



CHAPTER 11

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, page 11-1](#)
- [HDLC Over MPLS, page 11-6](#)
- [PPP Over MPLS, page 11-10](#)
- [Configuring MPLS QoS, page 11-13](#)
- [Configuring MPLS VPN, page 11-17](#)
- [Configuring MPLS VPN QoS, page 11-20](#)
- [Any Transport over MPLS, page 11-22](#)
- [Ethernet over MPLS, page 11-27](#)
- [ATM AAL5 over MPLS VC-Mode, page 11-46](#)
- [ATM Cell Relay over MPLS VC-Mode, page 11-49](#)
- [Frame Relay Over MPLS, page 11-53](#)
- [Layer 2 Local Switching, page 11-57](#)
- [DE/CLP and EXP Mapping on FR/ATMoMPLS VC, page 11-69](#)
- [HQoS for EoMPLS Virtual Circuits, page 11-89](#)
- [AToM Load Balancing, page 11-111](#)
- [Virtual Private LAN Services on the Optical Services Modules, page 11-113](#)

Configuring MPLS

These sections describe MPLS and provides configuration information:

- [Understanding MPLS, page 11-2](#)
- [MPLS Support on OSMs, page 11-2](#)
- [Supported Features, page 11-115](#)
- [MPLS Limitations and Restrictions, page 11-5](#)

- [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/fs2s1dp.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/fdserv3.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/ftcsc113.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/ftias113.htm>.

- 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/en/US/docs/ios/12_2t/12_2t11/feature/guide/ft11bmpl.html
- 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 11-22.

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 11-19, “[Ethernet over MPLS Restrictions](#)” section on page 11-23, “[ATM AAL5 over MPLS Restrictions](#)” section on page 11-24, “[ATM Cell Relay over MPLS Restrictions](#)” section on page 11-24, “[Frame Relay over MPLS Restrictions](#)” section on page 11-24, and “[Restrictions for VPLS](#)” section on page 11-114.

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/ios122/122cgr/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-10C48-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 serialslot/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. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface serialslot/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.
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 12 vc 101**

Local intf	Local circuit	Dest address	VC ID	Status
PO1/8	HDLC	33.33.33.33	101	UP

PE1# **sh mpls 12 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
----	----------	------------	--------	-------	-------------

```
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 12 vc 101**

Local intf	Local circuit	Dest address	VC ID	Status
PO8/1	HDLC	11.11.11.11	101	UP

PE2#sh mpls 12 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
```


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 serialslot/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. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface serialslot/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.
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 12 vc 101

```

Local intf	Local circuit	Dest address	VC ID	Status
PO1/8	PPP	33.33.33.33	101	UP

```

PE2# sh mpls 12 vc 101

```

Local intf	Local circuit	Dest address	VC ID	Status
PO8/1	PPP	11.11.11.11	101	UP

```

PE1# sh mpls 12 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 12 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 to CE.

Configuring MPLS QoS

These sections provide configuration information for MPLS QoS:

- [Supported MPLS QoS Features](#), page 11-13
- [Understanding the MPLS Experimental Field](#), page 11-13
- [Configuring Class-Based Marking for MPLS \(Supervisor Engine 2\)](#), page 11-14
- [Configuration Examples](#), page 11-16

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 9-2.
- 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 11-78.

Understanding the MPLS Experimental Field

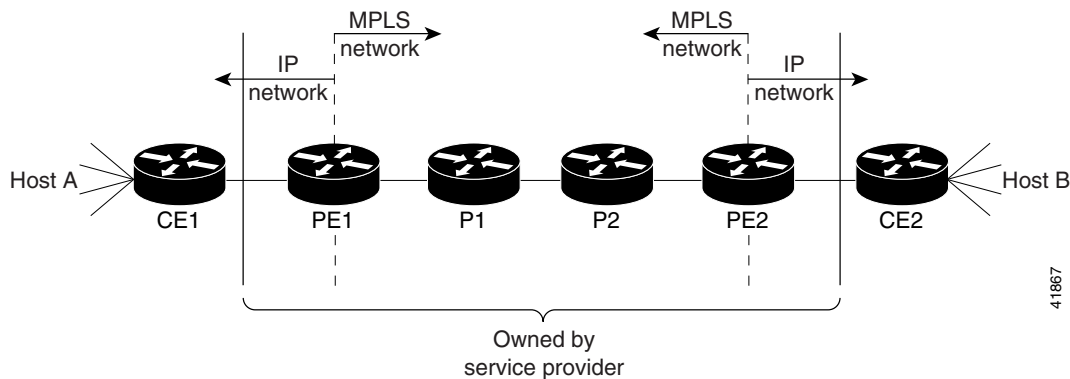
Setting the MPLS experimental field value satisfies the requirement of service providers that do not want the value of the IP precedence field modified within IP packets transported through their networks.

