Chapter 5

Configuring Layer 2 Services over MPLS

This chapter contains the following sections:

Before You Configure Layer 2 Services over MPLS

Before you configure layer 2 services over Multiprotocol Label Switching (MPLS), we recommend you be thoroughly familiar with MPLS and the type of layer 2 interfaces that you want to configure.

Before you configure layer 2 services over MPLS, you must configure the layer 2 interfaces and MPLS.

This chapter describes how to configure different types of layer 2 services over MPLS. Each procedure uses either the **mpls-relay** command or the **route-interface** command to configure MPLS tunneling.

Related Topics

- *Chapter 4, Layer 2 Services over MPLS Overview*
- For more information about configuring MPLS and the layer 2 interfaces that support L2VPNs, see *Layer 2 Services over MPLS References* on page 488.

Configuring Frame Relay Layer 2 Services

To configure Frame Relay layer 2 services over MPLS:

1. Configure the Frame Relay interface.

   ```
   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
   ```

2. Specify MPLS tunneling by using the appropriate command.

   ```
   host1(config-if)# 10.10.100.2 45
   ```

   or

   ```
   host1(config-if)# mpls:tunnel6 45
   ```

3. Configure Frame Relay and MPLS on the remote PE router.

Related Topics

- For information about configuring a more complex Frame Relay over MPLS topology, see *Frame Relay over MPLS Configuration Example* on page 515.
- **encapsulation frame-relay ietf** command
- **frame-relay interface-dlci ietf** command
- **frame-relay intf-type** command
- **frame-relay lmi-type** command
- **interface serial** command
- **mpls-relay** command
- **route interface** command

Configuring Ethernet/VLAN Layer 2 Services

To configure Ethernet/VLAN layer 2 services over MPLS:

1. Configure the Ethernet/VLAN interface.

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

2. Specify MPLS tunneling by using the appropriate command.

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

   or

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

3. Configure Ethernet/VLAN and MPLS on the remote PE router.

Related Topics

- **encapsulation vlan** command
- **interface fastEthernet** command
- **mpls-relay** command
- **route interface** command
- **vlan id** command

Configuring S-VLAN Tunnels for Layer 2 Services

When you configure Ethernet or bridged Ethernet layer 2 services over MPLS, you can use the **svlan id** command with the **any** keyword to create a stacked VLAN (S-VLAN) tunnel that uses a single interface to tunnel traffic from multiple VLANs across an MPLS network. The S-VLAN tunnel enables multiple VLANs, each configured with a different VLAN ID tag and a common S-VLAN ID, to share a common VC label while traversing an MPLS network.

You can use the **svlan id** command with the **any** keyword only with the **mpls-relay** command or the **route interface** command to configure layer 2 services over MPLS.

To configure S-VLAN tunnels for Ethernet/VLAN layer 2 services over MPLS:

1. Configure the Ethernet/VLAN interface.

   host1(config)#**interface fastEthernet 8/1**
   host1(config-if)#**encapsulation vlan**
   host1(config-if)#**interface fastEthernet 8/1.1**

2. Create the S-VLAN tunnel and assign the S-VLAN Ethertype. For example, the following commands tunnel traffic from VLANs configured with an S-VLAN ID of 33 and any VLAN ID to the same destination across the MPLS network.

   host1(config-if)#**svlan id 33 any**
   host1(config-if)#**svlan ethertype 8100**

3. Specify MPLS tunneling by using the appropriate command. For example:

   host1(config-if)#**route interface tunnel mpls:tunnel3 45**

   or

   host1(config-if)#**mpls-relay 10.10.100.2 45**

4. Repeat these steps, using unique values to configure the S-VLAN tunnel and MPLS on the remote PE router.

Related Topics

- For more information about S-VLANs, including complete configuration instructions, see *JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces*.
- **encapsulation vlan** command
- **interface fastEthernet** command
- **mpls-relay** command
- **route interface** command
- **svlan ethertype** command
- **svlan id** command

