport vc
osr1#show mpls l2transport vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Vl2            Eth VLAN 2           11.11.11.11     2          UP

osr3#show mpls l2transport vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Gi7/4          Ethernet             13.13.13.13     2          UP
```

Step 6 Add the keyword **detail** to see detailed information about each VC.

```
Router# show mpls l2transport vc detail
osr1#show mpls l2transport vc detail
Local interface: Vl2 up, line protocol up, Eth VLAN 2 up
  Destination address: 11.11.11.11, VC ID: 2, VC status: up
    Tunnel label: 17, next hop 34.0.0.2
    Output interface: GE3/3, imposed label stack {17 18}
  Create time: 00:15:13, last status change time: 00:11:46
  Signaling protocol: LDP, peer 11.11.11.11:0 up
    MPLS VC labels: local 20, remote 18
    Group ID: local 71, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 407857, send 407684
    byte totals:   receive 53827205, send 55444697
    packet drops:  receive 0, send 0
```

ATM AAL5 over MPLS VC-Mode

ATM AAL5 over MPLS encapsulates ATM AAL5 SDUs in MPLS packets and forwards them across the MPLS network. Each ATM AAL5 SDU is transported as a single packet.

Supported OSMs

The following Catalyst 6000 family and Cisco 7600 series OSMs that support ATM AAL5 over MPLS:

- WS-X6182-2PA FlexWAN
- WS-X6582-2PA Enhanced FlexWAN
- OSM-2OC12-ATM-SI+
- OSM-2OC12-ATM-MM+
- ATM PA-A3
- ATM PA-A6

Configuring ATM AAL5 over MPLS VC-Mode

You can enable the MPLS backbone network to accept AAL5 PDUs by configuring the provider edge (PE) routers at the both ends of the MPLS backbone. To transport AAL5 PDUs over MPLS, set up a virtual circuit from the ingress PE router to the egress PE router. This virtual circuit transports the AAL5 PDUs from one PE router to the other.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm***slot/port*
4. **pvc** *vpi/vci* **l2transport**
5. **encapsulation aal5**
6. **mpls l2transport route** *destination vc-id*

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**interface atm***slot/port* **Example:** `Router(config)# interface atm1/1`	Specifies an ATM interface and enters interface configuration mode.
Step 4	**pvc** *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Assigns a virtual path identifier (VPI) and virtual circuit identifier (VCI). The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC. You can configure ATM AAL5 on PVCs only.
Step 5	**encapsulation aal5** **Example:** `Router(config-atm-vc)# encapsulation aal5`	Specifies ATM AAL5 encapsulation for the PVC. Make sure you specify the same encapsulation type on the PE and CE routers.
Step 6	`Router(config-if)#` **mpls l2transport route** *destination vc-id* **Example:** `Router(config-if)# mpls l2transport route 12.12.12.12 300`	Creates the VC to transport the Layer 2 packets.

Note You can configure VCs under point-to-point and multipoint subinterfaces, and all main interfaces.

Note You cannot configure multiple VCs with mixed encapsulations on OC-12 ATM OSMs under a multipoint subinterface or main interface; you can, however, configure multiple VCs with mixed encapsulation on WS-X6182-2PA FlexWAN or WS-X6582-2PA Enhanced FlexWAN modules under a multipoint subinterface or main interface with an Enhanced ATM Port Adapter (ATM PA).

The following example shows an AAL5 over MPLS configuration.

PE1

```
mpls label protocol ldp
mpls ldp router-id Loopback 0 force

!
interface Loopback0
 ip address 131.131.131.131 255.255.255.255

interface ATM9/1.502 point-to-point
 mls qos trust dscp
 pvc 4/42 l2transport
  encapsulation aal5
  mpls l2transport route 123.123.123.123 502
 !
```

PE2

```
mpls label protocol ldp
mpls ldp router-id Loopback 0 force

!
interface Loopback0
 ip address 123.123.123.123 255.255.255.255

!
interface ATM9/1.502 point-to-point
 description hi-there!
 mls qos trust dscp
 pvc 4/42 l2transport
  encapsulation aal5
  mpls l2transport route 131.131.131.131 502
 !
```

Verifying the Configuration

The **show running-config** command displays the contents of the currently running configuration file or the configuration for a specific interface (example is for PE1 above).

```
c31#show running-config interface ATM9/1.502
Building configuration...

Current configuration : 155 bytes
!
interface ATM9/1.502 point-to-point
 mls qos trust dscp
 pvc 4/42 l2transport
  encapsulation aal5
  mpls l2transport route 123.123.123.123 502
 ! !
end
```

The following **show mpls 12transport vc** command shows that the interface is configured for AAL5 over MPLS:

```
c31#show mpls l2transport vc vcid 502 detail
Local interface: AT9/1.502 up, line protocol up, ATM AAL5 4/42 up
  Destination address: 123.123.123.123, VC ID: 502, VC status: up
    Tunnel label: 25, next hop point2point
    Output interface: PO4/1, imposed label stack {25 20}
  Create time: 1d02h, last status change time: 00:33:28
  Signaling protocol: LDP, peer 123.123.123.123:0 up
    MPLS VC labels: local 19, remote 20
    Group ID: local 82, remote 80
    MTU: local 4470, remote 4470
    Remote interface description: hi-there!
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 1554872, send 1558795
    byte totals:   receive 2280634366, send 2281764774
    packet drops:  receive 0, send 0
```

The **show atm pvc** command shows all ATM permanent virtual connections (PVCs) and traffic information.

```
c31#
c31#show atm pvc 4/42
ATM9/1.502: VCD: 2, VPI: 4, VCI: 42
UBR, PeakRate: 599040
AAL5 over MPLS, etype:0x1C, Flags: 0xC3F, VCmode: 0x0
InPkts: 1573889, OutPkts: 1569951, InBytes: 2297940310, OutBytes: 2296823212
InPRoc: 0, OutPRoc: 0
InFast: 0, OutFast: 0, InAS: 1573889, OutAS: 1569951
InPktDrops: 0,  OutPktDrops: 0
InByteDrops: 0, OutByteDrops: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP
```

The **show atm vc** command displays all ATM permanent virtual circuits (PVCs) and switched virtual circuits (SVCs) and traffic information.

```
c31#show atm vc 2
ATM9/1.502: VCD: 2, VPI: 4, VCI: 42
UBR, PeakRate: 599040
AAL5 over MPLS, etype:0x1C, Flags: 0xC3F, VCmode: 0x0
InPkts: 1573896, OutPkts: 1569957, InBytes: 2297940836, OutBytes: 2296823668
InPRoc: 0, OutPRoc: 0
InFast: 0, OutFast: 0, InAS: 1573896, OutAS: 1569957
InPktDrops: 0,  OutPktDrops: 0
InByteDrops: 0, OutByteDrops: 0
OAM cells received: 0
OAM cells sent: 0
Status: UP
```

Troubleshooting Tips

The **debug acircuit**, **debug mpls l2transport ipc**, **debug cwan atom**, and **debug mpls l2transport vc** commands help in troubleshooting.

ATM Cell Relay over MPLS VC-Mode

The single cell relay feature allows you to insert one ATM cell in each MPLS packet.

Configuring ATM Cell Relay over MPLS VC-Mode

Perform this task to configure ATM cell relay over MPLS VC-Mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm***slot/port*
4. **pvc** *vpi/vci* **l2transport**
5. **encapsulation aal0**
6. **mpls l2transport route** *destination vc-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm***slot/port* **Example:** `Router(config)# interface atm1/1`	Specifies an ATM interface and enters interface configuration mode.
Step 4	**pvc** *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 0/100 l2transport`	Assigns a virtual path identifier (VPI) and virtual circuit identifier (VCI). The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation aal0** **Example:** `Router(config-atm-vc)# encapsulation aal0`	For ATM Cell Relay, this command specifies raw cell encapsulation for the interface.
Step 6	`Router(config-if)#` **mpls l2transport route** *destination vc-id* **Example:** `Router(config-if)# mpls l2transport route 13.13.13.13 100`	Creates the VC to transport the Layer 2 packets.

Note You can configure VCs under point-to-point or multipoint subinterfaces, and all main interfaces.

Note You cannot configure multiple VCs with mixed encapsulations on OC-12 ATM OSMs under a multipoint subinterface or main interface; you can, however, configure multiple VCs with mixed encapsulation on WS-X6182-2PA FlexWAN or WS-X6582-2PA Enhanced FlexWAN modules under a multipoint subinterface or main interface with an Enhanced ATM Port Adapter (ATM PA).

Note If each of the PE routers has an OC-12 ATM OSM interface, the path identifiers/virtual channel identifiers (VPIs/VCIs) do not need to match. If one of the PE routers at an end of the VC has a WS-X6182-2PA FlexWAN or a WS-X6582-2PA Enhanced FlexWAN with an ATM port adapter (PA) interface, then the VPIs/VCIs must match.

The following example shows a Cell Relay over MPLS configuration.

PE1	PE2

mpls label protocol ldp
mpls ldp router-id Loopback 0 force
 	
!
interface Loopback0
 ip address 131.131.131.131 255.255.255.255
 	
!
interface ATM9/1.501 point-to-point
 mls qos trust dscp
 pvc 4/41 l2transport
  encapsulation aal0
  mpls l2transport route 123.123.123.123 501
	mpls label protocol ldp
	mpls ldp router-id Loopback 0 force
	
	!
	interface Loopback0
	 ip address 123.123.123.123 255.255.255.255
	
	!
	interface ATM9/1.501 point-to-point
	 mls qos trust dscp
	 pvc 4/41 l2transport
	  encapsulation aal0
	  mpls l2transport route 131.131.131.131 501
	 !

Verifying the Configuration

The **show running-config** command displays the contents of the currently running configuration file or the configuration for a specific interface (this is for PE1 above).

```
c31#show running-config interface ATM9/1.501
Building configuration...

Current configuration : 155 bytes
!
interface ATM9/1.501 point-to-point
 mls qos trust dscp
 pvc 4/41 l2transport
  encapsulation aal0
  mpls l2transport route 123.123.123.123 501
 !
end
```

The **show mpls 12transport** command shows that the interface is configured for VC mode cell relay.

```
c31#show mpls l2transport vc vcid 501 detail
Local interface: AT9/1.501 up, line protocol up, ATM VCC CELL 4/41 up
  Destination address: 123.123.123.123, VC ID: 501, VC status: up
    Tunnel label: 25, next hop point2point
    Output interface: PO4/1, imposed label stack {25 19}
  Create time: 1d01h, last status change time: 00:15:55
  Signaling protocol: LDP, peer 123.123.123.123:0 up
    MPLS VC labels: local 18, remote 19
    Group ID: local 82, remote 80
    MTU: local n/a, remote n/a
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 48755771, send 48895612
    byte totals:   receive 2535300092, send 2542571824
    packet drops:  receive 0, send 0

c31#
```

The **show atm pvc** command shows all ATM permanent virtual connections (PVCs) and traffic information.

```
c31#show atm pvc 4/41
```

```
ATM9/1.501: VCD: 1, VPI: 4, VCI: 41
UBR, PeakRate: 599040
AAL0-Cell Relay over MPLS, etype:0x1B, Flags: 0xC3E, VCmode: 0x0
InBytes: 2567612684, OutBytes: 2560342200
Status: UP
```

The **show atm vc** command shows all ATM permanent virtual circuits (PVCs) and switched virtual circuits (SVCs) and traffic information.

```
c31#show atm vc 1

ATM9/1.501: VCD: 1, VPI: 4, VCI: 41
UBR, PeakRate: 599040
AAL0-Cell Relay over MPLS, etype:0x1B, Flags: 0xC3E, VCmode: 0x0
InBytes: 2567615492, OutBytes: 2560345424
Status: UP
```

Troubleshooting Tips

The **debug acircuit**, **debug mpls l2transport ipc**, **debug cwan atom**, and **debug mpls l2transport vc** commands help in troubleshooting.

Frame Relay Over MPLS

Frame Relay over MPLS encapsulates Frame Relay protocol data units (PDUs) in MPLS packets and forwards them across the MPLS network.

Supported Platforms and OSMs

FRoMPLS is supported on the following Catalyst 6000 family and Cisco 7600 series OSMs:

- OC-3 POS:
  - OSM-4OC3-POS-SI+
  - OSM-8OC3-POS-SI+, SL+
- OC-12 POS:
  - OSM-2OC12-POS-MM+, SI+, SL+
  - OSM-4OC12-POS-MM+, SI+, SL+
- OC-48 POS:
  - OSM-1OC48-POS-SS+, SI+, SL+
- Channelized:
  - OSM-1CHOC12/T3-SI
  - OSM-1CHOC12/T1-SI
- Channelized T3:
  - OSM-12CT3/T1
- OC-48 DPT/POS:
  - OSM-20C48/1DPT-SI

Note FRoMPLS is supported on any FlexWAN PA that supports Frame Relay encapsulation on the media type.

Configuring Frame Relay over MPLS with DLCI-to-DLCI Connections

Perform this task to configure Frame Relay over MPLS with DLCI-to-DLCI connections.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **frame-relay switching**
4. **interface serial***slot/port*
5. **frame-relay intf-type dce**
6. **encapsulation frame-relay** [**cisco** | **ietf**]
7. **connect** *connection-name interface dlci* **l2transport**

8. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**frame-relay switching** **Example:** `Router(config)# frame-relay switching`	Enables permanent virtual circuit (PVC) switching on a Frame Relay device.
Step 4	**interface serial***slot/port* **Example:** `Router(config)# interface Serial3/10/1`	Specifies a serial interface and enters interface configuration mode.
Step 5	**encapsulation frame-relay** [**cisco** | **ietf**] **Example:** `Router(config-if)# encapsulation frame-relay ietf`	Specifies Frame Relay encapsulation for the interface. You can specify different types of encapsulations. You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.
Step 6	**frame-relay intf-type dce** **Example:** `Router(config-if)# frame-relay intf-type dce`	Specifies that the interface is a DCE switch. You can also specify the interface to support NNI and DTE connections.
Step 7	**connect** *connection-name interface dlci* **l2transport** **Example:** `Router(config)# connect fr1 Serial5/1/0 1000 l2transport`	Defines connections between Frame Relay PVCs. Using the **l2transport** keyword specifies that the PVC will not be a locally switched PVC, but will be tunneled over the backbone network. The *connection-name* argument is a text string that you provide. The *interface* argument is the interface on which a PVC connection will be defined. The *dlci* argument is the DLCI number of the PVC that will be connected.
Step 8	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-fr-pw-switching)# xconnect 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets. In a DLCI-to DLCI connection type, Frame Relay over MPLS uses the **xconnect** command in connect submode.

The example below shows a Frame Relay over MPLS with DLCI-to-DLCI configuration.

PE1	PE2
frame-relay switching mpls label protocol ldp mpls ldp router-id Loopback0 force tag-switching id ! interface Loopback0 ip address 13.13.13.13 255.255.255.255 ! interface POS1/1 mtu 5000 no ip address encapsulation frame-relay IETF mls qos trust dscp clock source internal frame-relay lmi-type ansi frame-relay intf-type dce ! ! P router facing interface POS4/1 ! interface POS4/1 mtu 5000 ip address 32.0.0.1 255.0.0.0 mpls label protocol ldp tag-switching ip mls qos trust dscp clock source internal ! router ospf 100 log-adjacency-changes passive-interface POS1/1 network 13.13.13.13 0.0.0.0 area 100 network 32.0.0.0 0.255.255.255 area 100 ! connect atom_1 POS1/1 16 l2transport xconnect 11.11.11.11 100 encapsulation mpls	frame-relay switching mpls label protocol ldp mpls ldp router-id Loopback0 force tag-switching id ! interface Loopback0 ip address 11.11.11.11 255.255.255.255 ! interface POS7/1 mtu 5000 no ip address encapsulation frame-relay IETF mls qos trust dscp clock source internal frame-relay lmi-type ansi frame-relay intf-type dce ! ! P router facing interface POS8/2 ! interface POS8/2 mtu 5000 ip address 35.0.0.1 255.0.0.0 mpls label protocol ldp tag-switching ip mls qos trust dscp clock source internal ! router ospf 100 log-adjacency-changes passive-interface POS7/1 network 11.11.11.11 0.0.0.0 area 100 network 35.0.0.0 0.255.255.255 area 100 ! connect atom_1 POS7/1 17 l2transport xconnect 13.13.13.13 100 encapsulation mpls

Note It is not necessary for the DLCI of interface POS1/1 and the DLCI of interface POS7/1 to match. The DLCIs can be two separate DLCIs that you connect using the **connect** command.

Verifying the Configuration

Use the **show mpls l2transport vc** command to verify the configuration.