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 11-1 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 11-14](#)
- [Configuring a Policy Map to Set the MPLS Experimental Field, page 11-14.](#)



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 <i>class-name</i>	Specifies the class map to which packets will be matched.
Step 2	Router(config-cmap)# match mpls experimental <i>value</i>	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 <i>policy-name</i>	Creates a policy map that can be attached to one or more interfaces to specify a service policy.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of the class map previously designated in the class-map command.
Step 3	Router(config-pmap-c)# set mpls experimental <i>value</i> 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 <i>name</i>	Designates the output interface.
Step 2	Router(config-if)# service-policy {input output} <i>policy-name</i>	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 [<i>interface-name</i>]	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, page 11-18](#)
- [MPLS VPN Limitations and Restrictions, page 11-19](#)
- [MPLS VPN Memory Requirements and Recommendations, page 11-19](#)
- [MPLS Per-Label Load Balancing, page 11-20](#)

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/DPT¹:
 - 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 11-20). 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 11-1](#) apply:

Table 11-1 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 than 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 9-2.
- 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 802.1p 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 restriction applies 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/VCI 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/VCI 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 11-25](#)
- [Compatibility with Previous Releases of AToM, page 11-26](#)
- [Benefits of AToM, page 11-26](#)

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 11-27](#) 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 11-78](#) and [“HQoS for EoMPLS Virtual Circuits” section on page 11-89](#) 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 11-2 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. <ul style="list-style-type: none"> 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	<code>interface vlan <i>vlanid</i></code>	Creates a unique VLAN ID number.
	Example: Router(config)# <code>interface vlan <i>vlanid</i></code>	
Step 11	<code>mpls l2transport route <i>destination</i> <i>vc-id</i></code>	Specifies the VC to use to transport the Layer 2 VLAN packets.
	Example: Router(config-if)# <code>mpls l2transport route 9.9.11.11 3</code>	The argument <i>destination</i> specifies the loopback address of the remote router. The argument <i>vc-id</i> 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. <ul style="list-style-type: none"> • 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 <i>slot/interface.subinterface</i> 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 <i>vlan-id</i> 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 <i>peer-router-id vcid encapsulation mpls</i> 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/rcv
      LDP Id: 12.12.12.12:0
  Targeted Hellos:
    13.13.13.13 -> 11.11.11.11 (ldp): active/passive, xmit/rcv
      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  12ckt(2)      133093      V12      point2point
21   Untagged  12ckt(3)      185497      V13      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
V12	Eth VLAN 2	11.11.11.11	2	UP
V13	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: V12 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: V13 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. <ul style="list-style-type: none"> • 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	<code>switchport mode dot1qtunnel</code> Example: Router(config-if)# switchport mode dot1qtunnel	Set the trunking mode to tunneling.
Step 8	<code>switchport access vlan vlan_id</code> Example: Router(config-if)# switchport access vlan 7	Configures the port to accept traffic from the specified VLAN.
Step 9	<code>exit</code> Example: Router(config-if)# exit	Exits interface configuration mode.
Step 10	<code>interface vlan vlanid</code> Example: Router(config)# interface vlan vlanid	Creates a unique VLAN ID number.
Step 11	<code>mpls l2transport route destination vc-id</code> 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 <i>destination</i> specifies the loopback address of the remote router. The argument <i>vc-id</i> 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. <ul style="list-style-type: none"> • 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 <i>slot/interface</i> 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 <i>peer-router-id vcid encapsulation mpls</i> 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 bpdupfilter 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 a 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/rcv
    LDP Id: 12.12.12.12:0
Targeted Hellos:
 13.13.13.13 -> 11.11.11.11 (ldp): active/passive, xmit/rcv
    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   12ckt(2)        55146580   V12        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
-----
V12           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: V12 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. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

Step 3	<code>interface atmslot/port</code> Example: Router(config)# interface atm1/1	Specifies an ATM interface and enters interface configuration mode.
Step 4	<code>pvc vpi/vci l2transport</code> 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	<code>encapsulation aal5</code> 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)# <code>mpls l2transport route destination vc-id</code> 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	PE2
<pre>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 !</pre>	<pre>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 !</pre>

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 l2transport 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. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface atmslot/port Example: Router(config)# interface atml/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 aa10 Example: Router(config-atm-vc)# encapsulation aa10	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/VCI) 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/VCI must match.