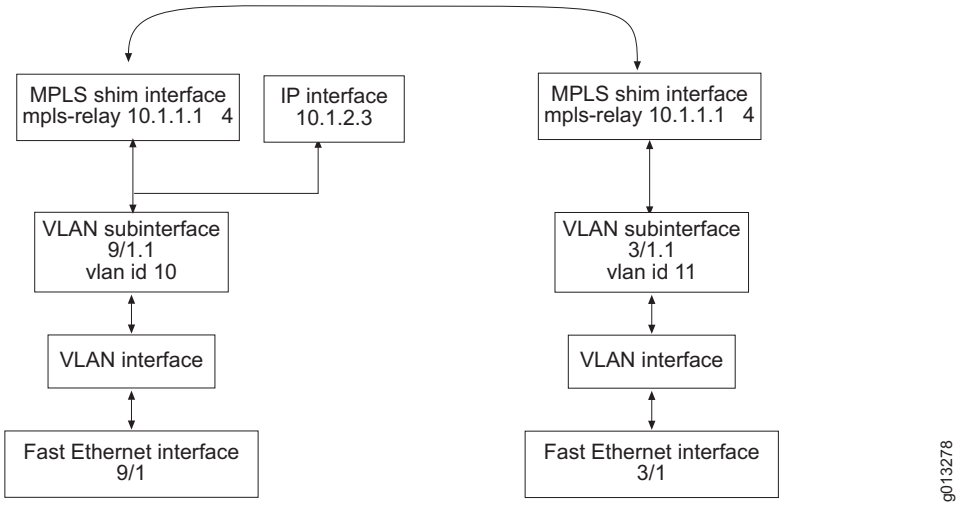
Configuring Local Cross-Connects Between Ethernet/VLAN Interfaces

This section provides an example for configuring a local cross-connect that uses MPLS between two Ethernet/VLAN interfaces.

NOTE: You must use the **mpls-relay** command instead of the **route interface** command to configure a local cross-connect, regardless of the MPLS tunneling method used in the core network.

Figure 117 shows the interface stack that the router builds for this configuration.

Figure 117: Local Cross-Connect Between Ethernet/VLAN Interfaces



To configure the application shown in Figure 117:

1. Configure a local IP address.

   You can use any reachable local IP address. This example uses a loopback interface to provide the local IP address.

   host1(config)#**interface loopback 0**
   host1(config-if)#**ip address 10.1.1.1 255.255.255.255**
   host1(config-if)#**exit**

2. Configure the Ethernet/VLAN interface on one side of the local cross-connect.

   host1(config)#**interface fastEthernet 9/1**
   host1(config-if)#**encapsulation vlan**
   host1(config-if)#**exit**
   host1(config)#**interface fastEthernet 9/1.1**
   host1(config-if)#**vlan id 10**

3. (Optional) If you are configuring a multiservice local cross-connect, assign an IP address and mask to the Ethernet/VLAN interface.

   host1(config-if)#**ip address 10.1.2.3 255.255.255.0**

4. Configure MPLS tunneling on this side of the connection by issuing the **mpls-relay** command.

   When you issue the **mpls-relay** command, you must use a reachable local IP address and the same VC ID value (4) on both sides of the connection.

   ```
   host1(config-if)#mpls-relay 10.1.1.1 4
   host1(config-if)#exit
   ```

5. Configure the Ethernet/VLAN interface on the other side of the local cross-connect.

   ```
   host1(config)#interface fastEthernet 3/1
   host1(config-if)#encapsulation vlan
   host1(config-if)#exit
   host1(config)#interface fastEthernet 3/1.1
   host1(config-if)#vlan id 11
   ```

6. (Optional) If you are configuring a multiservice local cross-connect, assign an IP address and mask to the Ethernet/VLAN interface.

   ```
   host1(config-if)#ip address 10.1.2.4 255.255.255.0
   ```

7. Configure MPLS tunneling on this side of the connection by issuing the **mpls-relay** command.

   You must use a reachable local IP address and the same VC ID value (4) specified in Step 4.

