

Configuring Layer 2 Services over MPLS

This chapter describes how to configure layer 2 services over MPLS in your ERX edge router.



Note: It is important to be familiar with MPLS and the type of layer 2 interfaces you want to configure before configuring layer 2 services over MPLS.

Topic	Page
Overview	5-1
References	5-2
How Layer 2 Services over MPLS Work	5-3
Configuring Layer 2 Services over MPLS	5-4
Configuration Example	5-6
Monitoring Layer 2 Services over MPLS	5-10

Overview

Many Internet service providers offer multiple services such as Frame Relay, ATM, and IP to their customers, but are consolidating to a single, packet-based, optical network from several service-specific legacy layer 2 networks. While legacy layer 2 network links are disappearing from the Internet service provider's network, the legacy layer 2 network links and services to customers must remain.

Layer 2 services over IP/MPLS allow service providers to emulate the legacy layer 2 network links and services over their IP/MPLS-based network. Layer 2 services over MPLS are especially desirable because MPLS provides features such as traffic engineering and congestion control. These MPLS features can be used to emulate certain layer 2 service characteristics.

Figure 1-1 illustrates layer 2 services being supported over an Internet service provider's MPLS network. Customers using Frame Relay or other legacy layer 2 connections to ERX edge routers are unaware that MPLS tunneling is used.

The ERX layer 2 services currently supported are:

- Frame Relay over MPLS
- Ethernet/VLAN over MPLS

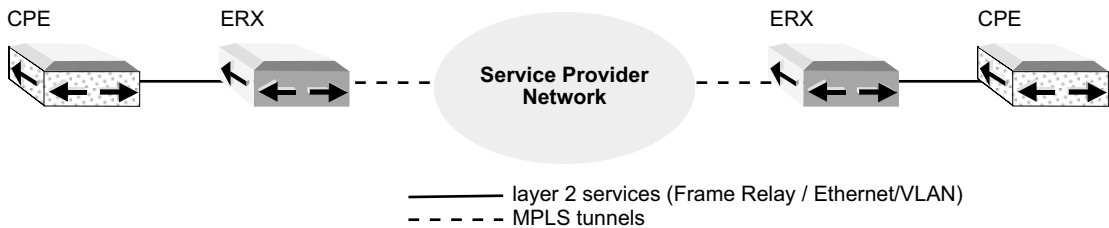


Figure 5-1 Layer 2 services over a provider's MPLS network

References

The following references provide additional information about layer 2 services, Frame Relay, Ethernet, and MPLS.

For information about layer 2 services, see:

- Transport of Layer 2 Frames Over MPLS – draft-martini-12circuit-trans-mpls-10.txt (February 2003 expiration)
- Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks, draft-martini-12circuit-encap-mpls-04.txt (May 2002 expiration)
- Framework for Pseudo Wire Emulation Edge-to-Edge (PWE3) – draft-pate-pwe3-framework-01.txt (December 2002 expiration)

For information about Frame Relay, see:

- *ERX Physical and Link Layers Configuration Guide, Chapter 11, Configuring Frame Relay*
- RFC 2115 – Management Information Base for Frame Relay DTEs Using SMIv2 (September 1997)
- RFC 2863 – The Interfaces Group MIB (June 2000)
- RFC 2427 – Multiprotocol Interconnect over Frame Relay (September 1998)

For information about Ethernet, see:

- IEEE 802.1q (FE and GE)
- IEEE 802.3 (FE and GE)
- IEEE 802.3u (FE only)
- IEEE 802.3z (GE only)
- RFC 826 – An Ethernet Address Resolution Protocol (November 1982)
- RFC 894 – A Standard for the Transmission of IP Datagrams over Ethernet Networks (April 1984)
- RFC 1042 – A Standard for the Transmission of IP Datagrams over IEEE 802 Networks (February 1988)
- RFC 1112 – Host Extensions for IP Multicasting (August 1989)

For information about MPLS, see:

- *ERX Routing Protocols Configuration Guide, Vol. 2, Chapter 2, Configuring MPLS*

How Layer 2 Services over MPLS Work

Figure 1-2 is an example of layer 2 service layering in a Frame Relay scenario.