The following example shows a Cell Relay over MPLS configuration.

PE1	PE2
<pre> 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 </pre>	<pre> 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 ! </pre>

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 l2transport** 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 serialslot/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. <ul style="list-style-type: none">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 serialslot/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 <i>connection-name</i> argument is a text string that you provide. The <i>interface</i> argument is the interface on which a PVC connection will be defined. The <i>dlci</i> 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
<pre> 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 </pre>	<pre> 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 </pre>



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 12 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 DLCI local switching and includes the following procedures:

- [Configuring ATM VC to VC Local Switching with AAL5 Encapsulation, page 11-58](#)
- [Configuring ATM VC to VC Local Switching with AAL0 Encapsulation, page 11-60](#)
- [Configuring ATM VP to VP Local Switching with AAL0 Encapsulation, page 11-62](#)
- [Configuring Frame Relay DLCI Local Switching, page 11-64](#)

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 11-3 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** *atmslot/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. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface atm <i>slot/port</i> Example: Router(config)# interface atm1/0	Specifies an ATM interface and enters interface configuration mode.
Step 4	pvc <i>vpi/vci</i> 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 <i>connection-name</i> atm <i>slot/port-1</i> [<i>vpi/vci</i>] atm <i>slot/port-2</i> [<i>vpi/vci</i>] 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 [vpilvci]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface atm <i>slot/port</i> Example: Router(config)# interface atm1/0	Specifies an ATM interface and enters interface configuration mode.
Step 4	pvc <i>vpi/vci</i> 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 <i>connection-name</i> atm <i>slot/port-1</i> [<i>vpi/vci</i>] atm <i>slot/port-2</i> [<i>vpi vci</i>] 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 atmslot/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 <i>slot/port</i> 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 serialslot/port**
5. **encapsulation frame-relay [cisco | ietf]**
6. **frame-relay intf-type dce**
7. **connect connection-name interface_1 dlc1_1 interface_2 dlc1_2**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • 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 serialslot/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 <i>connection-name</i> argument is a text string that you provide. The <i>interface</i> argument is the interface on which a PVC connection will be defined. The <i>dlci</i> 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

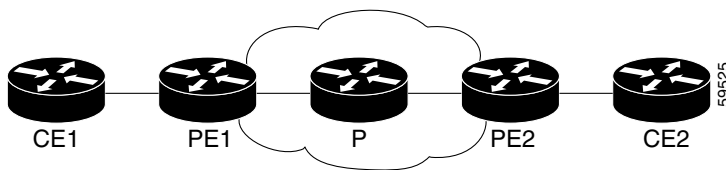
To determine the PVC status, LMI checks that a PVC is available from the reporting device to the Frame Relay end-user device. If a PVC is available, LMI reports that the status is “Active,” which means that all interfaces, line protocols, and core segments are operational between the reporting device 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 11-2 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

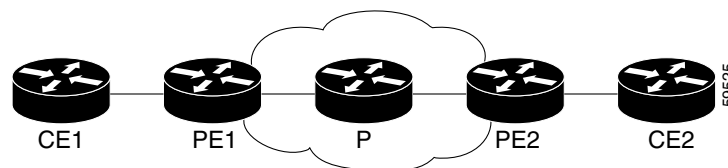
The DE/CLP and EXP Mapping on FR/ATMoMPLS VC feature allows you to map the Frame Relay Discard Eligibility (DE) bit or the ATM Congestion Loss Priority (CLP) bit to the MPLS EXP value at the ingress to an MPLS AToM network and to map the MPLS EXP value to the FR-DE or ATM CLP bit at the egress of an 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 11-3 DE/CLP and EXP Mapping



See the following sections:

- [Match on ATM CLP Bit, page 11-69](#)
- [Match on FR-DE Bit, page 11-71](#)
- [Set on ATM CLP Bit, page 11-75](#)
- [Set on FR-DE Bit, page 11-77](#)

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 <i>class-name</i>	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 <i>policy-name</i>	Specifies the name of the traffic policy to configure.
Step 6	Router(config-pmap)# class <i>class-name</i>	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 <i>value</i>	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Step 8	Router(config)# interface atm <i>interface-number</i>	Enters interface configuration mode.
Step 9	Router(config-if)# pvc [<i>name</i>] <i>vpi/vci</i> [<i>l2transport</i>]	Enters ATM virtual circuit configuration mode.
Step 10	Router(config-if)# service-policy input <i>policy-name</i>	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 <i>class-name</i>	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 <i>policy-name</i>	Specifies the name of the traffic policy to configure.
Step 6	Router(config-pmap)# class <i>class-name</i>	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 <i>value</i>	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 <i>class-map-name</i> 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 <i>class-name</i>	Specifies the user-defined name of the traffic class.