   ```
   host1(config-if)#mpls-relay 10.1.1.1 4
   host1(config-if)#exit
   ```

Related Topics

- **encapsulation vlan** command
- **interface fastEthernet** command
- **interface loopback** command
- **ip address** command
- **mpls-relay** command
- **vlan id** command

Configuring Local ATM Cross-Connects with AAL5 Encapsulation

To create a local cross-connect between two ATM 1483 subinterfaces on the same router, you create a loopback interface, configure your ATM PVCs, and then create an MPLS relay connection from the PVCs to the loopback interface. You do not need to configure any other MPLS commands.

The following commands create an ATM cross-connect between two ATM subinterfaces on the same router.

NOTE: Although this procedure uses AAL5 encapsulation to configure a local cross-connect between two ATM 1483 subinterfaces within the same router, you can also use AAL5 encapsulation when you configure an MPLS pseudowire (tunnel) connection between two ATM VCCs on different routers.

1. Create a loopback interface. All local cross-connects can share the same loopback interface.

   ```
   host1(config)#interface loopback 0
   host1(config-if)#ip address 10.1.1.1 255.255.255.255
   host1(config)#exit
   ```

2. Create an ATM 1483 subinterface and ATM PVC with **aal5all** encapsulation on the ingress interface.

   ```
   host1(config)#interface atm 2/0.1
   host1(config-subif)#atm pvc 1 0 100 aal5all
   ```

3. Create an MPLS relay connection to the loopback interface. Include the address of the loopback interface and a VC ID.

   ```
   host1(config-subif)#mpls-relay 10.1.1.1 2
   host1(config-subif)#exit
   ```

4. Create an ATM 1483 subinterface and ATM PVC with **aal5all** encapsulation on the egress interface.

   ```
   host1(config)#interface atm 2/0.2
   host1(config-subif)#atm pvc 2 0 101 aal5all
   ```

5. Create an MPLS relay connection to the loopback interface. The VC ID must be the same on both sides of the connection.

   ```
   host1(config-subif)#mpls-relay 10.1.1.1 2
   host1(config-subif)#exit
   ```

6. (Optional) Display your configuration.

   ```
   host1#show mpls cross-connects atm
                                                    Cate Peak
   Interface  VPI VCI Interface  VPI VCI VC-ID Encap gory Rate  Status
   ---------- --- --- ---------- --- --- ----- ----- ---- ---- --------
   ATM2/0.1     0 100 ATM2/0.2     0 101     2 AAL5  UBR     0 State UP
   1 local connection(s) found
   ```

Related Topics

- **atm pvc** command
- **interface atm** command
- **interface loopback** command
- **ip address** command
- **mpls-relay** command
- **show mpls cross-connects atm** command
- **vlan id** command

Configuring an MPLS Pseudowire with VCC Cell Relay Encapsulation

The following commands create an ATM layer 2 services over MPLS pseudowire connection between two ATM 1483 subinterfaces on different routers. This procedure uses the **aal0** encapsulation keyword for each ATM PVC to indicate that the router receive raw ATM cells on these circuits and forward the cells without performing AAL5 packet reassembly. The procedure also includes optional steps for configuring nondefault values for the ATM Martini cell packing timers and cell concatenation parameters.

NOTE: Although this procedure uses AAL0 encapsulation to configure an MPLS pseudowire (tunnel) connection between two ATM VCCs on different routers, you can also use AAL0 encapsulation when you configure a local cross-connect between two ATM 1483 subinterfaces within the same router.

To create an MPLS pseudowire connection with VCC cell relay encapsulation:

1. (Optional) Configure values for the three ATM Martini cell packing timers on the ingress router to define the cell collection time threshold.

   host1(config)#**atm mcpt-timers 1500 2500 3500**

2. Configure a loopback interface.

   host1(config)#**interface loopback 0**
   host1(config-if)#**ip address 5.1.1.1 255.255.255.255**
   host1(config)#**exit**

3. Create an ATM 1483 subinterface and ATM PVC with **aal0** encapsulation on the ingress interface.

   host1(config)#**interface atm 2/0.100**
   host1(config-subif)#**atm pvc 100 0 100 aal0**