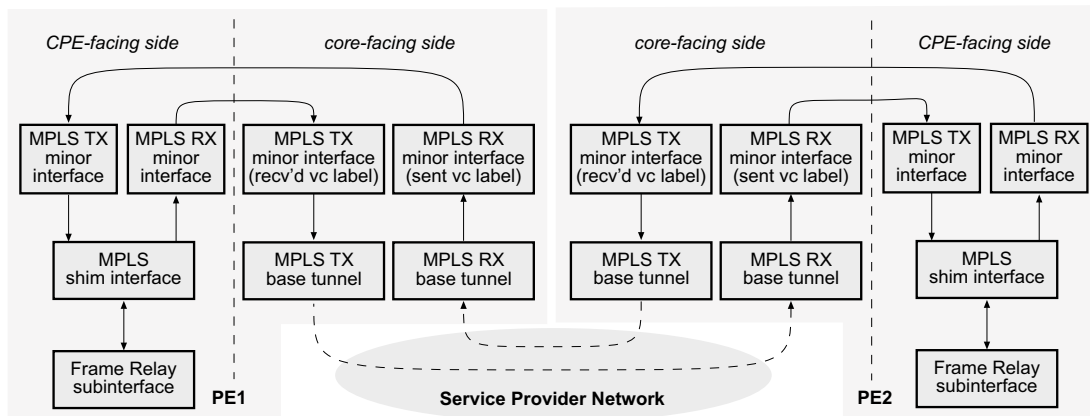


Figure 5-2 Frame Relay over MPLS layering

When layer 2 services are configured over MPLS, layer 2 traffic is encapsulated in MPLS frames and sent over MPLS tunnels. A virtual

circuit (VC) label that indicates a specific layer 2 connection, such as a Frame Relay DLCI, is pushed into the label stack between the tunnel label and the layer 2 data.

An optional, service-specific control word may be placed between the layer 2 data and the VC label. The control word is used for frame sequencing and/or carrying service-specific information, such as Frame Relay forward explicit congestion notification (FECN) and backward explicit congestion notification (BECN) information. At the tunnel end, the VC label is used to find the layer 2 interface over which the traffic is sent. The control word, if present, is used to convert the encapsulated layer 2 traffic into its native format.

Since MPLS labels are unidirectional, two VC labels are required for each layer 2 connection. In the ERX implementation, there are two MPLS minor interfaces that correspond to the two service labels. These MPLS minor interfaces are label-switched to two other minor interfaces that are stacked on top of the MPLS shim interface, which is stacked above the layer 2 interface. The shim interface is used to group the two unidirectional MPLS minor interfaces into a bidirectional conduit.

The VC labels are distributed by the Label Distribution Protocol (LDP) in downstream unsolicited (DU) mode over targeted sessions between the two routers. The layer 2 connection status signaling may be emulated by advertising and withdrawing the VC labels. For example, if the Frame Relay subinterface between a customer premises equipment (CPE) and a provider edge router (PE) goes down, the corresponding VC label is withdrawn by the PE. When the remote PE at the other end receives the label withdrawal, it translates the label withdrawal into LMI notifications to its CPE. When the Frame Relay subinterface comes back, a VC label is advertised, and the remote PE again translates it into LMI notifications.

Configuring Layer 2 Services over MPLS

Before you configure layer 2 services over MPLS, configure the CPE-facing and core-facing interfaces.



Note: To configure Ethernet or Frame Relay interfaces, see *ERX Physical and Link Layers Configuration Guide, Chapter 11, Configuring Frame Relay and Chapter 6, Configuring Ethernet Interfaces*.

