



Configuring MPLS and VPN

This chapter describes MPLS and VPN features used with the RPM in the MGX 8850 switch and covers the following topics:

- [MPLS Overview](#)
- [VPN Overview](#)
- [How VPNs Work](#)
- [Configuring a VPN](#)
- [Configuring an ATM PVC for Carrying Packet-Based MPLS](#)
- [Configuring a Virtual Path Tunnel](#)
- [RPM Configuration Examples for MPLS-based Virtual Private Networks](#)
- [Two PE–Three CE Configuration, RIP-connected CE Routers](#)
- [Two PE–Three CE Configuration, Static routing to CE Routers](#)
- [Support for LDP](#)
- [Support for Multi-VC on the RPM-PR](#)

For more information on MPLS, refer to the *Cisco MPLS Controller Software Configuration Guide*. For MPLS and VPN commands, refer to the *Cisco MPLS VPN Feature Guide*.

MPLS Overview

This section describes Multiprotocol Label Switching (MPLS) and the role of the Route Processor Module (RPM) as an Edge Label Switch Router (LSR) within the MGX 8850 switch. It also includes a description of how the RPM, as an Edge LSR, acts as an MPLS “feeder” to the BPX 8650 switch.

The labels used to forward packets are negotiated using Label Distribution Protocol (LDP) or Tag Distribution Protocol (TDP). In this context, the RPM functions as an Edge LSR to receive and label IP packets.

There are two different modes of MPLS operation, and the RPM supports both of them.

- Frame-based
- ATM cell-based

Frame-Based MPLS

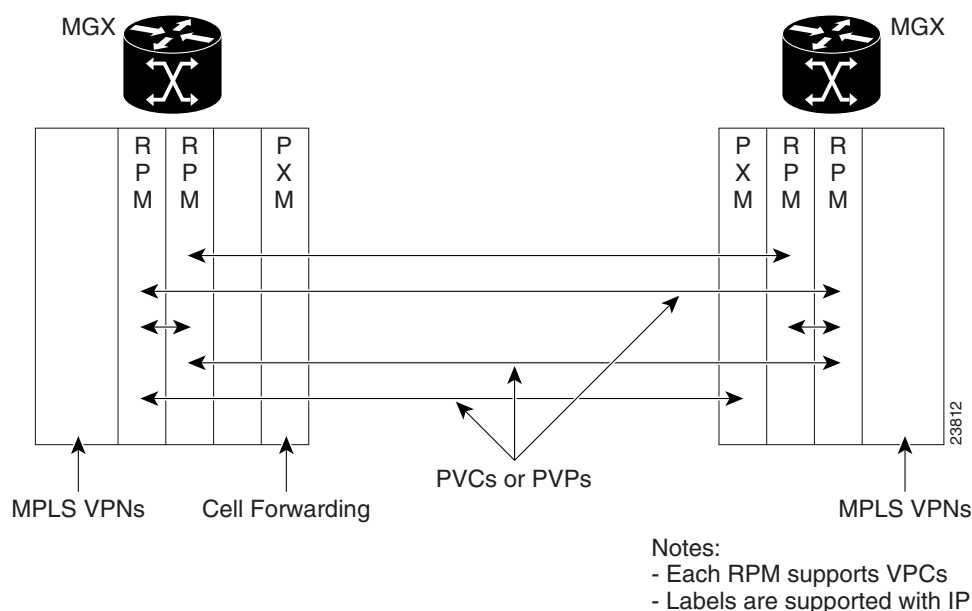
In the frame-based mode, all MPLS frames between two MPLS devices move within a single point-to-point channel. MPLS information is carried as a separate header in each frame. The RPM supports frame-based MPLS within permanent virtual circuits (PVCs). There are two typical uses of frame-based MPLS with RPMs.

- It is possible to use RPMs as Edge LSRs linked by PVCs, without using MPLS on the core switches. In this type of network, PVCs are typically used to connect every RPM in a network to every other RPM using a full mesh of PVCs. This setup is illustrated in [Figure 7-1](#).