4. (Optional) Configure the following cell concatenation parameters for the ATM 1483 subinterface:

   - Maximum number of ATM cells that the router can concatenate in a single packet
   - Identifier (1, 2, or 3) of the ATM Martini cell packing timer that you want to use to detect timeout of the cell collection threshold

   host1(config-subif)#**atm cell-packing 100 mcpt-timer 2**

5. Create an MPLS relay connection to the loopback interface on the egress router. The VC ID (1 in this example) must be the same on both sides of the connection.

   host1(config-subif)#**mpls-relay 6.1.1.1 1**
   host1(config-subif)#**exit**

6. Repeat Steps 1 through 5 on the egress router, creating an MPLS relay connection to the loopback interface on the ingress router.

   The values you configure for the ATM Martini cell packing timers and cell concatenation parameters need not be the same on the ingress and egress routers, although matching values are permitted. The virtual connection ID (VC ID) value in the **mpls-relay** command, however, must be the same on the ingress and egress routers.

   host2(config)#**atm mcpt-timers 1500 2500 3500**
   host2(config)#**interface loopback 0**
   host2(config-if)#**ip address 6.1.1.1 255.255.255.255**
   host2(config)#**exit**
   host2(config)#**interface atm 4/0.101**
   host2(config-subif)#**atm pvc 101 0 101 aal0**
   host2(config-subif)#**atm cell-packing 150 mcpt-timer 3**
   host2(config-subif)#**mpls-relay 5.1.1.1 1**
   host2(config-subif)#**exit**

7. (Optional) Use the appropriate **show** commands to verify your configuration.

```
host1#show atm mcpt-timers
ATM Martini cell aggregation timers:
        Timer1: 1500microseconds
        Timer2: 2500microseconds
        Timer3: 3500microseconds

host1#show atm subinterface atm 2/0.100
                                Circuit                              Interface
Interface   ATM-Prot VCD VPI VCI  Type   Encap MTU      Status        Type
---------   -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.100 ATM/MPLS 100   0 100 PVC     AAL0  9180 lowerLayerDown Static

Maximum number of cells per packet: 100
Cell aggregation timeout timer:     2

SNMP trap link-status: disabled
```

```
InPackets:                0
InBytes:                  0
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         0
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found

host2#show atm subinterface atm 4/0.101
                                 Circuit                              Interface
Interface   ATM-Prot VCD VPI VCI  Type   Encap MTU      Status         Type
---------   -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 4/0.101 ATM/MPLS 101   0 101 PVC     AAL0  9180 lowerLayerDown Static

Maximum number of cells per packet: 150
Cell aggregation timeout timer:     3

SNMP trap link-status: disabled

InPackets:                0
InBytes:                  0
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         0
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found
```

Related Topics

- **atm cell-packing** command
- **atm mcpt-timers** command
- **atm pvc** command
- **interface atm** command
- **interface loopback** command
- **ip address** command
- **mpls-relay** command
- **show atm mcpt-timers** command
- **show atm subinterface** command

Configuring HDLC Layer 2 Services

The following commands configure an HDLC layer 2 circuit over MPLS between an E-series router and a remote PE device.

To configure an HDLC layer 2 circuit over MPLS:

1. Configure a serial or POS interface on the ingress router.

   host1(config)#**interface serial 3/1:2/1**

   or

   host1(config)#**interface pos 4/0**

2. Use one of the following methods to create the HDLC layer 2 circuit over MPLS:

   - Use the **mpls-relay** or **route interface** command *without* the **relay-format ppp** keywords. This command causes the router to signal VC-type HDLC on the LDP session and use HDLC encapsulation. Use this command syntax if the traffic carried on the serial or POS interface is any kind of standard HDLC (including PPP) or Cisco HDLC.

     host1(config-if)#**mpls-relay 2.2.2.1 1**

     or

     host1(config-if)#**route interface tunnel mpls:tunnel-to-pe2 1**

   - Use the **mpls-relay** or **route interface** command *with* the **relay-format ppp** keywords. This command causes the router to signal VC-type PPP on the LDP session and use PPP encapsulation instead of the default VC-type HDLC signaling and HDLC encapsulation. Use this command syntax if the traffic carried on the serial or POS interface contains actual PPP packets.