	Command or Action	Purpose
Step 4	Router(config-cmap)# match qos-group <i>qos-group-value</i>	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 <i>policy-name</i>	Specifies the name of the traffic policy to configure.
Step 6	Router(config-pmap)# class <i>class-name</i>	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 <i>slot/port</i>	Enters the interface and enters interface configuration mode.
Step 9	Router(config-if)# service-policy input <i>policy-name</i>	Attaches a traffic policy to an interface.

The following shows how to configure Set on ATM CLP Bit:

```

arthos# show policy-map qq2clp
Policy Map qq2clp
  Class qq1
    set atm-clp
arthos# show class-map qq1
Class Map match-all qq1 (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 qq2clp

```

Verify the configuration with the **show policy-map interface** command.

```

arthos# show policy interface ATM9/1
ATM9/1: VC 1/100 -

Service-policy output: qq2clp

Class-map: qq1 (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 <i>class-name</i>	Specifies the user-defined name of the traffic class.
Step 4	Router(config-cmap)# match qos-group <i>qos-group-value</i>	Identifies a specific quality of service (QoS) group value as a match criterion where the range of the <i>qos-group-value</i> is 0-7.
Step 5	Router(config-cmap)# policy-map <i>policy-name</i>	Specifies the name of the traffic policy to configure.
Step 6	Router(config-pmap)# class <i>class-name</i>	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 <i>slot/port</i>	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 <i>policy-name</i>	Attaches a traffic policy to an interface.

The following shows how to configure Set on FR-DE Bit:

```
arthos# show policy-map qq2de
Policy Map qq2de
  Class qq1
    set fr-de
arthos# show class-map qq1
Class Map match-all qq1 (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-dcli 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 11-78](#)
- [Setting the Priority of Packets with EXP Bits, page 11-82](#)
- [Enabling Traffic Shaping, page 11-84](#)



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 11-79 and “Enabling Traffic Shaping” section on page 11-80.

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 <i>class-name</i>	Specifies the user-defined name of the traffic class.
Step 2	Router(config-cmap)# match cos <i>0-7</i>	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 <i>policy-name</i>	Specifies the name of the traffic policy to configure.
Step 4	Router(config-pmap)# class <i>class-name</i>	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 <i>value</i>	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Step 6	Router(config)# interface vlan <i>vlan-number</i>	Enters the VLAN interface and enters interface configuration mode.
Step 7	Router(config-if)# service-policy [input output] <i>policy-name</i>	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 <i>class-name</i>	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 <i>policy-name</i>	Specifies the name of the traffic policy to configure.
Step 4	Router(config-pmap)# class <i>class-name</i>	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 <i>cir</i> ¹ ²	Shapes traffic according to the bit rate you specify.

	Command	Purpose
Step 6	Router(config)# interface vlan <i>vlan-number</i>	Enters the VLAN interface and enters interface configuration mode.
Step 7	Router(config-if)# service-policy [input output] <i>policy-name</i>	Assigns a traffic policy to an interface.

1. Only supported parameters are shown.
2. See [Table 9-1 on page 9-5](#).

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. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	class-map <i>class-name</i> 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 <i>policy-name</i> Example: Router(config-cmap)# policy-map doe	Specifies the name of the traffic policy to configure.
Step 6	class <i>class-name</i> 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 <i>value</i> 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 <i>slot/port</i> Router(config)# interface atm4/0	Enters the interface and enters interface configuration mode.
Step 9	service-policy input <i>policy-name</i> 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. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	class-map <i>class-name</i> 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 <i>policy-name</i> Example: Router(config-cmap)# policy-map doe	Specifies the name of the traffic policy to configure.

	Command or Action	Purpose
Step 6	<pre>class class-name</pre> <p>Example: Router(config-pmap)# class jane</p>	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	<pre>shape average bit value</pre> <p>Example: Router(config-pmap-c)# shape average 2000000 8000 8000</p>	Shapes traffic according to the bit rate you specify.
Step 8	<pre>interface slot/port</pre> <p>Router(config)# interface atm4/0</p>	Enters the interface and enters interface configuration mode.
Step 9	<pre>service-policy input policy-name</pre> <p>Example: Router(config-if)# service-policy input doe</p>	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 percent** 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 9-17.