```
PE1# sh mpls l2 vc 100 detail
Local interface: PO1/1 up, line protocol up, FR DLCI 16 up
  Destination address: 11.11.11.11, VC ID: 100, VC status: up
    Tunnel label: 17, next hop point2point
    Output interface: PO4/1, imposed label stack {17 1009}
  Create time: 00:09:28, last status change time: 00:01:17
  Signaling protocol: LDP, peer 11.11.11.11:0 up
    MPLS VC labels: local 1009, remote 1009
    Group ID: local 0, remote 0
    MTU: local 5000, remote 5000
```

```
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 60, send 62
    byte totals:   receive 8870, send 9648
    packet drops:  receive 0, send 0


PE2# sh mpls l2 vc 100 detail
Local interface: PO7/1 up, line protocol up, FR DLCI 16 up
  Destination address: 13.13.13.13, VC ID: 100, VC status: up
    Tunnel label: 18, next hop point2point
    Output interface: PO8/2, imposed label stack {18 1009}
  Create time: 00:03:32, last status change time: 00:01:54
  Signaling protocol: LDP, peer 13.13.13.13:0 up
    MPLS VC labels: local 1009, remote 1009
    Group ID: local 0, remote 0
    MTU: local 5000, remote 5000
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 4, send 4
    byte totals:   receive 1416, send 1388
    packet drops:  receive 0, send 0


PE1# show frame-relay pvc 16

PVC Statistics for interface POS1/1 (Frame Relay DCE)

DLCI = 16, DLCI USAGE = SWITCHED(tag tunnel), PVC STATUS = ACTIVE, INTERFACE = POS1/1

  input pkts 68            output pkts 66           in bytes 11500
  out bytes 10688          dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 0         out bcast bytes 0
  switched pkts 0
  Detailed packet drop counters:
  no out intf 0            out intf down 0          no out PVC 0
  in PVC down 0            out PVC down 0           pkt too big 0
  shaping Q full 0         pkt above DE 0           policing drop 0
  pvc create time 00:16:28, last time pvc status changed 00:09:34


PE2#show frame-relay pvc 16

PVC Statistics for interface POS7/1 (Frame Relay DCE)

DLCI = 16, DLCI USAGE = SWITCHED(tag tunnel), PVC STATUS = ACTIVE, INTERFACE = POS7/1

  input pkts 27            output pkts 28           in bytes 5676
  out bytes 6110           dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 0         out bcast bytes 0
  switched pkts 0
  Detailed packet drop counters:
  no out intf 0            out intf down 0          no out PVC 0
  in PVC down 0            out PVC down 0           pkt too big 0
  shaping Q full 0         pkt above DE 0           policing drop 0
  pvc create time 00:10:50, last time pvc status changed 00:10:21
```

Layer 2 Local Switching

Local switching allows you to switch Layer 2 data between two interfaces of the same type (ATM to ATM or Frame Relay to Frame Relay). The interfaces can be on the same line card or on two different cards.

This section explains how to perform Layer 2 local switching-ATM to ATM and Frame Relay DCLI local switching and includes the following procedures:

- Configuring ATM VC to VC Local Switching with AAL5 Encapsulation, page 11-59
- Configuring ATM VC to VC Local Switching with AAL0 Encapsulation, page 11-61
- Configuring ATM VP to VP Local Switching with AAL0 Encapsulation, page 11-63
- Configuring Frame Relay DLCI Local Switching, page 11-65

Layer 2 Local Switching-ATM to ATM

Layer 2 Local Switching-ATM to ATM provides Layer 2 switching capability. It allow you to switch traffic coming from a customer ATM VC/VP to a Session Terminating Service Provider ATM VC/VP. Layer 2 Local Switching-ATM to ATM has three modes:

- ATM VC to VC local switching with AAL5 encapsulation
- ATM VC to VC local switching with AAL0 Encapsulation (Cell Relay mode)
- ATM VP to VP local switching with AAL0 Encapsulation

Supported Modules

Layer 2 Local Switching-ATM to ATM is supported on FlexWAN and Enhanced FlexWAN only.

Port adapter support is shown in Table 11-3.

Table 1-3 Layer 2 Local Switching-ATM to ATM Supported Port Adapters

ATM VC to VC Local Switching with AAL5 Encapsulation	ATM VC to VC Local Switching with AAL0 Encapsulation	ATM VP to VP Local Switching with AAL0 Encapsulation
PA-A3-OC3	PA-A3-OC3	PA-A3-OC3
PA-A3-E3	PA-A3-E3	PA-A3-E3
PA-A3-T3	PA-A3-T3	PA-A3-T3
PA-A6-OC3		
PA-A6-E3		
PA-A6-T3		

Restrictions

- ATM VC to VC local switching with AAL5 encapsulation
  - Does not support QoS.

  - Currently supported with supervisor engine 2 only.
- ATM VC to VC local switching with AAL0 encapsulation (Cell Relay mode)
  - Does not support QoS.
  - Currently supported with supervisor engine 2 only.
  - Each ATM cell is transported as a single packet; cell packing is not supported.
  - Configurable on permanent virtual circuits (PVCs) only.
  - Both ends of the connection require the same VPI/VCI. If the VPI/VCI is not same, then the connection comes up but the packet does not switch.
- ATM VP to VP local switching with AAL0 encapsulation
  - Does not support QoS.
  - Currently supported with supervisor engine 2 only.
  - Each ATM cell is transported as a single packet; cell packing is not supported
  - Configurable on permanent virtual pipes (PVPs) only.
  - Each ATM cell is transported as a single packet; cell packing is not supported.
  - Both ends of the connection require the same VPI/VCI. If the VPI/VCI is not same, then the connection comes up but the packet does not switch.

Configuring ATM VC to VC Local Switching with AAL5 Encapsulation

Perform this task to configure ATM VC to VC local switching.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm***slot/port*
4. **pvc** *vpi/vci* **l2transport**
5. **encapsulation aal5**

Note Repeat Steps 3 through 5 for the other interface.

6. **connect** *connection-name* **atm** *slot/port-1 |vpi/vci]* **atm** *slot/port-2* [*vpi*|*vci*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm***slot/port* **Example:** `Router(config)# interface atm1/0`	Specifies an ATM interface and enters interface configuration mode.
Step 4	**pvc** *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Assigns a virtual path identifier (VPI) and virtual circuit identifier (VCI). The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC. You can configure ATM AAL5 on PVCs only.
Step 5	**encapsulation aal5** **Example:** `Router(config-atm-vc)# encapsulation aal5`	Specifies ATM AAL5 encapsulation for the PVC.
Step 6	**connect** *connection-name* **atm** *slot/port-1* [*vpi/vci*] **atm** *slot/port-2* [*vpi*|*vci*] **Example:** `Router(config)# connect vp2vp ATM2/0/0 100 ATM2/1/0 100`	Connects the ATM interfaces.

The following example shows ATM VC to VC local switching with AAL5 Encapsulation.

```
Router(config)# int ATM2/0/0
Router(config-if)# pvc 100/100 l2transport
Router(config-atm-vc)# encapsulation aal5
Router(config)# int ATM2/1/0
Router(config-if)# pvc 105/105 l2transport
Router(config-atm-vc)# encapsulation aal5
Router(config)# connect vc2vc ATM2/0/0 100/100 ATM2/1/0 105/105
```

The **show atm pvc** command displays all ATM permanent virtual connections (PVCs) and traffic information.

```
router#show atm pvc 100/100
ATM2/0/0: VCD: 44, VPI: 100, VCI: 100
UBR, PeakRate: 149760
AAL5 L2transport, etype:0x1C, Flags: 0xC3F, VCmode: 0x0
InPkts: 0, OutPkts: 0, InBytes: 0, OutBytes: 0
InPRoc: 0, OutPRoc: 0, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
InPktDrops: 0,  OutPktDrops: 0
InByteDrops: 0, OutByteDrops: 0
```

```
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP

router#show atm pvc 105/105
ATM2/1/0: VCD: 46, VPI: 100, VCI: 100
UBR, PeakRate: 149760
AAL5 L2transport, etype:0x1C, Flags: 0xC3F, VCmode: 0x0
InPkts: 0, OutPkts: 0, InBytes: 0, OutBytes: 0
InPRoc: 0, OutPRoc: 0, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
InPktDrops: 0,  OutPktDrops: 0
InByteDrops: 0, OutByteDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP
```

Use the **show connection all** command to see all configured connections.

```
router#show connection all
ID   Name               Segment 1            Segment 2
State
================================================================================
36   vc2vc             ATM2/0/0 100/100     ATM2/1/0 105/105     UP
```

Configuring ATM VC to VC Local Switching with AAL0 Encapsulation

Perform this task to configure ATM VC to VC local switching with AAL0 encapsulation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm***slot/port*
4. **pvc** *vpi/vci* **l2transport**
5. **encapsulation aal0**

Note Repeat Steps 3 through 5 for the other interface.

6. **connect** *connection-name* **atm** *slot/port-1* |*vpi/vci*] **atm** *slot/port-2* [*vpi*|*vci*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm***slot/port* **Example:** `Router(config)# interface atm1/0`	Specifies an ATM interface and enters interface configuration mode.
Step 4	**pvc** *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Assigns a virtual path identifier (VPI) and virtual circuit identifier (VCI). The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation aal0** **Example:** `Router(config-atm-vc)# encapsulation aal0`	Specifies ATM AAL0 encapsulation for the PVC.
Step 6	**connect** *connection-name* **atm** *slot/port-1* *|vpi/vci|* **atm** *slot/port-2* [*vpi*|*vci*] **Example:** `Router(config)# connect vp2vp ATM2/0/0 100 ATM2/1/0 100`	Connects the ATM interfaces.

The following example shows ATM VC to VC local switching with AAL0 encapsulation.

```
Router(config)# int ATM2/0/0
Router(config-if)# pvc 100/100 l2transport
Router(config-atm-vc)# encapsulation aal0
Router(config)# int ATM2/1/0
Router(config-if)# pvc 100/100 l2transport
Router(config-atm-vc)# encapsulation aal0
Router(config)# connect vc2vc ATM2/0/0 100/100 ATM2/1/0 100/100
```

The **show atm pvc** command displays all ATM permanent virtual connections (PVCs) and traffic information.

```
router# show atm pvc 100/100
ATM2/0/0: VCD: 44, VPI: 100, VCI: 100
UBR, PeakRate: 149760
AAL5 L2transport, etype:0x1C, Flags: 0xC3F, VCmode: 0x0
InPkts: 0, OutPkts: 0, InBytes: 0, OutBytes: 0
InPRoc: 0, OutPRoc: 0, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
InPktDrops: 0,  OutPktDrops: 0
InByteDrops: 0, OutByteDrops: 0
```

```
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP

router# show atm pvc 100/100
ATM2/1/0: VCD: 46, VPI: 100, VCI: 100
UBR, PeakRate: 149760
AAL5 L2transport, etype:0x1C, Flags: 0xC3F, VCmode: 0x0
InPkts: 0, OutPkts: 0, InBytes: 0, OutBytes: 0
InPRoc: 0, OutPRoc: 0, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
InPktDrops: 0,  OutPktDrops: 0
InByteDrops: 0, OutByteDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP
```

Use the **show connection all** command to see all configured connections.

```
router# show connection all
ID   Name               Segment 1            Segment 2
State
============================================================================
36   vc2vc             ATM2/0/0 100/100     ATM2/1/0 105/105     UP
```

Configuring ATM VP to VP Local Switching with AAL0 Encapsulation

Perform this task to configure ATM VP to VP local switching with AAL0 encapsulation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm***slot/port*
4. **atm pvp** *vpi* **l2transport**

Note Repeat Steps 3 through 4 for the other interface.

5. **connect** *connection-name* **atm** *slot/port-1 |vpi/vci]* **atm** *slot/port-2* [*vpi*|*vci*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm***slot/port* **Example:** `Router(config)# interface atm1/0`	Specifies an ATM interface and enters interface configuration mode.
Step 4	**atm pvp** *vpi* **l2transport** **Example:** `Router(config-if)# atm pvp vpi 1 l2transport`	Specifies that the PVP is dedicated to transporting ATM cells. The **l2transport** keyword indicates that the PVP is for cell relay. Once you enter this command, you enter Layer 2 transport PVP submode. This submode is for Layer 2 transport only; it is not for regular PVPs.
Step 5	**connect** *connection-name* **atm** *slot/port-1* *|vpi/vci]* **atm** *slot/port-2* [*vpi*|*vci*] **Example:** `Router(config)# connect vp2vp ATM2/0/0 100 ATM2/1/0 100`	Connects the ATM interfaces.

The following example shows ATM VP to VP local switching.

```
Router(config)# int ATM2/0/0
Router(config-if)# atm pvp 100 l2transport
Router(config)# int ATM2/1/0
Router(config-if)# atm pvp 100 l2transport

Router(config)# connect vp2vp ATM2/0/0 100 ATM2/1/0 100
```

Use the **show atm vp** command to verify that the interface is configured for VP mode cell relay:

```
router# show connection all
ID   Name               Segment 1            Segment 2
State
========================================================================
36   vp2vp             ATM2/0/0 100         ATM2/1/0 100         UP

BRAS# show atm vp 100
ATM2/0/0 VPI: 100, Cell Relay,
ATM2/0/0  VPI: 100, PeakRate: 0, CesRate: 0, DataVCs: 0, CesVCs: 0, Status: ACTIVE

 VCD    VCI   Type   InPkts   OutPkts   AAL/Encap     Status
 45     3     PVC    0        0         F4 OAM        ACTIVE  46     4     PVC    0
0         F4 OAM        ACTIVE
TotalInPkts: 0, TotalOutPkts: 0, TotalInFast: 0, TotalOutFast: 0, TotalBroadcasts: 0
TotalInPktDrops: 0, TotalOutPktDrops: 0
ATM2/1/0 VPI: 100, Cell Relay,
```

```
ATM2/1/0  VPI: 100, PeakRate: 0, CesRate: 0, DataVCs: 0, CesVCs: 0, Status: ACTIVE

 VCD    VCI   Type   InPkts   OutPkts   AAL/Encap     Status
 47     3     PVC    0        0         F4 OAM        ACTIVE  48     4     PVC    0
0         F4 OAM        ACTIVE
TotalInPkts: 0, TotalOutPkts: 0, TotalInFast: 0, TotalOutFast: 0, TotalBroadcasts: 0
TotalInPktDrops: 0, TotalOutPktDrops: 0
```

Configuring Frame Relay DLCI Local Switching

Frame Relay DLCI local switching connects one DLCI on one interface to another DLCI on a different interface in the same Cisco 7600 series router. Perform this task to set up Frame Relay DLCI local switching.

Note You use the steps below on two DLCIs in order to connect them.

Note The **frame-relay route** command is no longer supported for this configuration; use the **connect** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **frame-relay switching**
4. **interface serial***slot/port*
5. **encapsulation frame-relay** [**cisco** | **ietf**]
6. **frame-relay intf-type dce**
7. **connect** *connection-name interface_1 dlci_1 interface_2 dlci_2*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**frame-relay switching** **Example:** `Router# frame-relay switching`	Enable permanent virtual switching (PVC) switching on a Frame Relay DCE device or a Network-to-Network Interface (NNI).
Step 4	**interface serial***slot/port* **Example:** `Router(config)# interface Serial3/1/0`	Specifies a serial interface.
Step 5	**encapsulation frame-relay** [**cisco** | **ietf**] **Example:** `Router(config)# encapsulation frame-relay ietf`	Specifies Frame Relay encapsulation for the interface. You can specify different types of encapsulations. You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.
Step 6	**frame-relay intf-type dce** **Example:** `Router(config)# frame-relay intf-type dce`	Specifies that the interface is a DCE switch. You can also specify the interface to support NNI and DTE connections.
Step 7	**connect** *connection-name interface_1 dlci_1 interface_2 dlci_2* **Example:** `Router(config)# connect fr-route1 pos1/1 110 serial6/1/0 61`	Defines connections between Frame Relay PVCs. The *connection-name* argument is a text string that you provide. The *interface* argument is the interface on which a PVC connection will be defined. The *dlci* argument is the DLCI number of the PVC that will be connected.

The following configuration provides an example of Frame Relay DLCI local switching on the same router OSR4) between a DLCI on interface POS4/1 to a DLCI on interface POS4/2 (OSR1 and OSR3 are CEs).

Note It is not necessary for the DLCI of interface POS4/1 and the DLCI of interface POS4/2 to match. The DLCIs can be two separate DLCIs that you connect using the **connect** command.

Configuration on OSR1

```
!
interface POS4/1
 mtu 9000
 no ip address
 encapsulation frame-relay
!
interface POS4/1.1 point-to-point
 ip address 11.11.1.1 255.255.255.0
 frame-relay interface-dlci 16
```

Configuration on OSR4

```
!
frame-relay switching
!
interface POS4/1
```

```
 mtu 9000
 no ip address
 encapsulation frame-relay
 clock source internal
 frame-relay intf-type dce
!
interface POS4/2
 mtu 9000
 no ip address
 encapsulation frame-relay
 clock source internal
 frame-relay intf-type dce
!
connect test1 POS4/1 16 POS4/2 16
!
```

Configuration on OSR3

```
!
interface POS8/2
 mtu 9000
 no ip address
 encapsulation frame-relay
!
interface POS8/2.1 point-to-point
 ip address 11.11.1.2 255.255.255.0
 frame-relay interface-dlci 16
!
```

Use the **ping** command to verify basic connectivity.

```
osr1#ping
Protocol [ip]:
Target IP address: 11.11.1.2
Repeat count [5]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 11.11.1.2, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/1/4 ms
osr1#
```

Use the **show frame pvc** command to view statistics for all Virtual Circuit (VC) components.

```
osr4#sh frame pvc 16

PVC Statistics for interface POS4/1 (Frame Relay DCE)

DLCI = 16, DLCI USAGE = SWITCHED(fr), PVC STATUS = ACTIVE, INTERFACE = POS4/1

  input pkts 100           output pkts 100          in bytes 10400
  out bytes 10400          dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                 out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 0         out bcast bytes 0
  switched pkts 0
  Detailed packet drop counters:
  no out intf 0            out intf down 0          no out PVC 0
  in PVC down 0            out PVC down 0           pkt too big 0
  shaping Q full 0         pkt above DE 0           policing drop 0
  pvc create time 02:11:44, last time pvc status changed 02:04:23
```

```
PVC Statistics for interface POS4/2 (Frame Relay DCE)

DLCI = 16, DLCI USAGE = SWITCHED(fr), PVC STATUS = ACTIVE, INTERFACE = POS4/2

  input pkts 100           output pkts 100          in bytes 10400
  out bytes 10400          dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 0         out bcast bytes 0
  switched pkts 0
  Detailed packet drop counters:
  no out intf 0            out intf down 0          no out PVC 0
  in PVC down 0            out PVC down 0           pkt too big 0
  shaping Q full 0         pkt above DE 0           policing drop 0
  pvc create time 02:11:45, last time pvc status changed 02:07:30
osr4#
```

Use the **show connect all** command to see the connections.