     host1(config-if)#**mpls-relay 2.2.2.1 1 relay-format ppp**

     or

     host1(config-if)#**route interface tunnel mpls:tunnel-to-pe2 1 relay-format ppp**

3. (Optional) Attach an MPLS policy to the HDLC layer 2 circuit by using the **mpls policy** command.

   host1(config-if)#**mpls policy input hdlc-policy**

4. Configure the serial or POS interface and MPLS on the remote PE device.

   The interfaces at either end of the HDLC layer 2 circuit can be different types and have different speeds. For example, you can configure an HDLC layer 2 circuit between a serial interface on a T1 circuit and a POS interface on an OC3 circuit.

Configuring Local Cross-Connects for HDLC Layer 2 Services

You can also configure an HDLC layer 2 circuit in a local cross-connect configuration between serial or POS interfaces within the same router.

The procedure is basically the same for configuring an HDLC layer 2 interface between two PE routers and for a local cross-connect, with the following differences for local cross-connects:

- You must use the **mpls-relay** command instead of the **route interface** command to configure a local cross-connect for HDLC layer 2 services.
- You use the IP address of the local router as the value for the destination IP address (remote address) in the **mpls-relay** command.

Related Topics

- *Local Cross-Connects* on page 497.
- For more information about attaching policies to MPLS layer 2 circuits, see *JUNOSe Policy Management Configuration Guide, Chapter 1, Managing Policies on the E-series Router*.
- For more information MPLS policies, see *JUNOSe Policy Management Configuration Guide, Chapter 2, Creating Classifier Control Lists for Policies*.
- **interface pos** command
- **interface serial** command
- **mpls policy** command
- **mpls-relay** command
- **route interface** command

Configuring CE-Side Load Balancing for Martini Layer 2 Transport

For layer 2 circuits over an MPLS core, each circuit normally has a single shim interface on the local router. In the case of a local cross-connects configuration, each end of the cross-connect has a single shim interface, creating a two-way cross-connect.

Alternatively, a given layer 2 circuit or each end of a local cross-connect can have many shim interfaces. In these cases, traffic destined for the CE routers is load-balanced among the multiple shim interfaces. This is known as CE-side load balancing. In the case of Ethernet/VLANs, CE-side load balancing enables an E-series router to interoperate with an 802.3ad switch.

You can configure load balancing in two different ways. You can configure many shim interfaces with the same peer, VC type, and VC ID. Alternatively, you can use the legacy method of configuring Martini circuits into load-balancing groups.

Configuring Many Shim Interfaces with the Same Peer, VC Type, and VC ID

The **mpls-relay** command enables you to specify the peer and the VC ID explicitly. The VC type is either automatically determined by the layer 2 interface type or you explicitly configure the VC type with the **relay-format** keyword in the **mpls-relay** command.

For example, the following commands create a single layer 2 circuit to the remote peer 10.9.1.3. Load balancing is established on two shim interfaces, fastEthernet 2/0.1 and fastEthernet 3/0.1. The VC type is determined by the layer 2 interface type.

```
host1(config)#interface fast 2/0.1
host1(config-subif)#vlan id 1
host1(config-subif)#mpls-relay 10.9.1.3 200001
host1(config-subif)#exit
host1(config)#interface fast 3/0.1
host1(config-subif)#vlan id 1
host1(config-subif)#mpls-relay 10.9.1.3 200001
```

In this example, the router advertises a single label, 53, to the remote peer, 10.9.1.3, and receives a single label, 55, from the peer, resulting in the following forwarding table:

```
host1:#show mpls forwarding brief
Platform label space

In Label  Owner                                Action
-------- -------- ------------------------------------------------------
53       ldp      l2transport to FastEthernet3/0.1
                  l2transport to FastEthernet2/0.1

L2transport

     Interface        Owner                      Action
------------------- -------- -------------------------------------------------
FastEthernet2/0.1    ldp      swap to 55, push 42 on ATM5/0.1, nbr 10.10.11.5
FastEthernet3/0.1    ldp      swap to 55, push 42 on ATM5/0.1, nbr 10.10.11.5
```

Traffic that arrives on either interface, 2/0.1 or 3/0.1, is forwarded to the remote peer with the same label stack (55, 42). Traffic from the remote peer with label 53 is forwarded to one of the two shim interfaces; the ECMP algorithm determines which of the two shim interfaces receives the traffic.