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 11-92.
- 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 11-95.
- Create a child policy map that uses these class maps. See the “[Creating the Child Policy Map for the Egress Interface](#)” section on page 11-97.
- 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 11-101.
- 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 11-103.



Note

For more information about hierarchical traffic shaping, see the section “[Configuring Hierarchical Traffic Shaping](#)” at “[Configuring Hierarchical Traffic Shaping](#)” section on page 9-17.

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 <i>policy-map-name</i> Example: Router(config)# policy-map cos-to-qosgrp-pmap	Creates a policy map with the specified name and enters policy-map configuration mode. <ul style="list-style-type: none"> • <i>policy-map-name</i>—Name of the policy map. The name must be a unique string of up to 40 alphanumeric characters.
Step 4	description <i>string</i> 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 <i>class-default</i> 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	<pre>set qos-group {cos ip-precedence}</pre> <p>Example: Router(config-pmap-c)# set qos-group cos</p>	<p>Sets a quality of service (QoS) group identifier (ID) that can be used later to classify packets.</p> <ul style="list-style-type: none"> 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. <p>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.</p>
Step 7	<pre>interface if-type {slot/port slot/subslot/port}</pre> <p>Example: Router(config-pmap-c)# interface GigabitEthernet 5/2</p>	<p>Enters interface configuration mode for the incoming interface.</p> <p>Note This interface must be a Layer 2 LAN interface. It cannot be a Layer 3 WAN interface.</p>
Step 8	<pre>service-policy input policy-map-name</pre> <p>Example: Router(config-if)# service-policy input cos-to-qosgrp-pmap</p>	<p>Attaches the specified policy map to the interface for input (ingress) traffic.</p> <ul style="list-style-type: none"> <i>policy-map-name</i>—Name of the policy map that was created in Step 3.
<p>Note Repeat Step 7 and Step 8 for each interface that should be marking the QoS group value on incoming traffic.</p>		
Step 9	<pre>show policy-map show policy-map policy-map-name [class class-map]</pre> <p>Example: Router# show policy-map cos-to-qosgrp-pmap</p>	<p>(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.</p>

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 11-95 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 11-92.
- 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] <i>class-map-name</i> Example: Router(config)# class-map group4	Creates a class map and enters class-map configuration mode. <ul style="list-style-type: none"> • 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. • <i>class-map-name</i>—Arbitrary string that identifies this class map.
Step 4	match qos-group <i>qos-group-value</i> Example: Router(config-cmap)# match qos-group 4	Matches packets with the specified QoS group marking. <ul style="list-style-type: none"> • <i>qos-group-value</i>—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 <i>class-map-name</i> 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 11-95.