To configure layer 2 services over MPLS:

- 1 Configure the layer 2 interface.
 - a Configure a Frame Relay interface. For example:

```
host1(config)# interface serial 4/1:1/1
```

```
host1(config-if)# encapsulation frame-relay ietf
host1(config-if)# frame-relay intf-type dte
host1(config-if)# frame-relay lmi-type ansi
host1(config-if)# interface serial 4/1:1/1.1
host1(config-subif)#frame-relay interface-dlci 17 ietf
```

b Configure an Ethernet/VLAN interface. For example:

```
host1(config)#interface fastethernet 4/0
host1(config-if)#encapsulation vlan
host1(config-if)#interface fastethernet 4/0.3
host1(config-if)#vlan id 201
```

2 Specify MPLS tunneling using the appropriate command.

```
host1(config)# mpls-relay 10.10.100.2 45
```

or

```
host1(config)# route interface tunnel mpls:tunnel6 45
```

3 Configure Frame Relay and MPLS on remote PE.

mpls-relay

- Use this command to route layer 2 traffic to the specified router.
- When you specify the remote destination address (DA), a tunnel to that remote DA is automatically found, if it exists.
- The VC-ID can be any 32-bit value; however, the VC-ID on both ends of the connection across the MPLS core must match.
- Optionally, you can use a group ID parameter to configure the route to be in a group of virtual connections. The ID can be any 32-bit value.
- Use the **no** version to negate this command.



Note: *The VLAN ID or DLCI may be different on each end of the connection.*

route interface

- Use this command to route layer 2 traffic on a specific tunnel interface.
- The VC ID can be any 32-bit value; however, the VC-ID on both ends of the connection across the MPLS core must match.
- Use the **no** version to negate this command.



Note: *The VLAN ID or DLCI may be different on each end of the connection.*

Configuration Example

The script provided in this section is one way to configure Frame Relay services over MPLS. Explanation notes are provided within the script. You must change the script for your specific configuration.

The topology example shown in Figure 1-3 further explains the configuration script.



Note: The `route interface` command is used toward the end of the configuration script. The `mpls-relay` command can be substituted, depending on the tunneling method best for your environment.

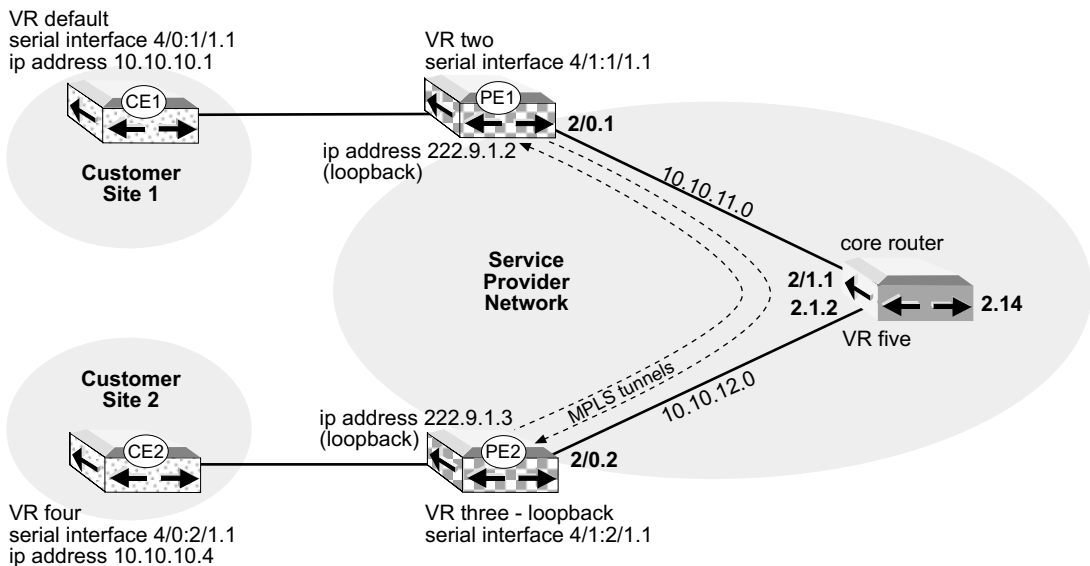


Figure 5-3 Configuration example of Frame Relay over MPLS

```

hostname "host 1"
exception protocol ftp anonymous null
!-----
!Configure CT3 module in slot 4 for Frame Relay.
!-----
!
controller t3 4/0
no shutdown
clock source internal module
cablelength 5
t1 1 clock source internal module
t1 1/1 timeslots 1-24 speed 64
t1 2 clock source internal module