In the case of a local cross-connects configuration, the following commands illustrate how a three-way cross-connect is created when 10.9.1.2 is a local address:

```
host1(config)#interface atm 6/0.101 point-to-point
host1(config-subif)#mpls-relay 10.9.1.2 600001
host1(config-subif)#exit
host1(config)#interface atm 6/2.101 point-to-point
host1(config-subif)#mpls-relay 10.9.1.2 600001
host1(config-subif)#exit
host1(config)#interface atm 6/2.103 point-to-point
host1(config-subif)#mpls-relay 10.9.1.2 600001
```

This configuration results in the following forwarding table:

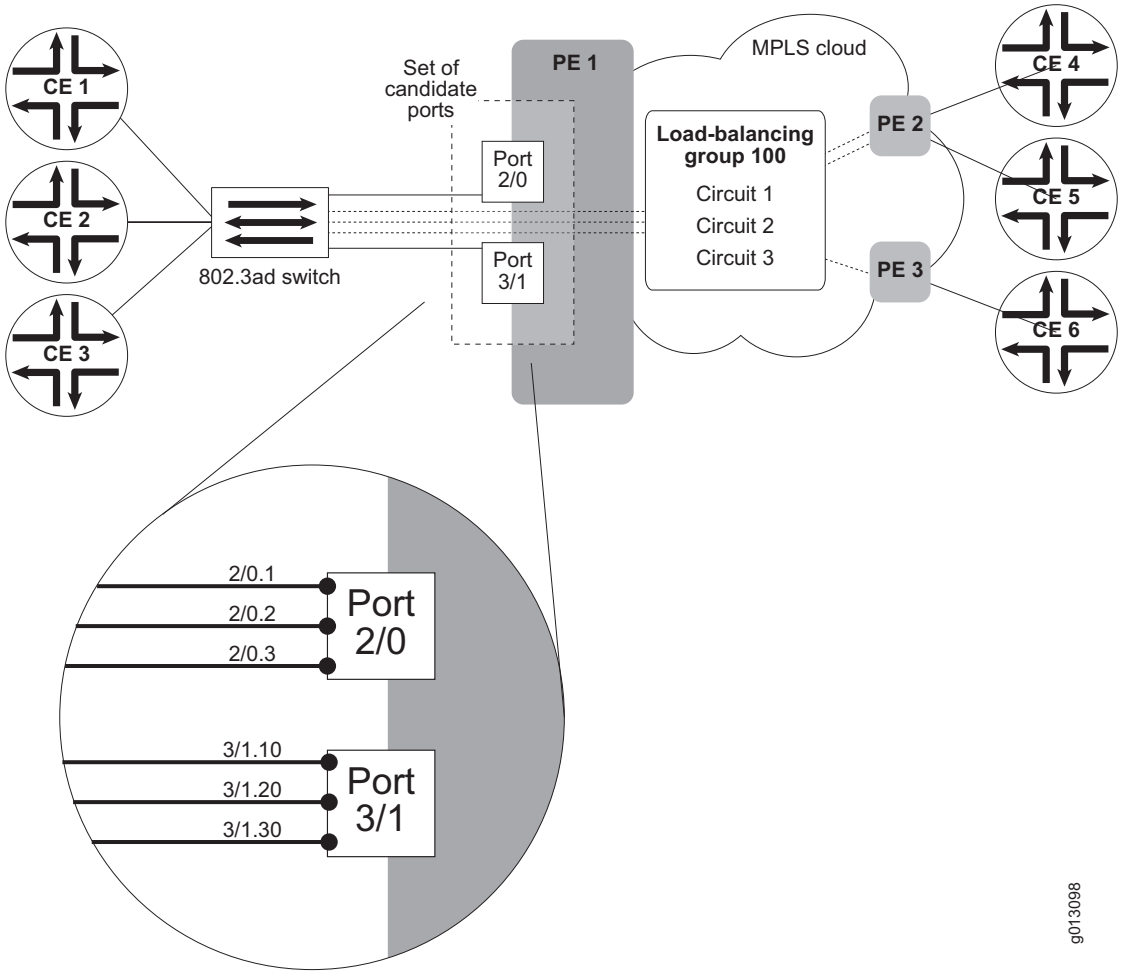
```
host1:two#show mpls forwarding brief
Platform label space
…
L2transport

     Interface        Owner                                                  Action
-------------------- -------- ------------------------------------------------------------------------
ATM6/0.101           ldp      l2transport to ATM6/2.101
                              l2transport to ATM6/2.103
ATM6/2.101           ldp      l2transport to ATM6/0.101
                              l2transport to ATM6/2.103
ATM6/2.103           ldp      l2transport to ATM6/0.101
                              l2transport to ATM6/2.101
```