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 <i>child-pmap-name</i> 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. <ul style="list-style-type: none"> <i>child-pmap-name</i>—Name of the child policy map. The name must be a unique string of up to 40 alphanumeric characters.
Step 4	description <i>string</i> 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 { <i>class-map-name</i> class-default }	Specifies the name of a class map that should be used with this policy, and enters policy-map class configuration mode. <ul style="list-style-type: none"> <i>class-map-name</i>—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
<p>Step 6</p> <pre>shape {average} mean-rate</pre> <p>Example: Router(config-pmap-c)# shape average 10000000</p>	<p>(Optional) Shapes the traffic in this class by the limits specified.</p> <ul style="list-style-type: none"> • average—Limits traffic to the maximum bit rate that is specified by the <i>mean-rate</i> parameter. • <i>mean-rate</i>—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.
<p>Step 7</p> <pre>class {class-map-name class-default}</pre> <p>Example: Router(config-pmap)# class qosgroup5 or Router(config-pmap)# class class-default</p>	<p>Specifies the name of a class map that should be used with this policy, and enters policy-map class configuration mode.</p> <ul style="list-style-type: none"> • <i>class-map-name</i>—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.
<p>Step 8</p> <pre>priority</pre> <p>Example: Router(config-pmap-c)# priority</p>	<p>(Optional) Specifies that traffic in this class is priority traffic.</p> <p>Note You cannot configure both the shape and the priority commands in the same class.</p>
<p>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.</p>	
<p>Step 9</p> <pre>police bps [burst-normal] [burst-max] conform-action action exceed-action action [violate-action action]</pre> <p>Example: Router(config-pmap-c)# police 8000 1000 conform-action transmit exceed-action drop</p>	<p>(Optional) Specifies the policing policy that should be used for traffic in this class.</p> <ul style="list-style-type: none"> • <i>bps</i>—Average rate in bits per second. The valid range is from 8,000 to 200,000,000. • <i>burst-normal</i>—(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. • <i>burst-max</i>—(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. • <i>action</i>—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	<p>class {<i>class-map-name</i> class-default}</p> <p>Example: Router(config-pmap)# class qosgroup6 or Router(config-pmap)# class class-default</p>	<p>Specifies the name of a class map that should be used with this policy, and enters policy-map class configuration mode.</p> <ul style="list-style-type: none"> <i>class-map-name</i>—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	<p>bandwidth {<i>bandwidth-kbps</i> remaining percent <i>percentage</i> percent <i>percentage</i>}</p> <p>Example: Router(config-pmap-c)# bandwidth percent 50</p>	<p>(Optional) Specifies the bandwidth that is allowed for traffic in this class.</p> <ul style="list-style-type: none"> <i>bandwidth-kbps</i>—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 <i>percentage</i> is from 1 to 100. percent—Amount of guaranteed bandwidth, based on an absolute percent of available bandwidth. The valid range for <i>percentage</i> is from 1 to 100. <p>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.</p>
	Note Repeat Step 10 through Step 11 for each class to be used in this child policy map.	
Step 12	<p>end</p> <p>Example: Router(config-pmap-c)# end</p>	<p>Exits policy-map class configuration mode and returns to privileged EXEC mode.</p>
Step 13	<p>show policy-map show policy-map <i>child-pmap-name</i> [class <i>class-map</i>]</p> <p>Example: Router# show policy-map child-policy1 (command output)</p>	<p>(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.</p>