```
osr4# sh connect all

ID   Name               Segment 1            Segment 2           State
======================================================================
1    test1             POS4/1 16            POS4/2 16           UP
```

Troubleshooting Tips

The **debug frame-relay event**, **debug acircuit**, **debug mpls l2transport ipc**, **debug cwan atom**, and **debug mpls l2transport vc** commands help in troubleshooting.

Enabling Other PE Devices to Transport Frame Relay Packets

You can configure an interface as a data terminal equipment (DTE) device or a data circuit-terminating equipment (DCE) switch, or as a switch connected to a switch with network-to-network interface (NNI) connections. Use the following command in interface configuration mode:

frame-relay intf-type [dce | dte | nni]

The keywords are explained in the following table:

Keyword	Description
dce	Enables the router or access server to function as a switch connected to a router.
dte	Enables the router or access server to function as a DTE device. DTE is the default.
nni	Enables the router or access server to function as a switch connected to a switch.

Local Management Interface and Frame Relay over MPLS

Local Management Interface (LMI) is a protocol that communicates status information about permanent virtual circuits (PVCs). When a PVC is added, deleted, or changed, the LMI notifies the endpoint of the status change. LMI also provides a polling mechanism that verifies that a link is up.

Note LMI is operational only when you enable keepalives on the interfaces (keepalive packets keep the interface active).

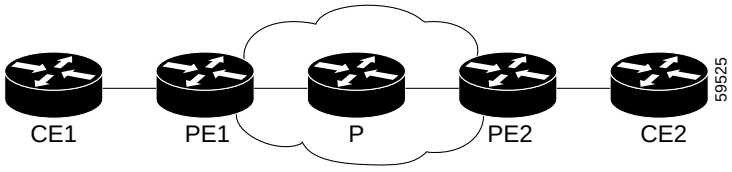
How LMI Works

To determine the PVC status, LMI checks that a PVC is available from the reporting device to the Frame Relay end-user device. If a PVC is available, LMI reports that the status is "Active," which means that all interfaces, line protocols, and core segments are operational between the reporting device and the Frame Relay end-user device. If any of those components is not available, the LMI reports a status of "Inactive."

Note Only the DCE and NNI interface types can report LMI status.

Figure 11-2 is a sample topology that helps illustrate how LMI works.

Figure 1-2 Sample Topology



Note the following:

- CE1 and PE1 and PE2 and CE2 are Frame Relay LMI peers.
- CE1 and CE2 can be Frame Relay switches or end-user devices.
- Each Frame Relay PVC is composed of multiple segments.
- The DLCI value is local to each segment and is changed as traffic is switched from segment to segment. Two Frame Relay PVC segments exist; one is between PE1 and CE1 and the other is between PE2 and CE2.

DLCI-to-DLCI Connections

If you have DLCI-to-DLCI connections, LMI runs locally on the Frame Relay ports between the PE and CE devices.

- CE1 sends an active status to PE1 if the PVC for CE1 is available. If CE1 is a switch, LMI checks that the PVC is available from CE1 to the user device attached to CE1.
- PE1 sends an active status to CE1 if the following conditions are met:
  - A PVC for PE1 is available.
  - PE1 has received an MPLS label from the remote PE router.
  - An MPLS tunnel label exists between PE1 and the remote PE.
  - CE2 reports an Active status to PE2. If CE2 is a switch, LMI checks that the PVC is available from PE1 to the end-user device attached to CE2.

For DTE/DCE configurations, the following LMI behavior exists:

The Frame Relay device accessing the network (DTE) does the polling. The network device (DCE) responds to the LMI polls. Therefore, if a problem exists on the DTE side, the DCE is not aware of the problem, because it does not poll.

For More Information About LMI

For information about LMI, including configuration instructions, see the following document:

Configuring Frame Relay, Configuring the LMI at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fwan_c/wcffrely.htm#xtocid8

DE/CLP and EXP Mapping on FR/ATMoMPLS VC

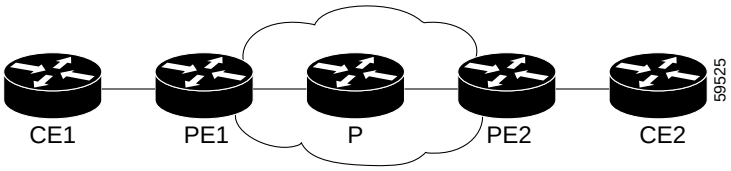
The DE/CLP and EXP Mapping on FR/ATMoMPLS VC feature allows you to map the Frame Relay Discard Eligibility (DE) bit or the ATM Congestion Loss Priority (CLP) bit to the MPLS EXP value at the ingress to an MPLS AToM network and to map the MPLS EXP value to the FR-DE or ATM CLP bit at the egress of an MPLS AToM network.

The DE bit indicates that a frame has lower importance than other frames. Similarly, the ATM CLP bit indicates whether the cell may be discarded if it encounters extreme congestion as it moves through the network.

In the figure below, the PE1 tags the incoming packet with the MPLS EXP value and sends the packet to the next hop. At each hop, matching is on the EXP value. At the PE2 egress, however, the packet is no longer MPLS but IP, so matching cannot occur on the EXP value.

Internally, the OSM preserves the EXP value in the QoS group so matching on the QoS group at the PE2 egress provides the same effect as matching on the EXP value.

Figure 1-3 DE/CLP and EXP Mapping



See the following sections:

- Match on ATM CLP Bit, page 11-70
- Match on FR-DE Bit, page 11-72
- Set on ATM CLP Bit, page 11-76
- Set on FR-DE Bit, page 11-78

Match on ATM CLP Bit

Use Match on ATM CLP Bit at the ingress to an MPLS AToM network to map the ATM cell loss priority (CLP) of the packet arriving at an interface to the EXP value, and then apply the desired QoS functionality and actions (for example, traffic policing) to those packets.

Restrictions for Match on ATM CLP Bit

The following restrictions apply:

- This feature is supported on policy maps attached to ATM permanent virtual circuits (PVCs) only.
- This feature is not supported on the OSM-2OC12-ATM-MM or OSM-2OC12-ATM-MM+.

Configuring Match on ATM CLP Bit for Ingress Policy

Perform the following steps to configure Match on ATM CLP Bit for the ingress policy:

	Command or Action	Purpose
Step 1	`Router# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Router(config)# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# class-map class-name`	Specifies the user-defined name of the traffic class.
Step 4	`Router(config-cmap)# match atm clp`	Enables packet matching on the basis of the ATM CLP bit set to 1.
Step 5	`Router(config-cmap)# policy-map policy-name`	Specifies the name of the traffic policy to configure.
Step 6	`Router(config-pmap)# class class-name`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 7	`Router(config-pmap-c)# set mpls experimental value`	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Step 8	`Router(config)# interface atm interface-number`	Enters interface configuration mode.
Step 9	`Router(config-if)# pvc [name] vpi/vci [l2transport]`	Entesr ATM virtual circuit configuration mode.
Step 10	`Router(config-if)# service-policy input policy-name`	Attaches a traffic policy to an interface.

The following is an example of a Match on ATM CLP Bit configuration:

```
Router# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#class-map CLP
Router(config-cmap)#match atm clp
Router(config-cmap)#exit
Router(config)#policy-map CLP2EXP
Router(config-pmap)#class CLP
Router(config-pmap-c)#set mpls experimental 1
Router(config-pmap-c)#exit
Router(config-pmap)#interface ATM3/0
Router(config-if)#pvc 1/100
Router(cfg-if-atm-l2trans-pvc)#service-policy input CLP2EXP
Router(cfg-if-atm-l2trans-pvc)#end
Router#
```

Use the **show policy-map interface** command to verify the Match on ATM CLP bit as in the following example:

```
CFLOW_PE1# show policy-map interface a3/0
ATM3/0/0: VC 1/100 -

Service-policy input: CLP2EXP

Class-map: CLP (match-all)
200 packets, 22400 bytes
5 minute offered rate 2000 bps, drop rate 0 bps
Match: atm clp
QoS Set
mpls experimental imposition 1
Packets marked 200

Class-map: class-default (match-any)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: any
CFLOW_PE1#
```

Match on FR-DE Bit

Use Match on FR-DE Bit at the ingress to an MPLS AToM network to map the Frame Relay discard eligible (DE) bit of the packet arriving at an interface to the EXP value.

Restrictions for Match on FR-DE Bit

The following restriction applies to this feature:

- Use policy matching on the FR-DE as an input policy only.

Configuring Match on FR-DE Bit for Ingress Policy

Perform the following steps to configure Match on FR-DE Bit for the ingress policy:

	Command or Action	Purpose
Step 1	`Router# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Router(config)# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# class-map` *class-name*	Specifies the user-defined name of the traffic class.
Step 4	`Router(config-cmap)# match fr-de`	Matches on packets that have the Frame Relay DE bit set to 1.
Step 5	`Router(config-cmap)# policy-map` *policy-name*	Specifies the name of the traffic policy to configure.
Step 6	`Router(config-pmap)# class` *class-name*	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 7	`Router(config-pmap-c)# set mpls experimental` *value*	Designates the value to which the MPLS bits are set if the packets match the specified policy map.

	Command or Action	Purpose
Step 8	`Router(config)# interface slot/port`	Enters the interface.
Step 9	`Router(config)# map-class frame-relay class-map-name`	Creates a Frame Relay map class where *class-map-name* is the name of the class map.
	Note In Step 10 below, you can apply the map-class policy to the main interface so that all DLCIs have the same policy or you can apply the map-class policy to each DLCI.	
Step 10	`Router(config-if)# service-policy input policy-name`	Attaches a traffic policy to an interface.

The following example shows how to configure Match on FR-DE Bit for the ingress policy by applying the map-class policy to the main interface:

```
osr3# show class-map match_fr-de
 Class Map match-all match_fr-de (id 2)
   Match fr-de

osr3# show policy fr-de_mpls4
  Policy Map fr-de_mpls4
    Class match_fr-de
      set mpls experimental imposition 4
    Class class-default
      set mpls experimental imposition 4

osr3# show run map-class | begin fr-de_mpls4
map-class frame-relay fr-de_mpls4
 service-policy input fr-de_mpls4
!
map-class frame-relay fr-de_mpls0
 service-policy input fr-de_mpls0
!

osr3# show run int pos1/0
Building configuration...

Current configuration : 196 bytes
!
interface POS1/0
 mtu 5000
 no ip address
 encapsulation frame-relay IETF
 no keepalive
 clock source internal
 pos scramble-atm
 frame-relay intf-type dce
end

connect frompls_1 POS1/0 16 l2transport
 xconnect 11.11.11.11 2001 encapsulation mpls
 !
!
connect frompls_2 POS1/0 17 l2transport
 xconnect 11.11.11.11 2002 encapsulation mpls
 !
!
connect frompls_3 POS1/0 18 l2transport
 xconnect 11.11.11.11 2003 encapsulation mpls
 !
!
osr3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
osr3(config)#interface POS1/0
osr3(config-if)#frame-relay class fr-de_mpls4
osr3(config-if)#
osr3(config-if)#^Z

osr3# sh run int pos1/0
Building configuration...

Current configuration : 196 bytes
!
interface POS1/0
 mtu 5000
 no ip address
 encapsulation frame-relay IETF
 no keepalive
 clock source internal
 pos scramble-atm
 frame-relay class fr-de_mpls4
 frame-relay intf-type dce
end
```

Verify the configuration with the **show policy-map interface** command.

```
osr3# show policy-map interface pos1/0
 POS1/0: DLCI 16 -

  Service-policy input: fr-de_mpls4

    Class-map: match_fr-de (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: fr-de
      QoS Set
        mpls experimental imposition 4
           Packets marked 0

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      QoS Set
        mpls experimental imposition 4
           Packets marked 0
 POS1/0: DLCI 1007 -

  Service-policy input: fr-de_mpls4

 --More--
```

The following example shows how to configure Match on FR-DE Bit for the ingress policy by applying the map-class policy to the different DLCIs:

```
osr1# show policy-map fr-de_mpls2
  Policy Map fr-de_mpls2
    Class match_fr-de
      set mpls experimental imposition 2
    Class class-default
      set mpls experimental imposition 2
osr1# show policy-map fr-de_mpls3
  Policy Map fr-de_mpls3
    Class match_fr-de
      set mpls experimental imposition 3
    Class class-default
```

```
      set mpls experimental imposition 3
osr1# show class-map match_fr-de
 Class Map match-all match_fr-de (id 1)
   Match fr-de

osr1# show run map-class | begin fr-de_mpls
map-class frame-relay fr-de_mpls2
 service-policy input fr-de_mpls2
!
map-class frame-relay fr-de_mpls3
 service-policy input fr-de_mpls3
!
osr1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
osr1(config)# interface pos1/7
osr1(config-if)# frame-relay interface-dlci 16 switched
osr1(config-fr-dlci)# class fr-de_mpls2
osr1(config-fr-dlci)# exit
osr1(config-if)#
osr1(config-if)# frame-relay interface-dlci 17 switched
osr1(config-fr-dlci)# class fr-de_mpls3
osr1(config-fr-dlci)#
osr1(config-fr-dlci)# exit
osr1(config-if)#
osr1(config-if)#^Z
osr1#

osr1# show run int pos1/7
Building configuration...

Current configuration : 39671 bytes
!
interface POS1/7
 mtu 5000
 no ip address
 encapsulation frame-relay IETF
 no keepalive
 mls qos trust dscp
 clock source internal
 pos scramble-atm
 frame-relay interface-dlci 16 switched
  class fr-de_mpls2
 frame-relay interface-dlci 17 switched
  class fr-de_mpls3
 frame-relay interface-dlci 18 switched
 frame-relay interface-dlci 19 switched
...
```

Verify the configuration with the **show policy-map interface** command.

```
osr1# show policy interface pos1/7
 POS1/7: DLCI 16 -

  Service-policy input: fr-de_mpls2

    Class-map: match_fr-de (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: fr-de
      QoS Set
        mpls experimental imposition 2
           Packets marked 0

    Class-map: class-default (match-any)
```

```
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      QoS Set
        mpls experimental imposition 2
           Packets marked 0
 POS1/7: DLCI 17 -

  Service-policy input: fr-de_mpls3

    Class-map: match_fr-de (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: fr-de
      QoS Set
        mpls experimental imposition 3
           Packets marked 0

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      QoS Set
        mpls experimental imposition 3
           Packets marked 0
osr1#
```

Set on ATM CLP Bit

Use Set on ATM CLP Bit at the egress of an MPLS AToM network to map the EXP value to the ATM CLP bit.

Restrictions for Set on ATM CLP Bit

The following restrictions apply to this feature:

- This feature is supported on policy maps attached to ATM permanent virtual circuits (PVCs) only.
- This feature is not supported on the OSM-2OC12-ATM-MM or OSM-2OC12-ATM-MM+

Configuring Set on ATM CLP Bit for Egress Policy

Perform the following steps to configure Set on ATM CLP Bit for the ingress policy:

	Command or Action	Purpose
Step 1	`Router# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Router(config)# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# class-map` *class-name*	Specifies the user-defined name of the traffic class.

	Command or Action	Purpose
Step 4	`Router(config-cmap)# match qos-group qos-group-value`	Identifies a specific quality of service (QoS) group value as a match criterion. The QoS group value has no mathematical significance. **Note** The QoS group concept is directly derived from the incoming MPLS EXP value and is valid only with AToM. You cannot use MQC to set QoS group value.
Step 5	`Router(config-cmap)# policy-map policy-name`	Specifies the name of the traffic policy to configure.
Step 6	`Router(config-pmap)# class class-name`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 7	`Router(config-pmap-c)# set atm-clp`	Sets the cell loss priority (CLP) bit when a policy map is configured.
Step 8	`Router(config)# interface slot/port`	Enters the interface and enters interface configuration mode.
Step 9	`Router(config-if)# service-policy input policy-name`	Attaches a traffic policy to an interface.

The following shows how to configure Set on ATM CLP Bit:

```
arthos# show policy-map qg2clp
 Policy Map qg2clp
   Class qg1
     set atm-clp
arthos# show class-map qg1
Class Map match-all qg1 (id 23)
  Match qos-group 1

arthos# show run int a9/1
interface ATM9/1
no ip address
atm clock INTERNAL
atm mtu-reject-call
mls qos trust dscp
pvc 1/100 l2transport
  encapsulation aal5
  mpls l2transport route 101.101.101.101 1000
  service-policy out qg2clp
```

Verify the configuration with the **show policy-map interface** command.

```
arthos# show policy interface ATM9/1
ATM9/1: VC 1/100 -

 Service-policy output: qg2clp

   Class-map: qg1 (match-all)
     1000 packets, 0 bytes
     5 minute offered rate 0 bps, drop rate 0 bps
     Match: qos-group 1
     QoS Set
       atm-clp
```

```
                Packets marked 1000
```

Set on FR-DE Bit

Use Set on FR-DE Bit at the egress of an MPLS AToM network to map the EXP value to the FR-DE bit.

Configuring Set on FR-DE for Egress Policy

Perform the following steps to configure Set on FR-DE Bit for the egress policy:

	Command or Action	Purpose
Step 1	`Router# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Router(config)# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# class-map` *class-name*	Specifies the user-defined name of the traffic class.
Step 4	`Router(config-cmap)# match qos-group` *qos-group-value*	Identifies a specific quality of service (QoS) group value as a match criterion where the range of the *qos-group-value* is 0-7.
Step 5	`Router(config-cmap)# policy-map` *policy-name*	Specifies the name of the traffic policy to configure.
Step 6	`Router(config-pmap)# class` *class-name*	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 7	`Router(config-pmap-c)# set fr-de`	Changes the DE bit setting in the address field of a Frame Relay frame to 1 for all traffic leaving an interface.
Step 8	`Router(config)# interface` *slot/port*	Enters the interface and enters interface configuration mode.
	Note In Step 9 below you can apply the map-class policy to main interface so that all DLCIs have the same policy or you can apply the map-class policy to each DLCI.	
Step 9	`Router(config-if)# service-policy output` *policy-name*	Attaches a traffic policy to an interface.

The following shows how to configure Set on FR-DE Bit:

```
arthos# show policy-map qg2de
 Policy Map qg2de
   Class qg1
     set fr-de
arthos# show class-map qg1
Class Map match-all qg1 (id 23)
  Match qos-group 1

arthos# show run int pos2/2/0
interface POS2/2/0
no ip address
```

```
encapsulation frame-relay
frame-relay interface-dlci 16 switched
   class QG2DE
```

Verify the configuration with the **show policy-map interface** command.

```
arthos# show policy interface POS2/2/0
POS2/2/0: DLCI 16 -

 Service-policy output: qg2de

   Class-map: qg1 (match-all)
     1000 packets, 0 bytes
     5 minute offered rate 0 bps, drop rate 0 bps
     Match: qos-group 1
     QoS Set
       fr-de
         Packets marked 1000
```

How to Configure QoS with AToM

The following QoS features are supported on AToM:

- Marking on CE facing card—(imposition packets) with match criteria, match-dlci, match-any, or class-default.

Note For Marking on CE facing card, match-dlci applies to the FlexWAN module only.

- Shaping on the core-facing card, with match exp, and match-any.
- Shaping on the CE-facing card - (disposition packets) with match-any.
- WRED on the core-facing card with match criteria, match-exp, or match-any

This section explains how to configure QoS with AToM and includes the following procedures:

- How to Set Experimental Bits with AToM, page 11-79
- Setting the Priority of Packets with EXP Bits, page 11-83
- Enabling Traffic Shaping, page 11-85

Note PFC QoS features do not apply to ATMoMPLS and FRoMPLS packets.

How to Set Experimental Bits with AToM

MPLS AToM uses the three experimental bits in a label to determine the queue of packets. You statically set the experimental bits in both the VC label and the LSP tunnel label, because the LSP tunnel label might be removed at the penultimate router. The following sections explain the transport-specific implementations of the EXP bits.

Ethernet over MPLS and EXP Bits

Note The information in this section is for OSM-based EoMPLS only. For information on PFC3BXL QoS, see http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/swcg/qos.htm.

OSM-based EoMPLS supports the following QoS implementations:

- VLAN interface policies
- Core-facing interface policy

You apply a VLAN interface policy to an individual VLAN. You may configure a unique policy for each individual VLAN. Within a policy, you can classify on 802.1q P bits to set the MPLS experimental bits. You can also implement a single traffic shaper that applies to all traffic within the VLAN.

Note Within a VLAN interface policy, only the **shape average** and **set mpls experimental** commands are supported. Within the **shape average** command, only the *cir* argument is valid for EoMPLS.

You apply a core-facing interface policy to the EoMPLS uplink interface. This policy applies to traffic from all VLANs. It does not distinguish between different VLANs. Within a policy, you can classify on MPLS experimental bits and configure the following features:

- Class-based traffic shaping
- Class-based weighted fair queuing (CBWFQ)
- Low latency queuing (LLQ)
- Weighted random early detection (WRED)

Note You cannot use both VLAN interface policies and core-facing interface policies at the same time. If you configure QoS for OSM-based EoMPLS, you must select either VLAN interface policies or a core-facing interface policy.

For more information on VLAN interface policies, see "Setting the Priority of Packets with the Experimental Bits" section on page 11-80 and "Enabling Traffic Shaping" section on page 11-81.

For more information on core-facing policies, see "Configuring MPLS QoS" section on page 11-13.

For more information on the commands used to enable Quality of Service, see the following documents:

- *Modular Quality of Service Command-Line Interface*
- *Cisco IOS Quality of Service Solutions Command Reference, Release 12.2*

Setting the Priority of Packets with the Experimental Bits

Ethernet over MPLS provides Quality of Service (QoS) using the three experimental bits in a label to determine the priority of packets. To support QoS between LERs, set the experimental bits in both the VC and tunnel labels. If you do not assign values to the experimental bits, the priority bits in the 802.1q header's "tag control information" field and are written into the experimental bit fields.

Perform the following steps to set the experimental bits:

	Command	Purpose
Step 1	`Router(config)# class-map class-name`	Specifies the user-defined name of the traffic class.
Step 2	`Router(config-cmap)# match cos 0-7`	Specifies that IEEE 802.1Q packets with the cos-values of 0-7 be matched. As an alternative, you can use the **match any** command.
Step 3	`Router(config-cmap)# policy-map policy-name`	Specifies the name of the traffic policy to configure.
Step 4	`Router(config-pmap)# class class-name`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 5	`Router (config-pmap-c)# set mpls experimental value`	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Step 6	`Router(config)# interface vlanvlan-number`	Enters the VLAN interface and enters interface configuration mode.
Step 7	`Router(config-if)# service-policy [input | output] policy-name`	Attaches a traffic policy to an interface.

Note You can enable traffic shaping and set experimental bits in the same policy-map.

Note You can configure the service-policy for either the input or the output direction. However, the policy is always implemented on the core-facing OSM port and is applied only to the traffic leaving the core-facing OSM port.

Enabling Traffic Shaping

Traffic shaping limits the rate of transmission of data. Average rate shaping limits the transmission rate to the committed information rate (CIR). To add traffic shaping, issue the following commands:

	Command	Purpose
Step 1	`Router(config)# class-map class-name`	Specifies the user-defined name of the traffic class.
Step 2	`Router(config-cmap)# match any`	Specifies that all packets will be matched. (Using the class-default in the policy-map would have the same effect.)
Step 3	`Router(config-cmap)# policy-map policy-name`	Specifies the name of the traffic policy to configure.
Step 4	`Router(config-pmap)# class class-name`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 5	`Router (config-pmap-c)# shape average cir [1] [2]`	Shapes traffic according to the bit rate you specify.

	Command	Purpose
Step 6	`Router(config)# interface vlan`*vlan-number*	Enters the VLAN interface and enters interface configuration mode.
Step 7	`Router(config-if)# service-policy [input | output]` *policy-name*	Assigns a traffic policy to an interface.

1. Only supported parameters are shown.
2. See Table 1-1 on page 1-5.

The shape average rate is rounded to the nearest multiple of the link rate divided by 255. If the shape value is lower than the link rate divided by 255, it is rounded up to link rate divided by 255.

This example shows how the shape value is rounded:

```
Router# show pol p2
 Policy Map p2
  class  any-pkt
   shape average 2000000 8000 8000

Router# show pol int

 Vlan101

  service-policy input:p2

    class-map:any-pkt (match-all)
      2018169 packets, 4575195376 bytes
      30 second offered rate 295768000 bps, drop rate 0 bps
      match:any
      queue size 0, queue limit 0
      packets input 40492, packet drops 1977677
      tail/random drops 0, no buffer drops 0, other drops 1977677
      shape:cir 2000000,  Bc 8000,  Be 8000
      (shape parameter is rounded to 2439000 due to granularity)
      input bytes 40847436, shape rate 1874000 bps

    class-map:class-default (match-any)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      match:any
        0 packets, 0 bytes
        30 second rate 0 bps
```

To display the traffic policy attached to an interface, issue the following command:

```
Router# show policy-map vlan50
service-policy input: badger

    class-map: blue (match-all)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      match: any
      queue size 0, queue limit 2
      packets input 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0
      shape: cir 2000000,  Bc 8000,  Be 8000
        output bytes 0, shape rate 0 bps

    class-map: class-default (match-any)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
```

```
match: any
  0 packets, 0 bytes
  30 second rate 0 bps
```

ATM AAL5 over MPLS and EXP Bits

- ATM AAL5 over MPLS allows you to statically set the experimental bits.
- If you do not assign values to the experimental bits, the priority bits in the header's “tag control information” field are set to zero.

ATM Cell Relay over MPLS and EXP Bits

- ATM Cell Relay over MPLS allows you to statically set the experimental bits in VC mode.
- If you do not assign values to the experimental bits, the priority bits in the header's “tag control information” field are set to zero.

Frame Relay over MPLS and EXP Bits

Frame Relay over MPLS provides QoS using the three experimental bits in a label to determine the priority of PDUs. If you do not assign values to the experimental bits, the priority bits in the header's “tag control information” field are set to zero. For FRoMPLS, you must set the experimental bits on a per-DLCI basis.

Setting the Priority of Packets with EXP Bits

Set the experimental bits in both the VC label and the LSP tunnel label. You set the experimental bits in the VC label, because the LSP tunnel label might be removed at the penultimate router.

Perform the following steps to set the experimental bits.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-name*
4. **match any**
5. **policy-map** *policy-name*
6. **class** *class-name*
7. **set mpls experimental** *value*
8. **interface***slot/port*
9. **service-policy input** *policy-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-name* **Example:** `Router(config)# class-map jane`	Specifies the user-defined name of the traffic class.
Step 4	**match any** **Example:** `Router(config-cmap)# match any`	Specifies that all packets will be matched. In this release, use only the **any** keyword. Other keywords might cause unexpected results.
Step 5	**policy-map** *policy-name* **Example:** `Router(config-cmap)# policy-map doe`	Specifies the name of the traffic policy to configure.
Step 6	**class** *class-name* **Example:** `Router(config-pmap)# class jane`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 7	**set mpls experimental** *value* **Example:** `Router(config-pmap-c)# set mpls experimental 7`	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Step 8	**interface***slot/port* `Router(config)# interface atm4/0`	Enters the interface and enters interface configuration mode.
Step 9	**service-policy input** *policy-name* **Example:** `Router(config-if)# service-policy input doe`	Attaches a traffic policy to an interface.

Enabling Traffic Shaping

Traffic shaping limits the rate of transmission of data. Average rate shaping limits the transmission rate to the committed information rate (CIR). To add traffic shaping, issue the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-name*
4. **match any**
5. **policy-map** *policy-name*
6. **class** *class-name*
7. **shape average** *bit rate*
8. **interface***slot/port*
9. **service-policy input** *policy-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-name* **Example:** `Router(config)# class-map jane`	Specifies the user-defined name of the traffic class.
Step 4	**match any** **Example:** `Router(config-cmap)# match any`	Specifies that all packets will be matched. In this release, use only the **any** keyword. Other keywords might cause unexpected results.
Step 5	**policy-map** *policy-name* **Example:** `Router(config-cmap)# policy-map doe`	Specifies the name of the traffic policy to configure.

	Command or Action	Purpose
Step 6	**class** *class-name* **Example:** `Router(config-pmap)# class jane`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 7	**shape average** *bit value* **Example:** `Router(config-pmap-c)# shape average 2000000 8000 8000`	Shapes traffic according to the bit rate you specify.
Step 8	**interface***slot/port* `Router(config)# interface atm4/0`	Enters the interface and enters interface configuration mode.
Step 9	**service-policy input** *policy-name* **Example:** `Router(config-if)# service-policy input doe`	Attaches a traffic policy to an interface.

Note You can enable traffic shaping and set experimental bits in the same policy-map.

Note EoMPLS VLAN Policing Exclusion—traffic on the EoMPLS uplink port is excluded from a VLAN-based ingress policer.

To display the traffic policy attached to an interface, use the **show policy-map interface** command.

EoMPLS QoS Example

If the egress MPLS tunnel is carried on an OSM WAN interface configured for fair queuing, the shape value is rounded to the nearest multiple of the link rate divided by 255. If the shape value is lower than the link rate divided by 255, it is rounded up to link rate divided by 255.

This example shows how the shape value is rounded:

```
Router# show pol p2
 Policy Map p2
  class  any-pkt
   shape average 2000000 8000 8000

Router# show pol int

 Vlan101

  service-policy input:p2

    class-map:any-pkt (match-all)
      2018169 packets, 4575195376 bytes
      30 second offered rate 295768000 bps, drop rate 0 bps
```

```
    match:any
    queue size 0, queue limit 0
    packets input 40492, packet drops 1977677
    tail/random drops 0, no buffer drops 0, other drops 1977677
    shape:cir 2000000,  Bc 8000,  Be 8000
    (shape parameter is rounded to 2439000 due to granularity)
    input bytes 40847436, shape rate 1874000 bps

  class-map:class-default (match-any)
    0 packets, 0 bytes
    30 second offered rate 0 bps, drop rate 0 bps
    match:any
      0 packets, 0 bytes
      30 second rate 0 bps
```

EoMPLS QoS Example—Displaying the Traffic Policy Assigned to an Interface

To display the traffic policy attached to an interface, issue the following command:

```
Router# show policy-map vlan50
service-policy input: badger

    class-map: blue (match-all)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      match: any
      queue size 0, queue limit 2
      packets input 0, packet drops 0
      tail/random drops 0, no buffer drops 0, other drops 0
      shape: cir 2000000,  Bc 8000,  Be 8000
        output bytes 0, shape rate 0 bps

    class-map: class-default (match-any)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      match: any
        0 packets, 0 bytes
        30 second rate 0 bps
```

EoMPLS QoS Example— Configuring QoS on VLAN

The following example show how to configure QoS on the VLAN.

```
class-map blue
match cos 1 2 3
!
policy-map badger
class blue
set mpls experimental 1
class class-default
shape average 2000000 8000 8000
!
interface vlan50
no ip address
no ip mroute-cache
load-interval 30
mpls l2transport route 192.168.255.255 50
service-policy input badger
no cdp enable
```

ATMoMPLS QoS Example—Configuring Ingress QoS

This example shows ingress QoS. On this side the policy attaches to a multipoint Layer 2 transport PVC. In this configuration the EXP bits are set to 5 for all packets on PVC 1/101.

```
class-map match-any anyclass
     match any
!
policy-map set-policy
  class anyclass
    set mpls experimental 5

interface ATM6/0/0
 no ip address
 logging event link-status
 atm clock INTERNAL
 pvc 1/101 l2transport
   encapsulation aal5
   mpls l2transport route 10.10.10.10 101
   service-policy input set-policy
```

For input shaping, the configuration is same as above but the action in the policy-map should be changed to shape.

```
policy-map shape-policy
  class anyclass
    shape average 16000 3200 3200

For output shaping based on MPLS EXP, the policy is configured on the main interface.
class-map match-any exp2
  match mpls experimental 2
!
policy-map shape-policy
  class exp2
    shape average 16000 32 32

interface POS4/1
ip address 20.1.1.1 255.255.255.0
 service-policy output shape-policy
 no ip mroute-cache
 load-interval 30
 no keepalive
 mpls label protocol ldp
 tag-switching ip
 mls qos trust dscp
 clock source internal
 no cdp enable
```

FRoMPLS QoS Example —Configuring Ingress QoS

On the ingress side, you attach the policy to a switched frame-relay DLCI. The configuration below matches frame relay packets with DLCI 10 and sets the EXP bits to 5 during label imposition for the matched packets.

```
class-map match-any anyclass
  match any
!
!
policy-map set-policy
  class anyclass
```

```
    set mpls experimental 5

map-class frame-relay dlci101
 service-policy input set-policy

interface POS1/1
 no ip address
 encapsulation frame-relay IETF
 no ip mroute-cache
 load-interval 30
 no keepalive
 mls qos trust dscp
 clock source internal
 frame-relay interface-dlci 101 switched
  class dlci101
```

For input shaping, the configuration is same as above but the action in the policy-map is changed to shape as follows:

```
policy-map shape-policy
  class anyclass
    shape average 1600000 6400 6400
```

For output shaping based on MPLS EXP, the policy is configured on the main interface.

```
class-map match-any exp2
  match mpls experimental 2
!
policy-map shape-policy
  class exp2
    shape average 1600000 6400 6400

interface POS4/1
ip address 20.1.1.1 255.255.255.0
 service-policy output shape-policy
 no ip mroute-cache
 load-interval 30
 no keepalive
 mpls label protocol ldp
 tag-switching ip
 mls qos trust dscp
 clock source internal
 no cdp enable
```

For WRED based on MPLS EXP, configure the policy on the main interface.

```
class-map match-any exp2
  match mpls experimental 2
!
policy-map wred-pol
  class exp2
    bandwidth percent 20
    random-detect

interface POS4/1
  service-policy output wred-pol
```

HQoS for EoMPLS Virtual Circuits

The Hierarchical Quality of Service (HQoS) for Ethernet over MPLS (EoMPLS) Virtual Circuits (VCs) feature enables hierarchical QoS services on WAN-based interfaces, allowing service providers to classify the traffic in customer EoMPLS networks before it is forwarded into the core network. This gives users of Cisco Catalyst 6500 series switches and Cisco 7600 series routers greater flexibility in providing QoS services to specific customers in their EoMPLS networks.

The HQoS for EoMPLS VCs feature allows you to classify EoMPLS networks in the following ways:

- Match on the VLAN ID that the packet contained when it was originally received at the input interface. You can match a single VLAN ID, a range of VLAN IDs, or a combination of the two, allowing you to match all or part of an EoMPLS network.
- Match on a QoS group value that is set to the same value of the IP precedence or CoS bits that are received with the packet at the input interface.

The use of hierarchical policy maps can simplify the configuration of the router, because the same child policy map can be used in multiple parent maps. You can also match multiple VLANs with one class map, as opposed to having separate class maps for each VLAN.

The HQoS for EoMPLS VCs feature does not require any upgrades to the customer-facing interfaces, because the HQoS policy map is applied to the WAN interface, allowing the customer-facing interfaces to be standard Ethernet interfaces.

Prerequisites for the HQoS for EoMPLS VCs Feature

- You must enable QoS on the router before using HQoS. To enable QoS globally on the router, use the **mls qos** command in global configuration mode. To enable QoS on an individual interface, use the **mls qos** interface configuration command. In addition, the **mls trust** command must be configured on the CE facing PE interfaces.

Restrictions for the HQoS for EoMPLS VCs Feature

The following section lists restrictions for the HQoS for EoMPLS VCs feature. Other restrictions may also apply to QoS services in general, depending on the supervisor module and line cards being used.

Note The HQoS for EoMPLS VCs feature is supported only on PXF- based QoS configured on switched virtual interfaces (SVIs).

- If a policy contains a class map with a **match input vlan** command, you cannot attach that policy map to an interface if you have already attached a service policy to a VLAN interface (a logical interface that has been created with the **interface vlan** command).

  Note This restriction means that **match input vlan** configurations and **interface vlan** configurations are mutually exclusive.

- The HQoS for EoMPLS VCs feature is supported only for output (egress) interfaces (policy maps must be attached to the interface using the **service-policy output** command).
- The HQoS for EoMPLS VCs feature supports only point-to-point VCs, not point-to-multipoint VCs.

- If the parent class contains a class map with a **match input vlan** command, you cannot use a **match exp** command in a child policy map.
- You cannot attach a child policy map to the parent class default.
- Child and parent policy maps do not support any marking, such as the **match ip dscp** and **set** commands.
- The HQoS for EoMPLS VCs feature does not support multiple levels of parent and child policy map nesting. Each parent policy map supports only one level of nesting. In other words, a traffic class in a parent policy map can have a maximum of one child policy map, and child policy maps cannot have their own child policy maps.

Note You can mix flat traffic classes (that do not refer to child policy maps) and hierarchical traffic classes (that do refer to child policy maps) in the same HQoS parent policy maps.

- You cannot apply both HQoS output policy on a main interface (using the **service-policy output** command) and an output policy (**service-policy output** command) on a subinterface of that same interface. If you attempt to do so, then attaching the HQoS output policy fails with the following error message:

```
Attaching service policy to main and sub-interface concurrently is not allowed
```

- In Cisco IOS Release 12.2(18)SXE and later releases, policy maps can contain a maximum of 255 class maps.
- Child policy maps support only strict priority (the **priority** command without any options). Parent policy maps do not support any form of the **priority** command.
- When using both the **priority** and **police** commands in more than one class in a child priority map, you must configure the commands in the following order:
  - In the first class to be configured on the priority map, specify the **priority** command first, and then the **police** command.
  - In the second and any additional classes to be configured on the priority map, specify the **police** command first, and then the **priority** command.
  - The **police cir** command is supported only on OSM interfaces.

Note The **priority** command can be configured only with the **police** command. You cannot use priority together with any forms of the **bandwidth** or **shape** commands.

- Class maps that use the **match input vlan** command support only the **match-any** option. You cannot use the **match-all** option in class maps that use the **match input vlan** command.
- Classes using the the **match input vlan** command should always be placed first in the policy maps, before any classes that use flat policies.
- Parent policy maps do not support the **fair-queue** command. Also, the **fair-queue** command is not supported for OSM interfaces.
- You must use class-default for the input service policy on a CE-PE interface that uses the **qos-group** command to set CoS or IP-Precedence.
- Service policies cannot be attached to subinterfaces for OSM interfaces.
- OSM interfaces support only the **shape average** command. Other forms of the **shape** command are not supported on OSM interfaces.

- The **bandwidth remaining precent** command is not supported on any OSM interfaces. However, the following OSMs support the **bandwidth** command in a parent class under a hierarchical policy map:
  - OSM-2+4GE-WAN-GBIC+
  - OSM-4OC3-POS-SI+
  - OSM-8OC3-POS-SI+
  - OSM-8OC3-POS-SL+

Note For the **bandwidth** command, the minimum rate and the granularity are 1/255 of the bandwidth.

Note For additional prerequisites and restrictions for HQoS in general, see the section "Configuring Hierarchical Traffic Shaping*"* at "Configuring Hierarchical Traffic Shaping" section on page 1-15.

Supported Features

The HQoS for EoMPLS VCs feature supports the following commands on the class maps and policy maps for output interfaces.

The following are supported on parent policy maps:

- **bandwidth**—Egress class-based weighted fair queuing (CBWFQ) (supported on parent policy maps only on OC-3 and OC-12 POS OSM interfaces, and on OSM-2+4GE-WAN-GBIC+ interfaces)
- **shape average**—Egress shaping

The following are supported on child policy maps:

- **bandwidth**—Egress class-based weighted fair queuing (CBWFQ)
- **priority**—Egress low latency queuing (LLQ) (Only strict priority is supported on child maps and on OSMs.)

  Note Strict priority is supported for OC-3 and OC-12 POS OSM and OSM-2+4GE-WAN-GBIC+ interfaces only.

- **queue-limit**—Queue throttling
- **random-detect**—Egress weighted random early detection (WRED)
- **shape average**—Egress shaping

Related Commands

Do not confuse the **match input vlan** command with the **match vlan** command, which is also a class-map configuration command.

- The **match vlan** command matches the VLAN ID on packets for the particular interface at which the policy map is applied. Policy maps using the **match vlan** command can be applied to either ingress or egress interfaces on the router, using the **service-policy** {**input** | **output**} command.

- The **match input vlan** command matches the VLAN ID that was on packets when they were received on the ingress interface on the router. Policy maps using the **match input vlan** command must be applied to egress interfaces on the router, using the **service-policy output** command.

The **match input vlan** command can also be confused with the **match input-interface vlan** command, which matches packets being received on a logical VLAN interface that is used for inter-VLAN routing.

Tip Because class maps also support the **match input-interface** command, you cannot abbreviate the **input** keyword when giving the **match input vlan** command.

Configuring the HQoS for EoMPLS VCs Feature

To use a hierarchical QoS policy map for EoMPLS traffic, you must perform the following tasks. (All tasks are required.)

- Apply a policy map to the input interface to set the QoS group value on incoming packets. See the "Creating and Assigning a Policy Map to Mark the QoS Group at the Incoming Interface" section on page 11-93.
- Create class maps that match packets on the basis of their QoS group values. See the "Configuring the Class Map to Match on a QoS Group" section on page 11-96.
- Create a child policy map that uses these class maps. See the "Creating the Child Policy Map for the Egress Interface" section on page 11-98.
- Create class maps that match packets on the basis of their input VLAN IDs. See the "Configuring the Class Maps for Matching on an Input VLAN" section on page 11-102.
- Create a parent policy map and apply it to the output interface. See the "Creating the Parent Policy Map and Attaching It to the Egress Interface" section on page 11-104.

Note For more information about hierarchical traffic shaping, see the section "Configuring Hierarchical Traffic Shaping" at "Configuring Hierarchical Traffic Shaping" section on page 1-15.

Creating and Assigning a Policy Map to Mark the QoS Group at the Incoming Interface

To be able to classify traffic on a QoS group, you must first create a policy map that marks incoming packets with the desired QoS group value. You can set the QoS group value to the value of either the IP precedence bits or 802.1P CoS bits of the incoming packets. You then must assign that policy map to the incoming interface (which must be a Layer 2 LAN interface). To perform these tasks, use the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **description** *string*
5. **class class-default**
6. **set qos-group** {**cos** | **ip-precedence**}

7. **interface** *if-type* {*slot/port* | *slot/subslot/port*}
8. **service-policy input** *policy-map-name*
9. **end**
10. **show policy-map**
    show policy-map *policy-map-name* [**class** *class-map*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map cos-to-qosgrp-pmap`	Creates a policy map with the specified name and enters policy-map configuration mode. • *policy-map-name*—Name of the policy map. The name must be a unique string of up to 40 alphanumeric characters.
Step 4	**description** *string* **Example:** `Router(config-pmap)# description Sets QoS group to 802.1P CoS of incoming packets`	(Optional) Arbitrary string, up to 200 characters long, that describes this policy map.
Step 5	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Specifies the default class to be used for traffic with this policy, and enters policy-map class configuration mode.

	Command or Action	Purpose
Step 6	**set qos-group** {**cos** | **ip-precedence**} **Example:** `Router(config-pmap-c)# set qos-group cos`	Sets a quality of service (QoS) group identifier (ID) that can be used later to classify packets. • **cos**—Sets the packet's QoS group value to the same value as the packet's original 802.1P Class of Service (CoS) bits. • **ip-precedence**—Sets the packet's QoS group value to the same value as the packet's original IP precedence bits. **Note** The **set qos-group** command also supports setting the QoS group to an arbitrary value from 0 to 99, but this configuration is not supported when using the HQoS for EoMPLS VCs feature. This command also supports the option of specifying a table map, but the HQoS for EoMPLS VCs feature does not support this option, because it always uses the default mappings.
Step 7	**interface** *if-type* {*slot/port* | *slot/subslot/port*} **Example:** `Router(config-pmap-c)# interface GigabitEthernet 5/2`	Enters interface configuration mode for the incoming interface. **Note** This interface must be a Layer 2 LAN interface. It cannot be a Layer 3 WAN interface.
Step 8	**service-policy input** *policy-map-name* **Example:** `Router(config-if)# service-policy input cos-to-qosgrp-pmap`	Attaches the specified policy map to the interface for input (ingress) traffic. • *policy-map-name*—Name of the policy map that was created in Step 3.
	Note Repeat Step 7 and Step 8 for each interface that should be marking the QoS group value on incoming traffic.	
Step 9	**show policy-map** **show policy-map** *policy-map-name* [**class** *class-map*] **Example:** `Router# show policy-map cos-to-qosgrp-pmap`	(Optional) Displays the configured class map to verify the configuration. To display all policy maps, enter the command without any options. To display a specific policy map, specify its name on the command line. You can also display a specific class that is part of a specific policy map by adding the **class** option.

The following policy map sets the QoS group value to match the CoS value of the incoming packets. The policy map is then assigned to two interfaces:

```
policy-map cos-to-qosgroup-pmap
   class class-default
      set qos-group cos
...
!
interface GE 6/0
 service-policy input cos-to-qosgroup-pmap
...
!
interface GE 6/1
 service-policy input cos-to-qosgroup-pmap
...
```

What to Do Next

After attaching the policy map to the input interface, create the class map to match on the QoS group value at the egress (outgoing) interface. See the "Configuring the Class Map to Match on a QoS Group" section on page 11-96 for details.

Configuring the Class Map to Match on a QoS Group

To be able to match EoMPLS traffic using QoS groups, you must create class maps to match traffic on the basis of the QoS group value at the egress (outgoing) interface. To create these class maps, use the following procedure.

Prerequisites

- You must create policy maps that contain class maps that use the **set qos-group** command to mark incoming packets with the desired QoS group values. Then attach those policy maps to the input interfaces that are receiving the incoming traffic. See the "Creating and Assigning a Policy Map to Mark the QoS Group at the Incoming Interface" section on page 11-93.
- Input interfaces must also be configured with **mls trust**.

Restrictions

- A policy map that refers to a class map that uses the **match qos-group** command cannot have other class maps that match on the following commands:
  - **match ip prec match**
  - **match mpls exp**
- The allowable range of values for QoS groups is from 0 to 99. The only valid values for EoMPLS traffic are from 0 to 7. This is because the QoS group value is set to the IP precedence or CoS fields in the incoming packets, and both of these fields are only 3-bit values that can range from 0 to 7.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match qos-group** *qos-group-value*
5. **end**
6. **show class-map** *class-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Router(config)# class-map group4`	Creates a class map and enters class-map configuration mode. • **match-all**—(Optional) All match criteria must be matched for a packet to be matched by this class map. This is the default if no option is specified. • **match-any**—(Optional) Only one match criterion must be matched for a packet to be matched by this class map. • *class-map-name*—Arbitrary string that identifies this class map.
Step 4	**match qos-group** *qos-group-value* **Example:** `Router(config-cmap)# match qos-group 4`	Matches packets with the specified QoS group marking. • *qos-group-value*—Specifies the QoS group value to be matched. The allowable range is from 0 to 99, but for EoMPLS traffic, the only valid values are from 0 to 7, because the QoS group value is set to the value of the IP precedence or CoS bits in the incoming packets.
Step 5	**end** **Example:** `Router(config-cmap)# end`	Exits class-map configuration mode and returns to privileged EXEC mode.
Step 6	**show class-map** *class-map-name* **Example:** `Router# show class-map group4`	(Optional) Displays the configured class map to verify the configuration.

The following configuration example shows all of the class maps that are allowed for matching on QoS groups for EoMPLS traffic.

```
class-map match-all group0
  match qos-group 0
class-map match-all group1
  match qos-group 1
class-map match-all group2
  match qos-group 2
class-map match-all group3
  match qos-group 3
class-map match-all group4
  match qos-group 4
class-map match-all group5
```

```
  match qos-group 5
class-map match-all group6
  match qos-group 6
class-map match-all group7
  match qos-group 7
```

What to Do Next

After creating all of the desired class maps, you must include them in a child policy map. See the next section, "Creating the Child Policy Map for the Egress Interface," for more information.

Creating the Child Policy Map for the Egress Interface

A hierarchical policy map is identical to the flat policy maps that were supported in earlier Cisco IOS software releases, except that at least one of the traffic class maps in the parent policy map refers to a child policy map. You must create the child policy maps before creating the parent policy maps.

To create a child policy map, use the following procedure. Repeat as needed to create the desired number of child policy maps.

Tip Different parent policy maps can use the same child policy maps, if desired.

Prerequisites

- You must first create the class maps to be used by this policy map. See the "Configuring the Class Map to Match on a QoS Group" section on page 11-96.

Restrictions

Child policy maps for EoMPLS traffic have the following restrictions:

- The **set** command is not supported on the child policy map.
- Child policy maps support only strict priority (the **priority** command without any options). Parent policy maps do not support any form of the **priority** command.
- When using both the **priority** and **police** commands in more than one class in a priority map, you must configure the commands in the following order:
  - In the first class to be configured on the priority map, specify the **priority** command first, and then the **police** command.
  - In the second and any additional classes to be configured on the priority map, specify the **police** command first, and then the **priority** command.
- You cannot use the **service-policy** *child-pmap-name* command in child policy maps, because multi-level nesting is not supported for HQoS for EoMPLS VCs policy maps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *child-pmap-name*
4. **description** *string*

5. **class** {*class-map-name* | **class-default**}

Note Each class action below must be preceded by a **class** command.

6. **shape** {**average**} *mean-rate*
7. **class** {*class-map-name* | **class-default**}
8. **priority**
9. **police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*]
10. **class** {*class-map-name* | **class-default**}
11. **bandwidth** {*bandwidth-kbps* | **remaining percent** *percentage* | **percent** *percentage*}
12. **end**
13. **show policy-map** *child-pmap-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *child-pmap-name* **Example:** `Router(config)# policy-map child-pmap-name`	Creates a policy map with the specified name, for use as a child policy map, and enters policy-map configuration mode. • *child-pmap-name*—Name of the child policy map. The name must be a unique string of up to 40 alphanumeric characters.
Step 4	**description** *string* **Example:** `Router(config-pmap)# description Child policy map for input VLAN parent class`	(Optional) Arbitrary string, up to 200 characters long, that describes this policy map.

Step 5	**class** {*class-map-name*	**class-default**}
		
		Example:
		`Router(config-pmap)# class qosgroup4`
		`Router(config-pmap-c)#`
		`or`
		`Router(config-pmap)# class class-default`
			Specifies the name of a class map that should be used with this policy, and enters policy-map class configuration mode.
			• *class-map-name*—Name of the class map to be used. This should be a class map that was created using the **class-map** command in previous configuration tasks.
			• **class-default**—Specifies the default class that should be used for this policy for unclassified traffic that does not match the other class maps for this policy.

	Command or Action	Purpose
Step 6	**shape** {**average**} *mean-rate* **Example:** `Router(config-pmap-c)# shape average 10000000`	(Optional) Shapes the traffic in this class by the limits specified. • **average**—Limits traffic to the maximum bit rate that is specified by the *mean-rate* parameter. • *mean-rate*—Maximum number of bits to transmitted, in bits per second. Also called the Committed Information Rate (CIR). The valid range is from 8000 to 4,000,000,000 bits per second, with no default.

Step 7	**class** {*class-map-name* | **class-default**}
	
	Example:
	`Router(config-pmap)# class qosgroup5`
	`or`
	`Router(config-pmap)# class class-default`
		Specifies the name of a class map that should be used with this policy, and enters policy-map class configuration mode.
		• *class-map-name*—Name of the class map to be used. This should be a class map that was created using the **class-map** command in previous configuration tasks.
		• **class-default**—Specifies the default class that should be used for this policy for unclassified traffic that does not match the other class maps for this policy.

Step 8	**priority** **Example:** `Router(config-pmap-c)# priority`	(Optional) Specifies that traffic in this class is priority traffic. **Note** You cannot configure both the **shape** and the **priority** commands in the same class.

Note When using both the **priority** and **police** commands in a class, you must configure them in the following order: In the first class to be configured on the priority map, specify the **priority** command first, and then the **police** command. In the second and any additional classes to be configured on the priority map, specify the **police** command first, and then the **priority** command.

Step 9	**police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*]
	
	Example:
	`Router(config-pmap-c)# police 8000 1000`
	`conform-action transmit exceed-action drop`
		(Optional) Specifies the policing policy that should be used for traffic in this class.
		• *bps*—Average rate in bits per second. The valid range is from 8,000 to 200,000,000.
		• *burst-normal*—(Optional) The normal maximum burst size in bytes. The valid range is from 1,000 to 51,200,000 bytes, with a default value of 1,500 bytes.
		• *burst-max*—(Optional) Excess burst size in bytes. The valid range is from 1,000 to 51,200,000.
		• **conform-action**—Specifies the action to take for packets that are within the specified rate limit.
		• **exceed-action**—Specifies the action to take for packets that exceed the specified rate limit.
		• **violate-action**—(Optional) Specifies the action to take for packets that violate the normal and maximum burst sizes.
		• *action*—Action to be taken for the specified condition. The most common values are **drop** (drop the packet) or **transmit** (transmits the packet without change). Additional values are possible for setting different class of service (CoS) parameters.

	Command or Action	Purpose

Step 10	**class** {*class-map-name* | **class-default**}
	
	Example:
	`Router(config-pmap)# class qosgroup6`
	`or`
	`Router(config-pmap)# class class-default`
		Specifies the name of a class map that should be used with this policy, and enters policy-map class configuration mode.
		• *class-map-name*—Name of the class map to be used. This should be a class map that was created using the **class-map** command in previous configuration tasks.
		• **class-default**—Specifies the default class that should be used for this policy for unclassified traffic that does not match the other class maps for this policy.

Step 11	**bandwidth** {*bandwidth-kbps* | **remaining percent** *percentage* | **percent** *percentage*} **Example:** `Router(config-pmap-c)# bandwidth percent 50`	(Optional) Specifies the bandwidth that is allowed for traffic in this class. • *bandwidth-kbps*—Amount of bandwidth, in kbps, to be assigned to the class. The valid range is from 1 to 2,000,000, but the allowable values vary according to the interface and platform in use. • **remaining percent**—Amount of guaranteed bandwidth, based on a relative percent of available bandwidth. The valid range for *percentage* is from 1 to 100. • **percent**—Amount of guaranteed bandwidth, based on an absolute percent of available bandwidth. The valid range for *percentage* is from 1 to 100. **Note** Versions of Cisco IOS software before Cisco IOS Release 12.2(18)SXE did not support the **bandwidth** command in parent policy maps. This restriction was removed in Cisco IOS Release 12.2(18)SXE and later releases for OC-3 and OC-12 POS OSM and OSM-2+4GE-WAN-GBIC+ interfaces only.
	Note Repeat Step 10 through Step 11 for each class to be used in this child policy map.	
Step 12	**end** **Example:** `Router(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.

Step 13	**show policy-map**
	show policy-map *child-pmap-name* [**class** *class-map*]
	
	Example:
	`Router# show policy-map child-policy1`
	`(command output)`
		(Optional) Displays the configured policy map to verify the configuration. To display all policy maps, enter the command without any options. To display a specific policy map, specify its name on the command line. You can also display a specific class that is part of a specific policy map by adding the **class** option.

The following sample configuration shows a typical child policy map that refers to two of the QoS group class maps that were defined in the "Configuring the Class Map to Match on a QoS Group" section on page 11-96.

```
policy-map child
! Class for QoS Group 3 performs LLQ
 class group3
```

```
  priority
  police 20000000 625000 625000 conform-action transmit exceed-action drop
! Class for QoS Group 4 performs CBWFQ when bandwidth usage is at 30 percent
 class group4
  bandwidth percent 30
```

Note When using both the **priority** and **police** commands in a class, you must configure them in the following order: In the first class to be configured on the priority map, specify the **priority** command first, and then the **police** command. In the second and any additional classes to be configured on the priority map, specify the **police** command first, and then the **priority** command.

What to Do Next

After creating the child policy map, you must create the parent policy map. See the "Creating the Parent Policy Map and Attaching It to the Egress Interface" section on page 11-104 for details.

Configuring the Class Maps for Matching on an Input VLAN

To match EoMPLS packets that are tagged with one or more specific VLAN IDs, you must create a class map that matches on those VLAN IDs. To do this, use the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map match-any** *class-map-name*
4. **match input vlan** *input-vlan-list*
5. **end**
6. **show class-map** *class-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map match-any** *class-map-name* **Example:** `Router(config)# class-map vlan-map`	Creates a class map and enters class-map configuration mode. • *class-map-name*—Arbitrary string that identifies this class map. **Note** Class maps that use the **match input vlan** command support only the **match-any** option. You cannot use the **match-all** option in these class maps.

	Command or Action	Purpose
Step 4	**match input vlan** *input-vlan-list* **Example:** `Router(config-cmap)# match input vlan 10 20 30 100-1999`	Matches packets that are tagged with a VLAN ID specified in the *input-vlan-list*, which can be one or both of the following: • Single VLAN IDs, separated by spaces. The valid range is 0 to 4094. • One or more ranges of VLAN IDs, separated by spaces. The allowable values are between 0 and 4094. **Note** Repeat this command, if desired, to specify additional VLANs. If you use multiple **match input vlan** commands, be sure to use the **match-any** keyword in Step 3 so that the class map can match on any of the VLAN IDs.
Step 5	**end** **Example:** `Router(config-cmap)# end`	Exits class-map configuration mode and returns to privileged EXEC mode.
Step 6	**show class-map** *class-map-name* **Example:** `Router# show class-map vlan-map`	(Optional) Displays the configured class map to verify the configuration.

The following configuration example shows a number of class maps that match either one specific VLAN ID, or a range of VLAN IDs. The last class map matches all valid VLAN IDs.

```
class-map match-any vlan1
  match input vlan 1
class-map match-any vlan2
  match input vlan 2
class-map match-any vlan3
  match input vlan 3
class-map match-any vlan4
  match input vlan 4
class-map match-any vlans1-4
  match input vlan 1-4
class-map match-any vlans-all
  match input vlan 1-4094
```

The following sample configuration shows multiple **match input vlan** commands being used in the traffic class map.