Traffic that arrives on interface 6/0.101 is forwarded by means of ECMP to both interface 6/2.101 and interface 6/2.103. Traffic that arrives on interface 6/2.101 is forwarded by means of ECMP to interface 6/0.101 and interface 6/2.103. Traffic that arrives on interface 6/2.103 is forwarded by means of ECMP to interface 6/0.101 and 6/2.101.

Configuring Load-Balancing Groups

Load-balancing groups are a legacy method of configuring CE-side load balancing. It was the only method available before Release 7.1.0. Load-balancing groups enable you to configure attributes for a group that are inherited by the member shim interfaces (Figure 118 on page 513).

Figure 118: CE-Side Load-Balancing Topology



In the topology shown in Figure 118, the two Ethernet ports on PE 1 (2/0 and 3/1) are connected to an 802.3ad-compliant switch, and comprise the set of candidate ports. Three VLAN subinterfaces are configured on each port. Load-balancing group 100 includes three Martini circuits, one for each pair of subinterfaces on the ports. That is, three circuits were created: one for the pair 2/0.1 and 3/1.10, one for the pair 2/0.2 and 3/1.20, and one for the pair 2/0.3 and 3/1.30. Each of the three Martini circuits connects to a remote PE router. The remote PE router receives and sends only a single VC label for each circuit.

Traffic from the switch can be received on all ports and sent over the appropriate Martini circuit. For example, traffic from CE 1 to be sent over Martini circuit 1 could arrive on port 2/0 or 3/1 and still be appropriately forwarded.

You configure each circuit for VLAN or S-VLAN subinterfaces that you create across a set of candidate Ethernet ports. The router distributes traffic from the core through the candidate ports used by the load-balancing group. If a port is disabled, traffic is redistributed to a working port.

MPLS Interfaces and Labels

When a layer 2 interface is added to a load-balancing group circuit, an MPLS shim interface is automatically created on top of that layer 2 interface. The attributes of the shim interface are inherited from the load-balancing group and cannot be configured.

All MPLS shim interfaces within a load-balancing group circuit point to the same MPLS next hop. Traffic arriving from the CE router over this set of MPLS shim interfaces is merged into a single LSP and sent to the remote PE router.

The VC in label for the layer 2 circuit points to a single ECMP MPLS next hop. The legs of this ECMP next hop are the member shim interfaces of the load-balancing group circuit. Consequently, ECMP is used to forward traffic arriving from the core across the MPLS shim interfaces to the CE router.

Configuring Load-Balancing Groups

You configure Martini circuits with load-balancing groups in a separate mode, in which the member layer 2 subinterfaces are entered one by one.