```

```
    t1 2/1 timeslots 1-24 speed 64
    !
controller t3 4/1
    no shutdown
    clock source internal module
    cablelength 5
    t1 1 clock source internal module
    t1 1/1 timeslots 1-24 speed 64
    t1 2 clock source internal module
    t1 2/1 timeslots 1-24 speed 64

!-----
!Create virtual router default.
!-----

    virtual-router default
    interface loopback 0
        ip address 222.9.1.1 255.255.255.255

!-----
!Configure Frame Relay interfaces.
!-----

    interface serial 4/0:1/1
        encapsulation frame-relay ietf
    interface serial 4/0:1/1.1
        frame-relay interface-dlci 17 ietf
    !
    interface serial 4/0:2/1
        encapsulation frame-relay ietf
    interface serial 4/0:2/1.1
        frame-relay interface-dlci 12 ietf
    !
    interface serial 4/1:1/1
        encapsulation frame-relay ietf
        frame-relay intf-type dce
    interface serial 4/1:1/1.1
        frame-relay interface-dlci 17 ietf
    !
    interface serial 4/1:2/1
        encapsulation frame-relay ietf
        frame-relay intf-type dce
    interface serial 4/1:2/1.1
        frame-relay interface-dlci 12 ietf
```

```
!-----  
!Create virtual router two. Configure MPLS.  
!-----  
        virtual-router two  
  
        mpls  
        mpls ldp tar send list 222.9.1.3  
  
        interface loopback 0  
            ip address 222.9.1.2 255.255.255.255  
            ip router isis  
  
        interface atm 2/0  
            atm clock inter mod  
        interface atm 2/0.1  
            atm pvc 1 1 11 aal5snap  
            ip address 10.10.11.2 255.255.255.0  
            ip router isis  
        mpls  
        mpls ldp  
        router isis  
            net 47.0005.80FF.F800.0000.0000.0004.0000.F209.0202.00  
            mpls traffic-eng router-id loopback 0  
            mpls traffic-eng level-1  
            metric-style wide  
  
!-----  
!Create virtual router three. Configure MPLS.  
!-----  
        virtual-router three  
  
        mpls  
        mpls ldp tar send list 222.9.1.2  
  
        interface loopback 0  
            ip address 222.9.1.3 255.255.255.255  
            ip router isis  
  
        interface atm 2/0.2  
            atm pvc 2 1 12 aal5snap  
            ip address 10.10.12.3 255.255.255.0  
            ip router isis  
        mpls  
        mpls ldp  
  
        router isis  
            net 47.0005.80FF.F800.0000.0000.0004.0000.F209.0303.00
```

```
mpls traffic-eng router-id loopback 0
mpls traffic-eng level-1
metric-style wide
!-----
!Create virtual router four.
!-----
virtual-router four

interface loopback 0
ip address 222.9.1.4 255.255.255.255

!-----
!Create virtual router five. Configure MPLS.
!-----

virtual-router five

mpls

interface loopback 0
ip address 222.9.1.5 255.255.255.255
ip router isis

interface atm 2/1.1
atm pvc 1 1 11 aal5snap
ip address 10.10.11.5 255.255.255.0
ip router isis
mpls
mpls ldp

interface atm 2/1.2
atm pvc 2 1 12 aal5snap
ip address 10.10.12.5 255.255.255.0
ip router isis
mpls
mpls ldp

router isis
net 47.0005.80FF.F800.0000.0000.0004.0000.F209.0505.00
mpls traffic-eng router-id loopback 0
mpls traffic-eng level-1
metric-style wide
```

```
!-----  
!Create MPLS tunnel from VR three to VR two. Route Frame Relay traffic via MPLS tunnel.  
!-----  
  
    vir three  
  
        interface tunnel mpls:3  
            tunnel mpls label cr-ldp  
            tunnel destination 222.9.1.2  
  
        interface serial 4/1:2/1.1  
        route interface tunnel mpls:3 45  
  
    vir two  
  
        interface tunnel mpls:2  
            tunnel mpls label cr-ldp  
            tunnel destination 222.9.1.3  
  
        int ser 4/1:1/1.1  
        route interface tunnel mpls:2 45
```