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 11-95](#).

```
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 11-103 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 <i>class-map-name</i> Example: Router(config)# class-map vlan-map	Creates a class map and enters class-map configuration mode. <ul style="list-style-type: none"> • <i>class-map-name</i>—Arbitrary string that identifies this class map. <p>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.</p>

	Command or Action	Purpose
Step 4	<p>match input vlan <i>input-vlan-list</i></p> <p>Example: Router(config-cmap)# match input vlan 10 20 30 100-1999</p>	<p>Matches packets that are tagged with a VLAN ID specified in the <i>input-vlan-list</i>, which can be one or both of the following:</p> <ul style="list-style-type: none"> • 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. <p>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.</p>
Step 5	<p>end</p> <p>Example: Router(config-cmap)# end</p>	<p>Exits class-map configuration mode and returns to privileged EXEC mode.</p>
Step 6	<p>show class-map <i>class-map-name</i></p> <p>Example: Router# show class-map vlan-map</p>	<p>(Optional) Displays the configured class map to verify the configuration.</p>

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 11-103,](#) 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 11-97 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 <i>parent-pmap-name</i> 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. <ul style="list-style-type: none"> <i>parent-pmap-name</i>—Name of the parent policy map. The name must be a unique string of up to 40 alphanumeric characters.
Step 4	description <i>string</i> Example: Router(config-pmap)# description Parent Policy Map	(Optional) Arbitrary string, up to 200 characters long, that describes this policy map.
Step 5	class { <i>class-map-name</i> } 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. <ul style="list-style-type: none"> <i>class-map-name</i>—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 11-101.
Step 6	shape { average } <i>mean-rate</i> Example: Router(config-pmap-c)# shape average 10000000	(Optional) Shapes the traffic in this class by the limits specified. <ul style="list-style-type: none"> average—Limits traffic to the maximum bit rate that is specified by the <i>mean-rate</i> parameter. <i>mean-rate</i>—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	<p>bandwidth {<i>bandwidth-kbps</i> percent <i>percentage</i>}</p> <p>Example: Router(config-pmap-c)# bandwidth percent 50</p>	<p>(Optional) Specifies the bandwidth that is allowed for traffic in this class.</p> <ul style="list-style-type: none"> <i>bandwidth-kbps</i>—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 <i>percentage</i> is from 1 to 100. <p>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.</p>
Step 8	<p>service-policy <i>child-pmap-name</i></p> <p>Example: Router(config-pmap-c)# service-policy child-pmap-name</p>	<p>Specifies a child policy map that should be applied to the traffic in this class:</p> <ul style="list-style-type: none"> <i>child-pmap-name</i>—Name of a child policy map that was created previously in the “Creating the Child Policy Map for the Egress Interface” section on page 11-97. (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.)
	<p>Note Repeat Step 5 through Step 8 for each class to be used to match VLANs in this parent policy map.</p>	
Step 9	<p>interface <i>if-type</i> {<i>slot/port</i> <i>slot/subslot/port</i>}</p> <p>Example: Router(config)# interface ge-wan 5/2</p>	<p>Enters interface configuration mode for the specified interface.</p>
Step 10	<p>service-policy output <i>parent-pmap-name</i></p> <p>Example: Router(config-pmap)# service-policy output parent-policy1</p>	<p>Attaches the specified parent policy map to the interface for outgoing traffic.</p> <ul style="list-style-type: none"> <i>parent-pmap-name</i>—Name of the policy map that was created in Step 3.
Step 11	<p>end</p> <p>Example: Router(config-pmap-c)# end</p>	<p>Exits policy-map class configuration mode and returns to privileged EXEC mode.</p>
Step 12	<p>show policy-map show policy-map <i>parent-pmap-name</i> [class <i>class-map</i>]</p> <p>Example: Router# show policy-map vlan-map</p>	<p>(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.</p>

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 11-107](#)

- [Complete Hierarchical QoS Example, page 11-107](#)
- [Multiple Parent Policies Using the Same Child Policy Example, page 11-109](#)
- [Common Class-Map Templates Example, page 11-109](#)

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 Ctrl-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, page 11-113](#)
- [Supported Features, page 11-115](#)
- [VPLS Services, page 11-116](#)
- [Benefits of VPLS, page 11-117](#)
- [Configuring VPLS, page 11-117](#)
- [Basic VPLS Configuration, page 11-118](#)
- [Full-Mesh Configuration Example, page 11-129](#)
- [H-VPLS with MPLS Edge Configuration Example, page 11-132](#)
- [Configuring Dot1q Transparency for EoMPLS, page 11-135](#)

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 11-4 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 *Cisco 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 11-118](#)
- [Configuring Layer 2 VLAN Instance on the PE, page 11-123](#)
- [Configuring MPLS WAN Interface on the PE, page 11-124](#)
- [Configuring MPLS in the PE, page 11-125](#)
- [Configuring the VFI in the PE, page 11-127](#)
- [Associating the Attachment Circuit with the VSI at the PE, page 11-128](#)

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	<code>interface type number</code> Example: Router(config)# interface fastethernet 2/4	Selects an interface to configure.
Step 2	<code>no ip address ip-address mask [secondary]</code> Example: Router(config)# no ip address	Disables IP processing and enters interface configuration mode.
Step 3	<code>switchport</code> Example: Router(config-if)# switchport	Modifies the switching characteristics of the Layer 2-switched interface.
Step 4	<code>switchport trunk encapsulation dot1q</code> Example: Router(config-if)# switchport trunk encapsulation dot1q	Sets the switch port encapsulation format to 802.1Q.
Step 5	<code>switchport trunk allow vlan</code> Example: Router(config-if)# switchport trunk allow vlan 501	Sets the list of allowed VLANs.
Step 6	<code>switchport mode trunk</code> 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 <i>type number</i> Example: Router(config)# interface GigabitEthernet4/4	Selects an interface to configure.
Step 2	no ip address <i>ip-address mask</i> [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 <i>vlan-id</i> 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 <i>type number</i> Example: Router(config)# interface GigabitEthernet4/4	Selects an interface to configure.
Step 2	no ip address <i>ip-address mask</i> [secondary] Example: Router(config)# no ip address	Disables IP processing and enters interface configuration mode.

Step 3	<pre>speed [1000 nonegotiate]</pre> <p>Example: Router(config-if)# speed nonegotiate</p>	Sets the port speed for an Ethernet interface; enables or disables the link negotiation protocol on the Gigabit Ethernet ports.
Step 4	<pre>switchport</pre> <p>Example: Router(config-if)# switchport</p>	Modifies the switching characteristics of the Layer 2-switched interface.
Step 5	<pre>switchport access vlan vlan-id</pre> <p>Example: Router(config-if)# switchport access vlan 501</p>	Sets the VLAN when the interface is in Access mode.
Step 6	<pre>switchport mode dot1q-tunnel</pre> <p>Example: Router(config-if)# switchport mode dot1q-tunnel</p>	Sets the interface as an 802.1Q tunnel port.
Step 7	<pre>l2protocol-tunnel [cdp stp vtp]</pre> <p>Example: Router(config-if)# l2protocol-tunnel cdp</p>	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 *Cisco 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](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 Example: Router(config)# vlan 809	<code>vlan <i>vlan-id</i></code>	Configures a specific virtual LAN (VLAN).
Step 2 Example: Router(config)# interface vlan 501	<code>interface vlan <i>vlan-id</i></code>	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	<code>interface type number</code> Example: Router(config)# interface pos 2/4	Selects an interface to configure.
Step 2	<code>ip address ip-address mask</code> 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	<code>tag-switching ip</code> Example: Router# tag-switching ip	Enables label switching of IPv4 packets on an interface.
Step 4	<code>mls qos trust [cos dscp ip-precedence]</code> 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 Shortest 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. <ul style="list-style-type: none"> • 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 manual**
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-horizon 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)# 12 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)# 12 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 l2transport 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 <i>vlan-id</i> 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 <i>vfi name</i> 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.

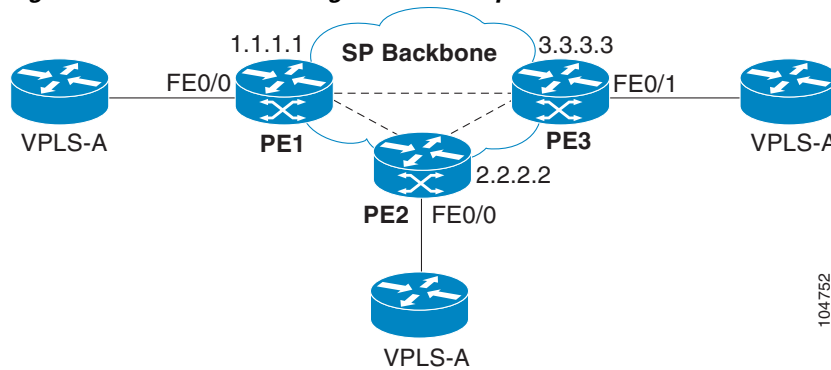
```
Router(config-if)# interface vlan 100
Router(config-if)# no ip address
Router(config-if)# xconnect vfi VPLS_501
```

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 11-5 VPLS Configuration Example

104752

Configuration on PE 1

This shows the creation of the virtual switch instances (VSIs) and associated VCs.