```
class-map match-any vlans-even
  match input vlan 2 4 6 8
  match input vlan 102 104 106 108
  match input vlan 202 204 206 208
```

What to Do Next

After creating all desired class maps, you must then create the parent policy map and assign it to the egress interface. See the next section, ""Creating the Parent Policy Map and Attaching It to the Egress Interface" section on page 11-104," for details.

Creating the Parent Policy Map and Attaching It to the Egress Interface

After creating the class maps and child policy maps, you must create a parent policy map and attach it to the appropriate egress (output) interface. To create and attach a parent policy map, use the following procedure. Repeat as needed to create the desired number of parent policy maps.

Prerequisites

Create at least one child policy map to be used in this parent policy map. See the "Creating the Child Policy Map for the Egress Interface" section on page 11-98 for details. (Different parent policies can use the same child policy maps, if desired.)

Restrictions

Parent policy maps have the following restrictions:

- You cannot attach a policy with the **match input vlan command** to an interface if you have already attached a service policy to its VLAN interface (a logical interface that has been created with the **interface vlan** command). If you attempt to do so, you must then remove both types of policy maps from all interfaces, and then reattach only one type of policy map to the interfaces.
- The **priority** and **fair-queue** commands are not supported in parent policy maps.
- Only the **shape** command and the **bandwidth** command are supported in parent classes; other actions are not supported.
- The **bandwidth** command is supported on parent policy maps only on OC-3 and OC-12 POS OSM interfaces, and on OSM-2+4GE-WAN-GBIC+ interfaces.

Note Versions of Cisco IOS software before Cisco IOS Release 12.2(18)SXE did not support the **bandwidth** command in parent policy maps when using HQoS configurations. This restriction no longer exists in Cisco IOS Release 12.2(18)SXE and later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *parent-pmap-name*
4. **description** *string*
5. **class** {*class-map-name*}
6. **shape** {**average** | **peak**} *mean-rate* [*Bc* [*Be*]]
7. **bandwidth** {*bandwidth-kbps* | **percent** *percentage*}
8. **service-policy** *child-pmap-name*
9. **interface** *if-type* {*slot/port* | *slot/subslot/port*}
10. **service-policy output** *parent-pmap-name*
11. **end**
12. **show policy-map** *parent-pmap-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *parent-pmap-name* **Example:** `Router(config)# policy-map parent-policy1`	Creates a policy map with the specified name, for use as a parent policy map, and enters policy-map configuration mode. • *parent-pmap-name*—Name of the parent policy map. The name must be a unique string of up to 40 alphanumeric characters.
Step 4	**description** *string* **Example:** `Router(config-pmap)# description Parent Policy Map`	(Optional) Arbitrary string, up to 200 characters long, that describes this policy map.

Step 5	**class** {*class-map-name*}
	
	Example:
	`Router(config-pmap)# class vlan100`
	`or`
	`Router(config-pmap)# class class-default`
		Specifies the name of a class-map that should be used with this policy, and enters policy-map class configuration mode.
		• *class-map-name*—Name of the class map to be used. This should be a class map that was created using the **class-map** command in the "Configuring the Class Maps for Matching on an Input VLAN" section on page 11-102.

Step 6	**shape** {**average**} *mean-rate*] **Example:** `Router(config-pmap-c)# shape average 10000000`	(Optional) Shapes the traffic in this class by the limits specified. • **average**—Limits traffic to the maximum bit rate that is specified by the *mean-rate* parameter. • *mean-rate*—Maximum number of bits to transmitted, in bits per second. Also called the Committed Information Rate (CIR). The valid range is from 8,000 to 4,000,000,000 bits per second, with no default.

	Command or Action	Purpose
Step 7	**bandwidth** {*bandwidth-kbps* | **percent** *percentage*} **Example:** `Router(config-pmap-c)# bandwidth percent 50`	(Optional) Specifies the bandwidth that is allowed for traffic in this class. • *bandwidth-kbps*—Amount of bandwidth, in kbps, to be assigned to the class. The valid range is from 1 to 2,000,000, but the allowable values vary according to the interface and platform in use. • **percent**—Amount of guaranteed bandwidth, based on an absolute percent of available bandwidth. The valid range for *percentage* is from 1 to 100. **Note** Versions of Cisco IOS software before Cisco IOS Release 12.2(18)SXE did not support the **bandwidth** command in parent policy maps. This restriction was removed in Cisco IOS Release 12.2(18)SXE and later releases for OC-3 and OC-12 OSM POS, and OSM-2+4GE-WAN-GBIC+ interfaces only.
Step 8	**service-policy** *child-pmap-name* **Example:** `Router(config-pmap-c)# service-policy child-pmap-name`	Specifies a child policy map that should be applied to the traffic in this class: • *child-pmap-name*—Name of a child policy map that was created previously in the "Creating the Child Policy Map for the Egress Interface" section on page 11-98. (The child policy map cannot be another parent policy map—that is, it cannot be a policy map that also uses the **service-policy** command.)
	Note Repeat Step 5 through Step 8 for each class to be used to match VLANs in this parent policy map.	
Step 9	**interface** *if-type* {*slot/port* | *slot/subslot/port*}] **Example:** `Router(config)# interface ge-wan 5/2`	Enters interface configuration mode for the specified interface.
Step 10	**service-policy output** *parent-pmap-name* **Example:** `Router(config-pmap)# service-policy output parent-policy1`	Attaches the specified parent policy map to the interface for outgoing traffic. • *parent-pmap-name*—Name of the policy map that was created in Step 3.
Step 11	**end** **Example:** `Router(config-pmap-c)# end`	Exits policy-map class configuration mode and returns to privileged EXEC mode.
Step 12	**show policy-map** **show policy-map** *parent-pmap-name* [**class** *class-map*] **Example:** `Router# show policy-map vlan-map`	(Optional) Displays the configured policy map to verify the configuration. To display all policy maps, enter the command without any options. To display a specific policy map, specify its name on the command line. You can also display a specific class that is part of a specific policy map by adding the **class** option.

The following sample configuration shows a parent policy map that shapes all of the traffic for three VLANs to specific maximum values. Each class in the parent policy map also specifies a child policy map that further shapes the VLAN traffic on the basis of each packet's QoS group value.

```
!
! Class maps to match on QoS groups (to be used in child policy map)
class-map match-all qosgroup0
  match qos-group 0
class-map match-all qosgroup1
  match qos-group 1
class-map match-all qosgroup2
  match qos-group 2
class-map match-all qosgroup3
  match qos-group 3
class-map match-all qosgroup4
  match qos-group 4
class-map match-all qosgroup5
  match qos-group 5
class-map match-all qosgroup6
  match qos-group 6
class-map match-all qosgroup7
  match qos-group 7
!
! Class maps to match on input vlan IDs (to be used in parent policy map)
class-map match-all vlan101
  match input  vlan 101
class-map match-all vlan102
  match input  vlan 102
class-map match-all vlan103
  match input  vlan 103
!
policy-map child-pmap
   description Child policy map to shape on the basis of the QoS group values
   class qosgroup1
      shape average 10000000
   class qosgroup2
      shape average 20000000
   class qosgroup5
      shape average 40000000
   class class-default
      shape average 10000000
!
policy-map parent-pmap
   description Parent pmap that shapes traffic for individual VLANs
   class vlan101
       shape average 70000000
          service-policy child-pmap
   class vlan102
       shape average 80000000
          service-policy child-pmap
   class vlan103
       shape average 90000000
          service-policy child-pmap
   class class-default
       shape average 10000000
```

Configuration Examples for the HQoS for EoMPLS VCs Feature

This section contains the following sample configurations for the HQoS for EoMPLS VCs feature:

- Simple Hierarchical Configuration Example, page 11-108

Simple Hierarchical Configuration Example

The following example shows a simple hierarchical QoS configuration with one parent policy and one child policy. This configuration performs the following:

- The parent policy shapes all outgoing traffic for VLAN 101 on the GE7/1 interface to a total maximum of 90 Mbps.
- The child policy performs LLQ on the VLAN 101 traffic that has the QoS group set to 1, giving it 10 percent of the bandwidth.
- The child policy allocates 10 percent of the bandwidth of the VLAN 101 traffic that has the QoS group set to 2.
- The child policy performs WRED on the remaining VLAN 101 traffic.

```
class-map match-any vlan101
  match input vlan 101
class-map match-all qos1
  match qos-group1
class-map match-all qos-group2
  match mpls experimental topmost 2
!
policy-map child-pmap
 class qos1
  priority
  police percent 10
 class qos-group2
  bandwidth percent 10
 class class-default
  random-detect
policy-map vlan101-pmap
 class vlan101
  shape average 90000000 360000 360000
  service-policy child-pmap

interface GigabitEthernet 7/1
 service-policy output vlan101-pmap
...
```

Complete Hierarchical QoS Example

The following example shows a hierarchical QoS configuration with one parent policy map and two child policy maps. This configuration performs the following:

- The input interface (Gigabit Ethernet 2/2) uses the cos-to-qosgroup-pmap policy map to set the QoS group value of incoming packets to match the packets' original 802.1P CoS values.
- The parent policy map shapes traffic for VLAN 101 and 102 to different bandwidths, and applies separate child policy maps to each. The rest of the traffic on the interface is shaped and made subject to the random-detect method.
- The child policy map for VLAN 101 allocates different bandwidth to traffic for QoS groups 1 and 2, and transmits all other traffic on that VLAN unchanged (subject to the parent policy map's bandwidth limitations).

- The child policy map for VLAN 102 marks traffic with QoS group set to 2 as priority traffic, and limits all other traffic to 40 percent of the bandwidth (subject to the parent policy map's bandwidth limitations).
- The outgoing interface (POS 8/7) attaches the parent policy map (vlan-parent) for outgoing traffic.

```
class-map match-any vlan101
  match input vlan 101
class-map match-any vlan102
  match input vlan 102
class-map match-all group1
  match qos-group 1
class-map match-all group2
  match qos-group 2

!
policy-map cos-to-qosgroup-pmap
   class class-default
      set qos-group cos

!
policy-map vlan-parent
  description top-level parent policy map
  class vlan101
   shape average 50000000 200000 200000
   service-policy 101qos
  class vlan102
   shape average 100000000 400000 400000
   service-policy 102qos
  class class-default
   shape average 50000000 200000 200000
   random-detect
!
policy-map 101qos
  description child-level policy map for VLAN 101
  class group1
    bandwidth percent 10
  class group2
    bandwidth percent 30
policy-map 102qos
  description child-level policy map for VLAN 102
  class group2
   police percent 10
   priority
  class class-default
    bandwidth percent 40

!
! Customer-facing interface - the cos-to-qosgroup-pmap policy map sets the
! packet's QoS group value to match the customer's original CoS values.
interface GigabitEthernet2/2
 description Customer-facing interface
 ip address 192.168.100.13 255.255.255.0
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 10
 switchport trunk allowed vlan 101-1000,1002-1005
 switchport mode trunk
 mls qos trust
 no cdp enable
 service-policy input cos-to-qosgroup-pmap

...
```

```
!
interface POS8/7
 description Network-Facing OSM POS
 ip address 10.11.0.5 255.255.255.0
 encapsulation ppp
 tag-switching ip
 mls qos trust dscp
 service-policy output vlan-parent
...
```

Multiple Parent Policies Using the Same Child Policy Example

This excerpt from a sample configuration file shows several parent policy maps using the same child map.

```
! You can enable QoS globally or per-interface
mls qos
!
class-map match-all group1
  match qos-group 1
class-map match-all group2
  match qos-group 2
class-map match-any vlan101
  match input vlan 101
class-map match-any vlan102
  match input vlan 102
class-map match-any vlan103
  match input vlan 103
class-map match-all exp-3
  match mpls experimental topmost 3
!
policy-map child-pmap
  class group1
    shape average 10000000
  class group2
    shape average 20000000
!
policy-map parent1-pmap
  class vlan101
    shape average 60000000
   service-policy child-pmap
  class vlan102
    shape average 80000000
   service-policy child-pmap
  class class-default
    shape average 100000000
!
policy-map parent2-pmap
  class vlan103
    shape average 55000000
   service-policy child-pmap
  class exp-3
    shape average 60000000
...
```

Common Class-Map Templates Example

This excerpt from a configuration file gives some common templates for class maps that can be used with your own policy maps.

```
! You can enable QoS globally or per-interface
mls qos

...

! Class Maps to Match on IP Precedence Bits
class-map match-any prec0
  match ip precedence 0
class-map match-any prec1
  match ip precedence 1
class-map match-any prec2
  match ip precedence 2
class-map match-any prec3
  match ip precedence 3
class-map match-any prec4
  match ip precedence 4
class-map match-any prec5
  match ip precedence 5
class-map match-any prec6
  match ip precedence 6
class-map match-any prec7
  match ip precedence 7
! Matches all non-priority precedence values
class-map match-any prec0-4
  match ip precedence 0 1 2 3 4
!
! Class-Maps to Match on QoS Groups
class-map match-all group0
  match qos-group 0
class-map match-all group1
  match qos-group 1
class-map match-all group2
  match qos-group 2
class-map match-all group3
  match qos-group 3
class-map match-all group4
  match qos-group 4
class-map match-all group5
  match qos-group 5
class-map match-all group6
  match qos-group 6
class-map match-all group7
  match qos-group 7
!
! Class Maps to Match on MPLS EXP Bits
class-map match-all exp0
  match mpls experimental topmost 0
class-map match-all exp1
  match mpls experimental topmost 1
class-map match-all exp2
  match mpls experimental topmost 2
class-map match-all exp3
  match mpls experimental topmost 3
class-map match-all exp4
  match mpls experimental topmost 4
class-map match-all exp5
  match mpls experimental topmost 5
class-map match-all exp6
  match mpls experimental topmost 6
class-map match-all exp7
  match mpls experimental topmost 7
class-map match-all exp1-4
  match mpls experimental topmost 1  2  3  4
!
```

```
! Sample Class-MapS to Match on VLAN
! Copy and Change the VLAN Number as Desired
class-map match-any vlan101
  match input vlan 101
class-map match-any vlan102
  match input vlan 102
class-map match-any vlan103
  match input vlan 103
class-map match-any vlan104
  match input vlan 104
class-map match-any vlans101-104
  match input vlan 101-104
!
```

AToM Load Balancing

Load-balancing allows a router to take advantage of multiple best paths to a given destination. By default most AToM modes (except SUP720-3BXL-based EoMPLS) use a similar load balancing mechanism to determine the tunnel label for the core facing interface: the router distributes AToM VCs across all available paths, irrespective of each link's load. The router hashes the VC label into an index value that is used to select a tunnel label. The selected tunnel label is placed on the top of the label stack of a particular VC.

The Cisco 7600 series router provides another way to load balance by selecting the path with the lowest use across all available paths based on the following order:

- Different ports on the same packet processor complex
- Different interfaces on a chosen port on the same packet processor complex.

Load Balancing Guidelines

Enable lowest use mode by entering configuration commands (one command per line) and pressing Ctrl-Z after each command.

```
PE-7600B#conf t
PE-7600B(config)#mpls load-balance per-l2transport-circuit
```

Disable lowest use mode by entering configuration commands (one command per line) and pressing Crtl-Z after each command.

```
PE-7600B#conf t
PE-7600B(config)#no mpls load-balance per-l2transport-circuit
```

Display the current load balancing mode using the **show cwan atom load-balance-mode** command.