Monitoring Layer 2 Services over MPLS

You can monitor layer 2 services over MPLS using the **show mpls interface [shim]** and the **show mpls forwarding** commands.

show mpls interface **show mpls interface shim**

- Use to display status and configuration information about MPLS interfaces.
- When the keyword **shim** is specified instead of a particular interface, all shim interfaces are displayed. Otherwise, the specified interface is displayed.
- Field descriptions
 - › Interface – specifier and status of each interface
 - › base-LSP/remote-addr – identifies either the tunnel that is selected to forward the traffic, or the address of the router at the other end.
 - › group-id – group ID number for the interface
 - › vc-id – VC ID number for the interface
 - › state/in/out-label – status of the layer 2 over MPLS connection or the incoming/outgoing VC label
 - › mtu – maximum transmission unit for the interface
 - › Mpls Statistics
 - pkts – number of packets received or sent
 - hcPkts – number of high-capacity (64-bit) packets received or sent
 - octets – number of octets received or sent

- hcOctets – number of high-capacity (64-bit) octets received or sent
- errors – number of packets that are dropped for some reason at receipt or before being sent
- discardPkts – number of packets that are discarded due to lack of buffer space at receipt or before being sent

- Examples

```
host1:two#show mpls interface shim brief
```

Interface	base-LSP/remote-addr	group-id	vc-id	state/in/out-label
serial4/1:1/1.1	tun mpls:1	0	1	20 45
serial4/1:1/1.2	tun mpls:1	0	2	21 46
serial4/1:1/1.3	tun mpls:1	0	3	22 47

```
host1:two#show mpls interface shim
```

```
serial4/1:1/1.1
```

```
base LSP tun mpls:1 to 222.9.1.3
```

```
group-id 0 vc-id 1 mtu 1600
```

```
State UP
```

```
In Label 20 on stack
```

```
0 pkts, 0 hcPkts, 0 octets
```

```
0 hcOctets, 0 errors, 0 discardPkts
```

```
Out Label 45 on tun mpls:1
```

```
0 pkts, 0 hcPkts, 0 octets
```

```
0 hcOctets, 0 errors, 0 discardPkts
```

show mpls forwarding

- Use to display configuration and statistics for all label-switching paths (LSPs) or for specific LSPs configured on the label-switching router (LSR).
- Field descriptions
 - › name/id – interface specifier
 - › destination – destination ip address
 - › In label – label sent to upstream neighbor for route
 - › Out label – label received from downstream neighbor for route
 - › pkts – number of packets sent across tunnel
 - › hcPkts – number of high-capacity (64-bit) packets sent across tunnel
 - › octets – number of octets sent across tunnel
 - › hcOctets – number of high-capacity (64-bit) octets sent across tunnel
 - › errors – number of packets that are dropped for some reason before being sent
 - › discardPkts – number of packets that are discarded due to lack of buffer space before being sent

- Examples

```
host1:two#show mpls forwarding brief
```

name/id	destination	label/intf
serial4/1:1/1.1	222.9.1.3	In 20
		Out 45 on tun mpls:1 nbr 222.9.1.3
serial4/1:1/1.2	222.9.1.3	In 21
		Out 46 on tun mpls:1 nbr 222.9.1.3

```
host1:two#show mpls forwarding
```

```
serial4/1:1/1/1/1/1.1 to 222.9.1.3
  In label 20
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts
  Out label 45 on tun mpls:1 nbr 222.9.1.3
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts
```