```
12 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.

```
12 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.

```
12 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 12 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
```

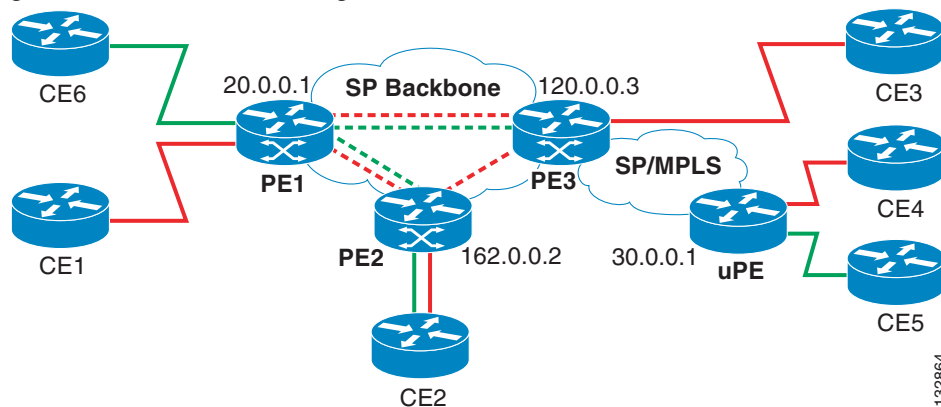
H-VPLS with MPLS Edge Configuration Example

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 11-6 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.

```
12 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

12 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.

```

12 vfi Internet manual
  vpn id 100
  neighbor 20.0.0.1 encapsulation mpls

12 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.

```
12 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

12 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 11-78.



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. <ul style="list-style-type: none"> • 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 <i>vlan</i> <i>vlanid</i> Example: Router(config)# interface vlan 566	Creates a unique VLAN ID number.
Step 5	no ip address <i>ip-address mask</i> [secondary] Example: Router(config)# no ip address	Disables IP processing.
Step 6	xconnect <i>peer-router-id vcid</i> encapsulation <i>mpls</i> 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 <i>policy-name</i> Example: Router(config-if)# service-policy output <i>policy-name ip</i>	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#
```