```
PE-7600B#sh cwan atom load-balance-mode
Current load balancing mode : per-l2transport-circuit
```

Note When the lowest use load balancing mode is enabled on a system that is already up, it only affects newer AToM VCs. Existing AToM VCs are not affected. To apply the lowest use load balancing mode to all the existing VCs, you can flap the VCs.

Lowest Use Mode Limitations

If the interfaces facing the MPLS core are a mix of WAN and LAN interfaces, then the AToM VCs remain active as long as there is a minimum of one usable WAN interface. However, this is not a recommended setup and the AToM VC may be dropping disposition packets that arrive on the LAN interface.

If you ignore the warning message that indicates this type of configuration, you risk losing disposition packets because the AToM VC may not be fully functioning.

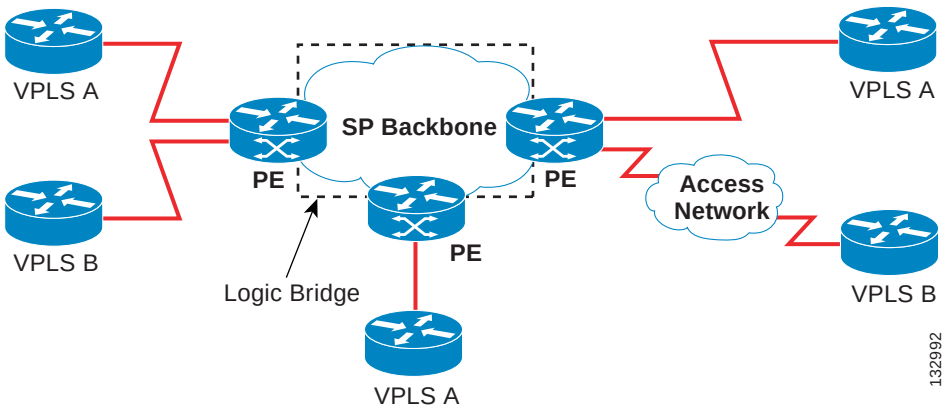
Virtual Private LAN Services on the Optical Services Modules

This section describes how to configure Virtual Private LAN Services (VPLS) on the Optical Services Modules (OSMs) and covers the topics below.

VPLS Overview

Virtual Private LAN Services (VPLS) uses the provider core to join multiple attachment circuits together to simulate a virtual bridge that connects the multiple attachment circuits together. From a customer point of view, there is no topology for VPLS. All of the CE devices appear to connect to a logical bridge emulated by the provider core. See Figure 11-4.

Figure 1-4 VPLS



Full-mesh 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.

The full-mesh configuration allows the PE router to maintain a single broadcast domain. Thus, when the PE router receives a broadcast, multicast, or unknown unicast packet on an attachment circuit, it sends the packet out on all other attachment circuits and emulated circuits to all other CE devices participating in that VPLS instance. The CE devices see the VPLS instance as an emulated LAN.

To avoid the problem of a packet looping in the provider core, the PE devices enforce a "split-horizon" principle for the emulated VCs. That means if a packet is received on an emulated VC, it is not forwarded on any other emulated VC.

After the VFI has been defined, it needs to be bound to an attachment circuit to the CE device.

The packet forwarding decision is made by looking up the Layer 2 virtual forwarding instance (VFI) of a particular VPLS domain.

A VPLS instance on a particular PE router receives Ethernet frames that enter on specific physical or logical ports and populates a MAC table similarly to how an Ethernet switch works. The PE router can use the MAC address to switch those frames into the appropriate LSP for delivery to the another PE router at a remote site.

If the MAC address is not in the MAC address table, the PE router replicates the Ethernet frame and floods it to all logical ports associated with that VPLS instance, except the ingress port where it just entered. The PE router updates the MAC table as it receives packets on specific ports and removes addresses not used for specific periods.

H-VPLS

Hierarchical VPLS (H-VPLS) reduces both signaling and replication overhead by using both full-mesh as well as hub and spoke configurations. Hub and spoke configurations operate with split horizon to allow packets to be switched between pseudo-wires (PWs), effectively reducing the number of PWs between PEs.

Note Split horizon is the default configuration to avoid broadcast packet looping. To avoid looping when using the **no-split-horizon** keyword, be very mindful of your network configuration.

Restrictions for VPLS

The following general restrictions pertain to all transport types under VPLS:

- Split horizon is the default configuration to avoid broadcast packet looping and to isolate Layer 2 traffic. With split horizon, a packet coming from a WAN interface never goes back to another WAN interface (it always get switched to a Layer 2 interface). Split horizon prevents packets received from an emulated VC from being forwarded into another emulated VC. This technique is important for creating loop-free paths in a full-meshed network.
- The Cisco 7600 series routers support a maximum of 60 peer PEs and a maximum of 15,000 VCs. For example, you can configure 15,000 VCs as 1,000 VFIs with 15 VPLS peers per VFI.

Note The 60 peer PEs are distributed between the MPLS edge and the core; do not assume there are 60 peer PEs on each side.

- No software-based data plane is supported.
- No auto-discovery mechanism is supported.
- Load sharing and failover on redundant CE-PE links are not supported.
- The addition or removal of MAC addresses with Label Distribution Protocol (LDP) is not supported.
- On the Cisco 7600 series router, the virtual forwarding instance (VFI) is supported only with the **interface vlan** command.

Supported Features

Multipoint-to-Multipoint Support

Two or more devices are associated over the core network. No one device is designated as the Root node, but all devices are treated as Root nodes. All frames can be exchanged directly between nodes.

Non-Transparent Operation

A virtual Ethernet connection (VEC) can be transparent or non-transparent with respect to Ethernet PDUs (that is, BPDUs). The purpose of VEC non-transparency is to allow the end user to have a Frame Relay-type service between Layer 3 devices.

Circuit Multiplexing

Circuit Multiplexing allows a node to participate in multiple services over a single Ethernet connection. By participating in multiple services, the Ethernet connection is attached to multiple logical networks. Some examples of possible service offerings are VPN services between sites, Internet services, and third-party connectivity for intercompany communications.

MAC-Address Learning Forwarding and Aging

PEs must learn remote MAC addresses and directly attached MAC addresses on customer facing ports. MAC address learning accomplishes this by deriving topology and forwarding information from packets originating at customer sites. A timer is associated with stored MAC addresses. After the timer expires, the entry is removed from the table.

Jumbo Frame Support

Jumbo frame support provides support for frame sizes between 1548 through 9216 bytes. You use the CLI to establish the jumbo frame size for any value specified in the above range. The default value is 1500 bytes in any Layer 2/VLAN interface. You can configure jumbo frame support on a per-interface basis.

Q-in-Q Support and Q-in-Q to EoMPLS Support

With 802.1Q tunneling (Q-in-Q), the CE issues VLAN-tagged packets and the VPLS forwards the packets to a far-end CE. Q-in-Q refers to the fact that one or more 802.1Q tags may be located in a packet within the interior of the network. As packets are received from a CE device, an additional VLAN tag is added to incoming Ethernet packets to segregate traffic from different CE devices. Untagged packets originating from the CE use a single tag within the interior of the VLAN switched network, while previously tagged packets originating from the CE use two or more tags.

VPLS Services

Transparent LAN Service (TLS) and Ethernet Virtual Connection Service (EVCS) are available for service provider and enterprise use.

- Transparent LAN Service (TLS)—Use when you need transparency of bridging protocols (for example, bridge protocol data units [BPDUs]) and VLAN values. Bridges see this service as an Ethernet segment.

Note You must enable Layer 2 protocol tunneling to run the Cisco Discovery Protocol (CDP), the VLAN Trunking Protocol (VTP), and the Spanning-Tree Protocol (STP). See Chapter 18, "Configuring IEEE 802.1Q Tunneling" in the C*isco 7600 Series Cisco IOS Software Configuration Guide, 12.2SX*.

- Ethernet Virtual Connection Service (EVCS)—Use when you need routers to reach multiple intranet and extranet locations from a single physical port. Routers see subinterfaces through which they access other routers.

Transparent LAN Service

TLS is an extension to the point-to-point port-based EoMPLS. With TLS, the PE router forwards all Ethernet packets received from the customer-facing interface (including tagged, untagged, and BPDUs) as follows:

- To a local Ethernet interface or an emulated VC if the destination MAC address is found in the Layer 2 forwarding table.
- To all other local Ethernet interfaces and emulated VCs belonging to the same VPLS domain if the destination MAC address is a multicast or broadcast address or if the destination MAC address is not found in the Layer 2 forwarding table.

Ethernet Virtual Connection Service

EVCS is an extension to the point-to-point VLAN-based EoMPLS. With EVCS, the PE router forwards all Ethernet packets with a particular VLAN tag received from the customer-facing interface (excluding BPDUs) as follows:

- To a local Ethernet interface or to an emulated VC if the destination MAC address is found in the Layer 2 forwarding table.
- To all other local Ethernet interfaces and emulated VCs belonging to the same VPLS domain if the destination MAC address is a multicast or broadcast address or if the destination MAC address is not found in the Layer 2 forwarding table.

Note Because it has only local significance, the demultiplexing VLAN tag that identifies a VPLS domain is removed before forwarding the packet to the outgoing Ethernet interfaces or emulated VCs.

Benefits of VPLS

VPLS (Virtual Private LAN Service) enables enterprises to link together their Ethernet-based LANs from multiple sites via the infrastructure provided by their service provider. From the enterprise perspective, the service provider's public network looks like one giant Ethernet LAN. For the service provider, VPLS provides an opportunity to deploy another revenue-generating service on top of their existing network without major capital expenditures. Operators can extend the operational life of equipment in their network.

Configuring VPLS

This section explains how to perform a basic VPLS configuration.

Note Provisioning a VPLS link involves provisioning the associated attachment circuit and the VFI on the PE.

Note VPLS is supported on SUP720-3BXL-based systems and on Supervisor Engine 2-based systems.

Prerequisites

Before you configure VPLS, ensure that the network is configured as follows:

- Configure IP routing in the core so that the PE routers can reach each other via IP.
- Configure MPLS in the core so that a label switched path (LSP) exists between the PE routers.
- Configure a loopback interface for originating and terminating Layer 2 traffic. Make sure the PE routers can access the other router's loopback interface. Note that the loopback interface is not needed in all cases. For example, tunnel selection does not need a loopback interface when VPLS is directly mapped to a TE tunnel.

Supported Modules

Core facing modules are shown below:

- OSM-1OC48-POS-SS+
- OSM-1OC48-POS-SI+
- OSM-1OC48-POS-SL+
- OSM-2OC12-POS-SI+
- OSM-2OC12-POS-MM+
- OSM-4OC12-POS-SI+
- OSM-2+4GE-WAN-GBIC+

- OSM-4OC3-POS-SI+
- OSM-8OC3-POS-SI+
- OSM-8OC3-POS-SL+
- OSM-2OC48/1DPT-SS+ (POS mode only)
- OSM-2OC48/1DPT-SI+ (POS mode only)
- OSM-2OC48/1DPT-SL+ (POS mode only)

Customer facing interfaces are all Ethernet/ Fast Ethernet/ Gigabit Ethernet interfaces based on Layer 2 Catalyst LAN ports. See the *Catalyst 6500 Switch Module Guide* at: http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/6000hw/mod_gd/index.htm.

Basic VPLS Configuration

VPLS configuration requires you to identify peer PE routers and to attach Layer 2 circuits to the VPLS at each PE router.

VPLS configuration requires the following:

- Configuring the PE Layer 2 Interface to the CE, page 11-119
- Configuring Layer 2 VLAN Instance on the PE, page 11-124
- Configuring MPLS WAN Interface on the PE, page 11-125
- Configuring MPLS in the PE, page 11-126
- Configuring the VFI in the PE, page 11-128
- Associating the Attachment Circuit with the VSI at the PE, page 11-129

Configuring the PE Layer 2 Interface to the CE

You must configure the Layer 2 interface as a switchport for local bridging. You have the option of selecting tagged or untagged traffic from the CE device.

Note It is important to define the trunk VLANs; use the **switchport trunk allow vlan** command as shown in the first example below.

SUMMARY STEPS

Option 1—802.1Q Trunk for Tagged Traffic from the CE

1. **interface** *type number*
2. **no ip address** *ip-address mask* [**secondary**]
3. **switchport**
4. **switchport trunk encapsulation dot1q**
5. **switchport trunk allow vlan**
6. **switchport mode trunk**

Note When EVCS is configured, the PE router forwards all Ethernet packets with a particular VLAN tag to a local Ethernet interface or emulated VC if the destination MAC address is found in the Layer 2 forwarding table.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Router(config)# interface fastethernet 2/4`	Selects an interface to configure.
Step 2	**no ip address** *ip-address mask* [**secondary**] **Example:** `Router(config)# no ip address`	Disables IP processing and enters interface configuration mode.
Step 3	**switchport** **Example:** `Router(config-if)# switchport`	Modifies the switching characteristics of the Layer 2-switched interface.
Step 4	**switchport trunk encapsulation dot1q** **Example:** `Router(config-if)# switchport trunk encapsulation dot1q`	Sets the switch port encapsulation format to 802.1Q.
Step 5	**switchport trunk allow vlan** **Example:** `Router(config-if)# switchport trunk allow vlan 501`	Sets the list of allowed VLANs.
Step 6	**switchport mode trunk** **Example:** `Router(config-if)# switchport mode trunk`	Sets the interface to a trunking VLAN Layer 2 interface.

This example shows how to configure the tagged traffic.

```
Router(config)# interface GigabitEthernet4/4
Router(config)# no ip address
Router(config-if)# switchport
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport trunk allow vlan 501
Router(config-if)# switchport mode trunk
```

This example shows how to use the **show run interface** command to verify the configuration.

```
Router# show run interface GigabitEthernet4/4
Building configuration...

Current configuration : 212 bytes
```

```
!
interface GigabitEthernet4/4
 no ip address
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 500-1999
 switchport mode trunk
end
```

SUMMARY STEPS

Option 2—802.1Q Access Port for Untagged Traffic from CE

1. **interface** *type number*
2. **no ip address** *ip-address mask* [secondary]
3. **speed** [**1000** | **nonegotiate**]
4. **switchport**
5. **switchport mode access**
6. **switchport access vlan** *vlan-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet4/4`	Selects an interface to configure.
Step 2	**no ip address** *ip-address mask* [**secondary**] **Example:** `Router(config)# no ip address`	Disables IP processing and enters interface configuration mode.
Step 3	**speed** [**1000** | **nonegotiate**] **Example:** `Router(config-if)# speed nonegotiate`	Sets the port speed for an Ethernet interface; enables or disables the link negotiation protocol on the Gigabit Ethernet ports.
Step 4	**switchport** **Example:** `Router(config-if)# switchport`	Modifies the switching characteristics of the Layer 2-switched interface.
Step 5	**switchport mode access** **Example:** `Router(config-if)# switchport mode access`	Sets the interface type to nontrunking, nontagged single VLAN Layer 2 interface.
Step 6	**switchport access vlan** *vlan-id* **Example:** `Router(config-if)# switchport access vlan 501`	Sets the VLAN when the interface is in Access mode.

This example shows how to configure the untagged traffic.

```
Router(config)# interface GigabitEthernet4/4
Router(config)# no ip address
Router(config-if)# speed nonegotiate
Router(config-if)# switchport
Router(config-if)# switchport mode access
Router(config-if)# switchport access vlan 501
```

This example shows how to use the **show run interface** command to verify the configuration.

```
Router# show run interface GigabitEthernet4/4
Building configuration...

Current configuration : 212 bytes
!
interface GigabitEthernet4/4
 speed nonegotiate
 switchport
 switchport mode access
 switchport access vlan 501
end
```

SUMMARY STEPS

Option 3—Using Q-in-Q to Place All VLANs into a Single VPLS

1. **interface** *type number*
2. **no ip address** *ip-address mask* [**secondary**]
3. **speed** [1000 | nonegotiate]
4. **switchport**
5. **switchport access vlan** *vlan-id*
6. **switchport mode dot1q-tunnel**
7. **l2protocol-tunnel** [**cdp** | **stp** | **vtp**]

Note When TLS is configured, the PE router forwards all Ethernet packets received from the CE device to all local Ethernet interfaces and emulated VCs belonging to the same VPLS domain if the MAC address is not found in the Layer 2 forwarding table.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet4/4`	Selects an interface to configure.
Step 2	**no ip address** *ip-address mask* [**secondary**] **Example:** `Router(config)# no ip address`	Disables IP processing and enters interface configuration mode.

Step 3	`speed [1000 | nonegotiate]`
	
	Example:
	`Router(config-if)# speed nonegotiate`
		Sets the port speed for an Ethernet interface; enables or disables the link negotiation protocol on the Gigabit Ethernet ports.

Step 4	**switchport** **Example:** `Router(config-if)# switchport`	Modifies the switching characteristics of the Layer 2-switched interface.
Step 5	**switchport access vlan** *vlan-id* **Example:** `Router(config-if)# switchport access vlan 501`	Sets the VLAN when the interface is in Access mode.
Step 6	**switchport mode dot1q-tunnel** **Example:** `Router(config-if)# switchport mode dot1q-tunnel`	Sets the interface as an 802.1Q tunnel port.
Step 7	**l2protocol-tunnel** [**cdp** | **stp** | **vtp**] **Example:** `Router(config-if)# l2protocol-tunnel cdp`	Enables protocol tunneling on an interface.

This example shows how to configure the tagged traffic.

```
Router(config)# interface GigabitEthernet4/4
Router(config)# no ip address
Router(config-if)# speed nonegotiate
Router(config-if)# switchport
Router(config-if)# switchport access VLAN 501
Router(config-if)# switchport mode dot1q-tunnel
Router(config-if)# l2protocol-tunnel cdp
```

This example shows how to use the **show run interface** command to verify the configuration.

```
Router# show run interface GigabitEthernet4/4
Building configuration...

Current configuration : 212 bytes
!
interface GigabitEthernet4/4
 no ip address
 speed nonegotiate
 switchport
 switchport access vlan 501
 switchport mode dot1q-tunnel
 l2protocol-tunnel cdp
end
```

Use the **show spanning-tree vlan** command to verify the port is not in a blocked state.

```
Router# show spanning-tree vlan 501

VLAN0501
Spanning tree enabled protocol ieee
  Root ID    Priority    33269
             Address     0001.6446.2300
             This bridge is the root
```

```
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    33269  (priority 32768 sys-id-ext 501)
             Address     0001.6446.2300
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 0

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- --------
--------------------------------
Gi4/4            Desg FWD 4         128.388  P2p
```

Use the **show vlan id** command to verify that a specific port is configured to send and receive a specific VLAN's traffic.

```
Router# show vlan id 501

VLAN Name                              Status    Ports
---- --------------------------------- ---------
501  VLAN0501                          active    Gi4/4

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1
Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------
501  enet  100501     1500  -      -      -        -    -        0      0

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

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

Configuring Layer 2 VLAN Instance on the PE

Configuring the Layer 2 VLAN interface on the PE enables the Layer 2 VLAN instance on the PE router to the VLAN database to set up the mapping between the VPLS and VLANs.

For more information, see "Configuring VLANs" in the C*isco 7600 Series Cisco IOS Software Configuration Guide, 12.2SX* at .http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sy/swcg/vlans.htm.

SUMMARY STEPS

1. **vlan** *vlan-id*
2. **interface vlan** *vlan-id*

DETAILED STEPS

Command or Action	Purpose

Step 1	**vlan** *vlan-id* **Example:** `Router(config)# vlan 809`	Configures a specific virtual LAN (VLAN).
Step 2	**interface vlan** *vlan-id* **Example:** `Router(config)# interface vlan 501`	Configures an interface on the VLAN.

This is an example of configuring a Layer 2 VLAN instance.

```
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# vlan 501
Router(config)# interface vlan 501
Router(config-if)#
```

Use the **show interfaces vlan** command to verify the VLAN is in the up state (example not shown).

Configuring MPLS WAN Interface on the PE

The following commands configure the MPLS WAN interface.

Note The MPLS uplink must be on one of the supported OSMs.

SUMMARY STEPS

1. **interface** *type number*
2. **ip address** *ip-address mask*
3. **tag-switching ip**
4. **mls qos trust** [**cos** | **dscp** | **ip-precedence**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Router(config)# interface pos 2/4`	Selects an interface to configure.
Step 2	**ip address** *ip-address mask* **Example:** `Router# ip address 100.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface and enters interface configuration mode.
Step 3	**tag-switching ip** **Example:** `Router# tag-switching ip`	Enables label switching of IPv4 packets on an interface.
Step 4	**mls qos trust [cos | dscp | ip-precedence]** **Example:** `Router# mls qos trust dscp`	Sets the trusted state of an interface to specify that the ToS bits in the incoming packets contain a DSCP value.

This is an example of configuring the WAN interface.

```
Router(config)# interface pos4/1
Router(config)# ip address 181.10.10.1 255.255.255.0
Router(config-if)# ip directed-broadcast
Router(config-if)# ip ospf network broadcast
Router(config-if)# no keepalive
Router(config-if)# mpls label protocol ldp
Router(config-if)# tag-switching ip
Router(config-if)# mls qos trust dscp
```

Use the **show tag-switching interfaces** command to verify operation.

```
Router# show tag-switching interfaces pos4/1
Interface              IP            Tunnel   Operational
POS4/1                 Yes (ldp)     Yes      Yes
Router#
```

Configuring MPLS in the PE

To configure MPLS in the PE, you must provide the required MPLS parameters.

Note Before configuring MPLS, ensure that you have IP connectivity between all PEs by configuring Interior Gateway Protocol (IGP) (Open Shortes Path First [OSPF] or Intermediate System to Intermediate System [IS-IS]) between the PEs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol {ldp | tdp}**

4. (Optional) **mpls ldp logging neighbor-changes**
5. **tag-switching tdp discovery {hello | directed hello} {holdtime | interval}** *seconds*
6. **tag-switching tdp router-id Loopback0 force**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol {ldp | tdp}** **Example:** `Router(config)# mpls label protocol ldp`	Specifies the default Label Distribution Protocol for a platform.
Step 4	**mpls ldp logging neighbor-changes** **Example:** `Router(config)# mpls ldp logging neighbor-changes`	(Optional) Determines logging neighbor changes.
Step 5	**tag-switching tdp discovery** {**hello** | **directed hello**} {**holdtime** | **interval**} *seconds* **Example:** `Router(config)# tag-switching tdp discovery hello holdtime 5`	Configures the interval between transmission of LDP (TDP) discovery hello messages, or the hold time for a LDP transport connection
Step 6	**tag-switching tdp router-id Loopback0 force** **Example:** `Router(config)# tag-switching tdp router-id Loopback0 force`	Configures MPLS.

This example shows global MPLS configuration.

```
Router(config)# mpls label protocol ldp
Router(config)# tag-switching tdp discovery directed hello
Router(config)# tag-switching tdp router-id Loopback0 force
```

Use the **show ip cef** command to verify that the LDP label is assigned.

```
Router# show ip cef 192.168.17.7
192.168.17.7/32, version 272, epoch 0, cached adjacency to POS4/1
0 packets, 0 bytes
  tag information set
    local tag: 8149
```

```
  fast tag rewrite with PO4/1, point2point, tags imposed: {4017}
via 11.3.1.4, POS4/1, 283 dependencies
  next hop 11.3.1.4, POS4/1
  valid cached adjacency
  tag rewrite with PO4/1, point2point, tags imposed: {4017}
```

Configuring the VFI in the PE

The virtual switch instance (VFI) specifies the VPN ID of a VPLS domain, the addresses of other PE routers in this domain, and the type of tunnel signaling and encapsulation mechanism for each peer. (This is where you create the VSI and associated VCs.) Configure a VFI as follows:

Note Only MPLS encapsulation is supported.