For example, the following commands configure two Martini circuits to different PE routers, in the same load-balancing group 100, sharing the candidate Ethernet ports 2/0 and 3/0:

```
host1(config)#mpls l2transport load-balancing-group 100 mpls-relay 10.1.1.1 30
host1(config-mpls-l2-group)#member interface fast 2/0.1
host1(config-mpls-l2-group)#member interface fast 3/0.100

host1(config)#mpls l2transport load-balancing-group 100 mpls-relay 10.2.2.2 22
host1(config-mpls-l2-group)#member interface fast 2/0.2
host1(config-mpls-l2-group)#member interface fast 3/0.200
```

Adding a Member Interface to a Group Circuit

You specify the lower interface as a member interface, as in the following example.

```
host1(config)#mpls l2transport load-balancing-group 100 mpls-relay 2.2.2.2 202
host1(config-mpls-l2-group)#member interface fast 2/0.500
```

Removing Member Subinterfaces from a Circuit

To remove a member subinterface from a circuit, either issue the **no member interface** command (from the L2 Transport Load-Balancing-Group Configuration mode) or issue the **no mpls-relay** command at the VLAN or S-VLAN subinterface level. Each of the following examples removes member Fast Ethernet subinterface 13/0.2 from the load-balanced Martini circuit:

```
host1(config)#mpls l2transport load-balancing-group 100 mpls-relay 2.2.2.2 202
host1(config-mpls-l2-group)#no member interface fast 13/0.2
```

or

```
host1()#interface fast 13/0.2
host1(config-subif)#no mpls-relay
```

Related Topics

- **member interface** command
- **mpls l2transport load-balancing-group** command

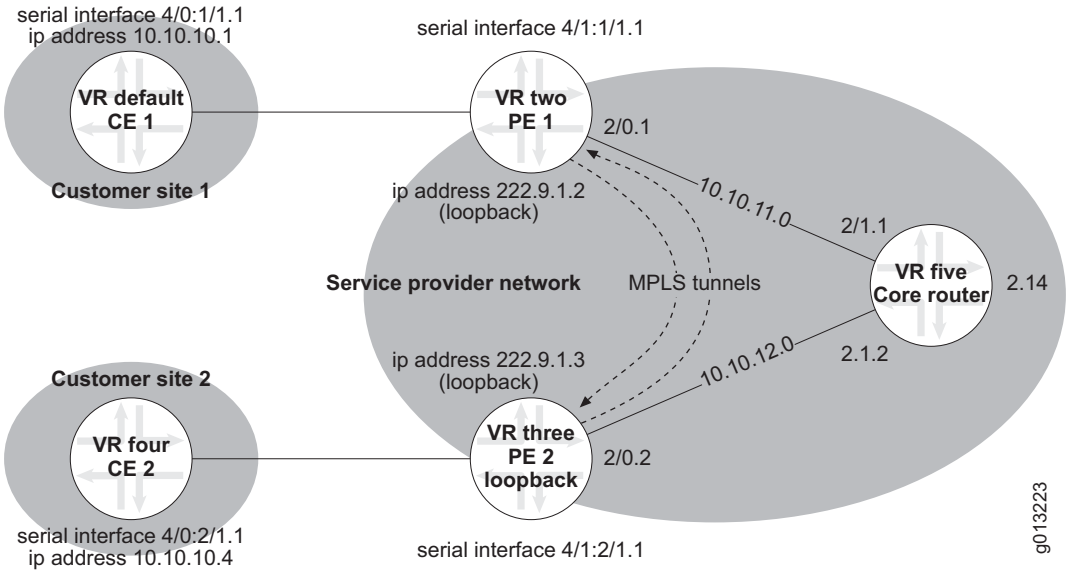
Frame Relay over MPLS Configuration Example

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 119 further explains the configuration script.

NOTE: The **route interface** command is used toward the end of the configuration script. You can substitute the **mpls-relay** command, depending on the tunneling method best for your environment.

Figure 119: Sample Frame Relay over MPLS Configuration




```
hostname "host 1"
exception protocol ftp anonymous null
!--------------------------------------------------------------------------
!Configure CT3 interfaces 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 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 destination 222.9.1.3

int ser 4/1:1/1.1
route interface tunnel mpls:2 45
```