SUMMARY STEPS

1. **l2 vfi** *name* **manua**l
2. **vpn id** *vpn id*
3. **neighbor** *remote router id* {**encapsulation** {**mpls**} [**no-split-horizon**]
4. **shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**l2 vfi** *name* **manual** **Example:** `Router(config)# l2 vfi vfi17 manual`	Enables the Layer 2 VFI manual configuration mode.
Step 2	**vpn id** *vpn-id* **Example:** `Router(config-vfi)# vpn id 17`	Configures a VPN ID for a VPLS domain. The emulated VCs bound to this Layer 2 VRF use this VPN ID for signaling.
Step 3	**neighbor** *remote router id* {**encapsulation mpls**} [**no-split-horizon**] **Example:** `Router(config-vfi)# neighbor 1.5.1.1 encapsulation mpls`	Specifies the remote peering router ID and the tunnel encapsulation type or the pseudo-wire property to be used to set up the emulated VC. **Note** Split horizon is the default configuration to avoid broadcast packet looping and to isolate Layer 2 traffic. Use the **no-split-horizo**n keyword to disable split horizon and to configure multiple VCs per spoke into the same VFI.
Step 4	**shutdown** **Example:** `Router(config-vfi)# shutdown`	Disconnects all emulated VCs previously established under the Layer 2 VFI and prevents the establishment of new attachment circuits. **Note** It does not prevent the establishment of new attachment circuits configured with the Layer 2 VFI using CLI.

The following example shows a VFI configuration.

```
Router(config)# l2 vfi VPLSA manual
Router(config-vfi)# vpn id 100
Router(config-vfi)# neighbor 11.11.11.11 encapsulation mpls
Router(config-vfi)# neighbor 33.33.33.33 encapsulation mpls
Router(config-vfi)# neighbor 44.44.44.44 encapsulation mpls
```

The following example shows a VFI configuration for hub and spoke.

```
Router(config)# l2 vfi VPLSA manual
Router(config-vfi)# vpn id 100
Router(config-vfi)# neighbor 9.9.9.9 encapsulation mpls
Router(config-vfi)# neighbor 12.12.12.12 encapsulation mpls
Router(config-vfi)# neighbor 33.33.33.33 encapsulation mpls no-split-horizon
```

The **show mpls 12transport vc** command displays various information related to PE1.

Note The **show mpls l2transport vc [detail]** command is also available to show detailed information about the VCs on a PE router as in the following example.

```
VPLS-PE2# show mpls l2transport vc 201

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
VFI test1      VFI                  153.1.0.1       201        UP
VFI test1      VFI                  153.3.0.1       201        UP
VFI test1      VFI                  153.4.0.1       201        UP
```

Note The VC ID in the output represents the VPN ID; the VC is identified by the combination of the Dest address and the VC ID as in the example below.

The **show vfi** *vfi name* command shows VFI status.

```
nPE-3# show vfi VPLS-2
VFI name: VPLS-2, state: up
  Local attachment circuits:
    Vlan2
  Neighbors connected via pseudowires:
  Peer Address     VC ID     Split-horizon
  1.1.1.1          2             Y
  1.1.1.2          2             Y
  2.2.2.3          2             N
```

Associating the Attachment Circuit with the VSI at the PE

After defining the VFI, you must bind it to one or more attachment circuits (interfaces, subinterfaces, or virtual circuits).

SUMMARY STEPS

1. **interface vlan** *vlan-id*
2. **no ip address** (Configuring an IP address causes a Layer 3 interface to be created for the VLAN.)
3. **xconnect vfi** *vfi name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface vlan** *vlan-id* **Example:** `Router(config-if)# interface vlan 100`	Creates or accesses a dynamic switched virtual interface (SVI).
Step 2	**no ip address** **Example:** `Router(config-if)# no ip address`	Disables IP processing. (You configure a Layer 3 interface for the VLAN if you configure an IP address.)
Step 3	**xconnect vfi** *vfi name* **Example:** `Router(config-if)# xconnect vfi vfi16`	Specifies the Layer 2 VFI that you are binding to the VLAN port.

This example shows an interface VLAN configuration.

```
Router(config-if)# interface vlan 100
Router(config-if)# no ip address
Router(config-if)# xconnect vfi VPLS_501
```

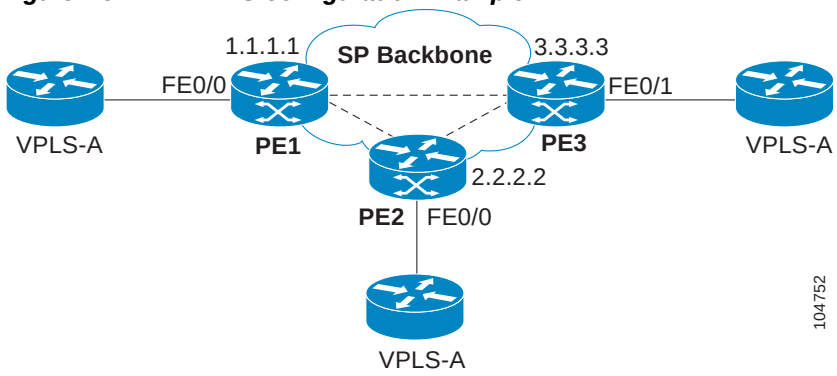
Use the **show vfi** command for VFI status.

```
Router# show vfi VPLS_501
VFI name: VPLS_501, state: up
  Local attachment circuits:
    vlan 100
  Neighbors connected via pseudowires:
    192.168.11.1  192.168.12.2  192.168.13.3  192.168.16.6
    192.168.17.7
```

Full-Mesh Configuration Example

In a full-mesh configuration, each PE router creates a multipoint-to-multipoint forwarding relationship with all other PE routers in the VPLS domain using a VFI. An Ethernet or VLAN packet received from the customer network can be forwarded to one or more local interfaces and or emulated VCs in the VPLS domain. To avoid broadcasted packets looping around in the network, no packet received from an emulated VC can be forwarded to any emulated VC of the VPLS domain on a PE router. That is, the Layer 2 split horizon should always be enabled as the default in a full-mesh network. Figure 11-5 shows the configuration example.

Figure 1-5 VPLS Configuration Example



Configuration on PE 1

This shows the creation of the virtual switch instances (VSIs) and associated VCs.

```
l2 vfi PE1-VPLS-A manual
  vpn id 100
  neighbor 2.2.2.2 encapsulation mpls
  neighbor 3.3.3.3 encapsulation mpls
!
interface Loopback 0
  ip address 1.1.1.1 255.255.255.255
```

This configures the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface FastEthernet0/0
  switchport
  switchport mode dot1qtunnel
  switchport access vlan 100
!
```

Here the attachment circuit (VLAN) is associated with the VSI.

```
interface vlan 100
  no ip address
 xconnect vfi PE1-VPLS-A
!
```

This is the enablement of the Layer 2 VLAN instance.

```
vlan 100
  state active
```

Configuration on PE 2

This shows the creation of the virtual switch instances (VSIs) and associated VCs.

```
l2 vfi PE2-VPLS-A manual
  vpn id 100
  neighbor 1.1.1.1 encapsulation mpls
  neighbor 3.3.3.3 encapsulation mpls
!
interface Loopback 0
  ip address 2.2.2.2 255.255.255.255
```

This configures the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface FastEthernet0/0
  switchport
  switchport mode dot1qtunnel
  switchport access vlan 100
```

```
!
```

Here the attachment circuit (VLAN) is associated with the VSI.

```
interface vlan 100
  no ip address
 xconnect vfi PE2-VPLS-A
!
```

This is the enablement of the Layer 2 VLAN instance.

```
vlan 100
  state active
```

Configuration on PE 3

This shows the creation of the virtual switch instances (VSIs) and associated VCs.

```
l2 vfi PE3-VPLS-A manual
  vpn id 100
  neighbor 1.1.1.1 encapsulation mpls
  neighbor 2.2.2.2 encapsulation mpls
!
interface Loopback 0
  ip address 3.3.3.3 255.255.255.255
```

This configures the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface FastEthernet0/1
  switchport
  switchport mode dot1qtunnel
  switchport access vlan 100
!
```

Here the attachment circuit (VLAN) is associated with the VSI.

```
interface vlan 100
  no ip address
 xconnect vfi PE3-VPLS-A .
!
```

This is the enablement of the Layer 2 VLAN instance.

```
vlan 100
  state active
```

The **show mpls l2 vc** command provides information on the status of the VC.

```
VPLS1# show mpls l2 vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Vi1            VFI                  22.22.22.22     100        DOWN
Vi1            VFI                  22.22.22.22     200        UP
Vi1            VFI                  33.33.33.33     100        UP
Vi1            VFI                  44.44.44.44     100        UP
Vi1            VFI                  44.44.44.44     200        UP
```

The **show vfi** command provides information on the VFI.

```
PE-1# show vfi PE1-VPLS-A
VFI name: VPLSA, state: up
  Local attachment circuits:
    Vlan100
  Neighbors connected via pseudowires:
    2.2.2.2  3.3.3.3
```

The **show mpls 12transport vc** command provides information the virtual circuits.

```
osr12# show mpls l2 vc det
Local interface: VFI vfi17 up
  Destination address: 1.3.1.1, VC ID: 17, VC status: up
    Tunnel label: imp-null, next hop point2point
    Output interface: PO3/4, imposed label stack {18}
  Create time: 3d15h, last status change time: 1d03h
  Signaling protocol: LDP, peer 1.3.1.1:0 up
    MPLS VC labels: local 18, remote 18
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, send 0
```

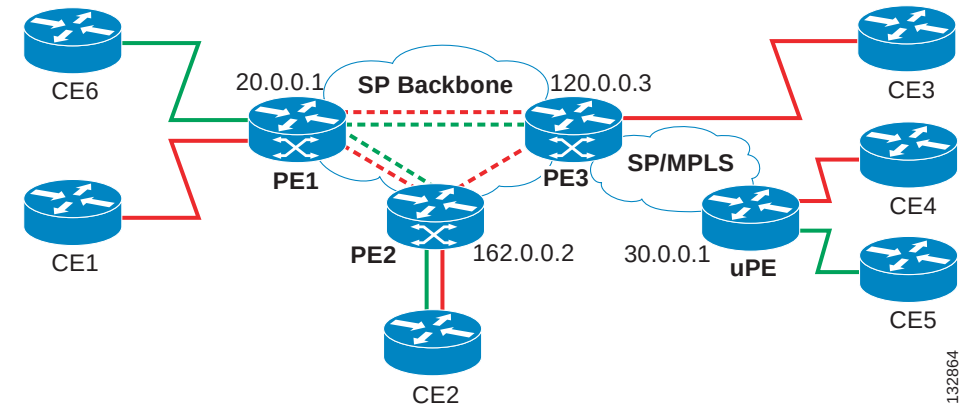
H-VPLS with MPLS Edge Configuration Example

The Hierarchical VPLS model comprises hub and spoke and full-mesh networks. In a full-mesh configuration, each PE router creates a multipoint-to-multipoint forwarding relationship with all other PE routers in the VPLS domain using VFIs.

In the hub and spoke configuration, a PE router can operate in a non-split-horizon mode that allows inter-VC connectivity without the requirement to add a Layer 2 port in the VLAN.

In the example below, the VLANs on CE1, CE2, CE3, and CE4 (in red color) connect through a full-mesh network. The VLANs on CE2, CE5, and ISP POP connect through a hub and spoke network where the ISP POP is the hub and CE2 and CE5 are the spokes. Figure 11-6 shows the configuration example.

Figure 1-6 H-VPLS Configuration



Configuration on PE1

This shows the creation of the virtual switch instances (VSIs) and associated VCs. Note that the VCs in green require the **no-split-horizon** keyword. The **no-split-horizon** command disables the default Layer 2 split horizon in the data path.

```
l2 vfi Internet manual
 vpn id 100
```

```
 neighbor 120.0.0.3 encapsulation mpls no-split-horizon
 neighbor 162.0.0.2 encapsulation mpls no-split-horizon

l2 vfi PE1-VPLS-A manual
 vpn id 200
 neighbor 120.0.0.3 encapsulation mpls
 neighbor 162.0.0.2 encapsulation mpls

interface Loopback 0
 ip address 20.0.0.1 255.255.255.255
```

This configures the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface GigEthernet1/1
 switchport
 switchport mode trunk
 switchport trunk encap dot1q
 switchport trunk allow vlan 1001,1002-1005
```

Here the attachment circuit (VLAN) is associated with the VFI.

```
interface Vlan 1001
 xconnect vfi Internet

interface FastEthernet2/1
 switchport
 switchport mode trunk
 switchport trunk encap dot1q
 switchport trunk allow vlan 211,1002-1005

interface Vlan 211
 xconnect vfi PE1-VPLS-A
```

Configuration on PE2

This shows the creation of the VFIs and associated VCs.

```
l2 vfi Internet manual
 vpn id 100
 neighbor 20.0.0.1 encapsulation mpls

l2 vfi PE2-VPLS-A manual
 vpn id 200:1
 neighbor 120.0.0.3 encapsulation mpls
 neighbor 20.0.0.1 encapsulation mpls

interface Loopback 0
 ip address 162.0.0.2 255.255.255.255
```

This configures the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface GigEthernet2/1
 switchport
 switchport mode trunk
 switchport trunk encap dot1q
 switchport trunk allow vlan 211,1001,1002-1005
```

Here the attachment circuit (VLAN) is associated with the VFI.

```
interface Vlan 1001
 xconnect vfi Internet

interface Vlan 211
 xconnect vfi PE2-VPLS-A
```

Configuration on PE3

This shows the creation of the VFIs and associated VCs.

```
l2 vfi Internet manual
 vpn id 100
 neighbor 20.0.0.1 encapsulation mpls
 neighbor 162.0.0.2 encapsulation mpls
 neighbor 30.0.0.1 encapsulation mpls no-split horizon

l2 vfi PE3-VPLS-A manual
 vpn id 200
 neighbor 162.0.0.2 encapsulation mpls
 neighbor 20.0.0.1 encapsulation mpls

interface Loopback 0
 ip address 120.0.0.3 255.255.255.255
```

This configures the CE device interface.

```
interface GigEthernet6/1
 switchport
 switchport mode trunk
 switchport trunk encap dot1q
 switchport trunk allow vlan 211
```

This configures the attachment circuits.

```
interface Vlan 1001
 xconnect vfi Internet

interface Vlan 211
 xconnect vfi PE3-VPLS-A
```

This configures port-based EoMPLS on the uPE device.

```
interface GigEthernet 1/1
 xconnect 120.0.0.3 100 encapsulation mpls
```

MAC Limit Per VLAN

VPLS provides the ability to limit the maximum number of MAC entries per VLAN to avoid exhausting resources. To enable the MAC limit feature, use the **mac-address-table limit** command; see http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/cmdref/index.htm.

Traffic Engineering for Transport Tunnel

MPLS traffic engineering software enables an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks. See

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt3/xcftagov.htm#1022001.

Load Balancing

Load balancing describes a functionality in a router that distributes packets across multiple links. For information on load balancing, see

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcprt1/1cfsflb.htm#1007566.

QoS

VPLS uses PFC-based QoS on the input side; on the core-facing interface, VPLS uses OSM-based features similar to EoMPLS, except for shaping.

Per-VLAN Shaping

Per-VLAN traffic shaping in an VPLS environment has different characteristics from EoMPLS. The queues are based on the shaping parameter on a per-MPLS port basis. A VLAN configured for a 100 Mbps shaper creates a 100 M queue on each physical MPLS uplink port in the VPLS domain. In a PE with four MPLS uplinks, this allows up to 400 Mbps of traffic to be forwarded into the core network. If two VCs share an egress interface, they would also share the same 100M shaper.

The following configuration matches all traffic input and shapes the traffic on each egress interface to 100 Mbps.

```
class-map match-all all
  match any

policy-map shape100
  class all
    shape average 100000000

interface Vlan100
 no ip address
xconnect vfi 100
service-policy output shape100
```

For information on PFC-based QoS, see "Configuring PFC QoS" at http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/swcg/qos.htm.

For information on QoS for the core-facing interface, see "How to Configure QoS with AToM" section on page 11-79.

Note If you are shaping policy to both the VLAN interface and the core-facing interface, then the policy on the VLAN interface overrides the policy on the core-facing interface.

Note VPLS supports a maximum of up to 30,000 VCs; for this number, we recommend that you configure a maximum of five different EXP classifications.

Note If a service policy is applied on the core-facing interface, then the number of VPLS VCs going out of the interfaces on a single PXF processor cannot exceed 21,000.

Configuring Dot1q Transparency for EoMPLS

The Dot1q Transparency for EoMPLS feature allows a service provider to modify the MPLS EXP bits for core-based QoS policies while leaving any VPLS customer 802.1p bits unchanged.

With releases before 12.2(18)SXF1, when applying a service policy to an EoMPLS configured VLAN interface that sets the MPLS EXP bits, the set effects both the Interior Gateway Protocol (IGP) label and the VC label. If the customer traffic includes an 802.1q label with associated 802.1p bits, the 802.1p

bits are rewritten on the egress PE based on the received VC EXP bits. If the policy sets the MPLS EXP bits to a different value from the received 802.1p bits, the rewriting on the egress PE results in a modification of the customer's 802.1p bits.

The Dot1q Transparency for EoMPLS feature provides the option for the VLAN-applied policy to affect only the IGP label (for core QoS) and leaves the VC label EXP bits equal to the 802.1p bits. On the egress PE, the 802.1p bits are still rewritten based on the received VC EXP bits, however, because the EXP bits now match the ingress 802.1p bits, a VPLS customer's 802.1p bits do not change.

Restrictions

The following restrictions apply to the Dot1q Transparency for EoMPLS feature:

- Global configuration applies to all virtual forwarding instance (VFI) and switched virtual interface (SVI) EoMPLS VCs configured on the Cisco 7600 series routers.
- Only supported on OSMs.
- Interoperability requires applying the Dot1q Transparency for EoMPLS feature to all participating PE routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **platform vfi dot1q-transparency**
4. **interface vlan**
5. **no ip address**
6. **xconnect** *peer-router-id vcid* **encapsulation mpls**
7. **service-policy output**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`platform vfi dot1q-transparency`
	
	Example:
	`Router(config)# platform vfi dot1q-transparency`
		Sets the EXP value in the remote VC label with the DBUS CoS value.

	Command or Action	Purpose
Step 4	**interface vlan** *vlanid* **Example:** `Router(config)# interface vlan 566`	Creates a unique VLAN ID number.
Step 5	**no ip address** *ip-address mask* [**secondary**] **Example:** `Router(config)# no ip address`	Disables IP processing.
Step 6	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-subif)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.
Step 7	`Router(config-if)#` **service-policy output** *policy-name* **Example:** `Router(config-if)# service-policy output policy-name ip`	Attaches a traffic policy to an interface.

This is an example of configuring the Dot1q Transparency feature.

```
platform vfi dot1q-transparency
!
l2 vfi customer-A manual
vpn id 200
neighbor 1.0.10.1 encapsulation mpls
neighbor 1.0.11.1 encapsulation mpls
neighbor 1.0.111.1 encapsulation mpls
!
class-map match-all any
match any
!
policy-map mpls-set-exp-1
class any
set mpls experimental imposition 1
!
interface Vlan200
no ip address
xconnect vfi customer-A
service-policy input mpls-set-exp-1
```

Use the **show cwan vfi dot1q-transparent** command to verify the VLAN is in the up state.

```
Router# show cwan vfi dot1q-transparency
 VFI dot1q transparency is enabled
Router#
```

INDEX

Numerics

A

B

C

D

E

F

I

L

M

O

P

Q

R

S

T

U

V

W

X