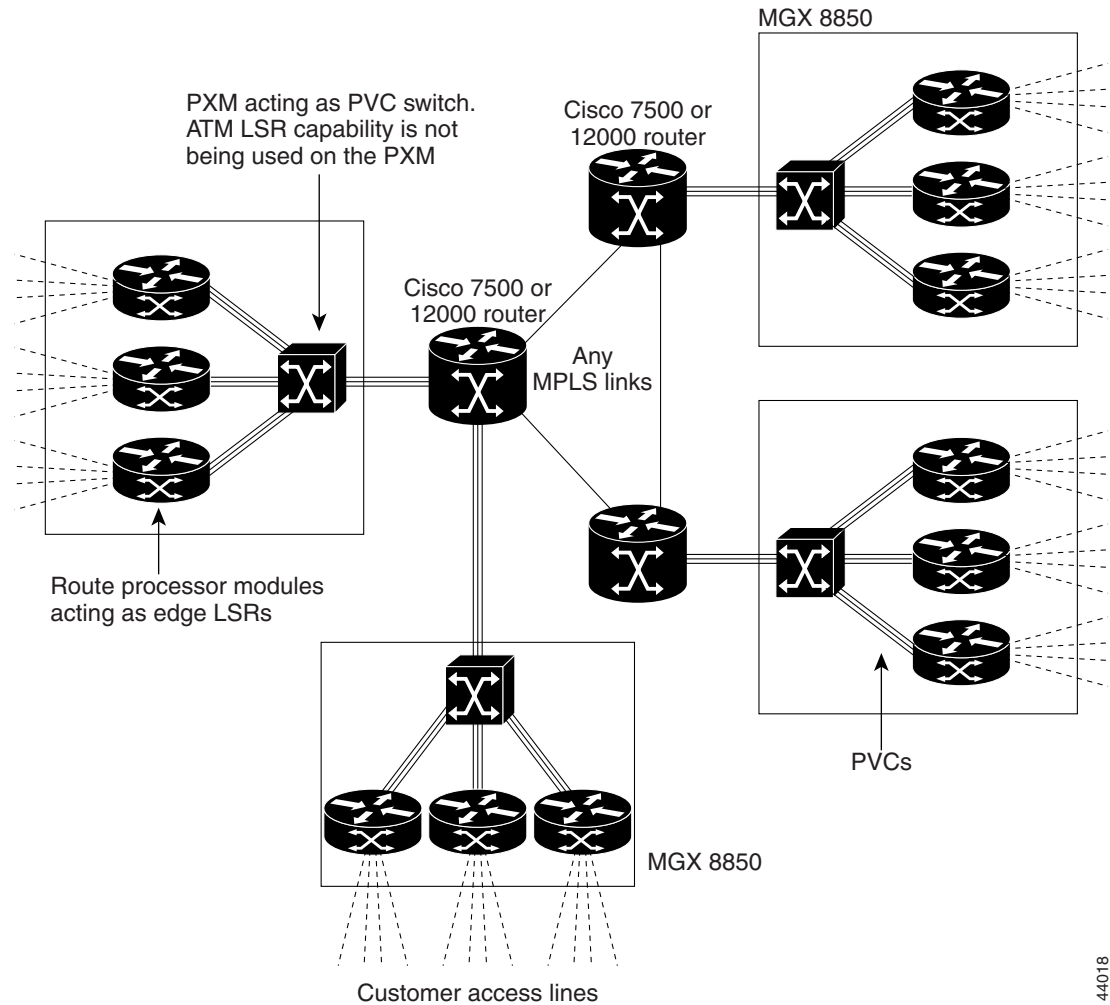

Note

Careful network design is required to use fully-meshed networks with more than about 30 Edge LSRs. Contact TAC if you need assistance.

Figure 7-1 MPLS for the MGX 8850 in a Fully-Meshed PVC or PVP Configuration



- PVCs running packet-based MPLS can also be used to connect RPMs to other Cisco packet-based MPLS devices, such as Cisco 7500-series or 12000-series routers. In such a network, the RPMs would typically act as Edge LSRs, and the other routers as (core) LSRs. This setup is illustrated in [Figure 7-2](#).

Figure 7-2 MPLS for the MGX 8850 with a Router Network Core

The RPM also supports packet-based MPLS over its Ethernet and Fast Ethernet port adaptors.

ATM MPLS

Numerous VCs, known as MPLS Label VCs (LVCs), are used to connect a pair of ATM MPLS devices. The LVCs are established under the direct control of MPLS signaling, and each LVC corresponds to a distinct MPLS label value.

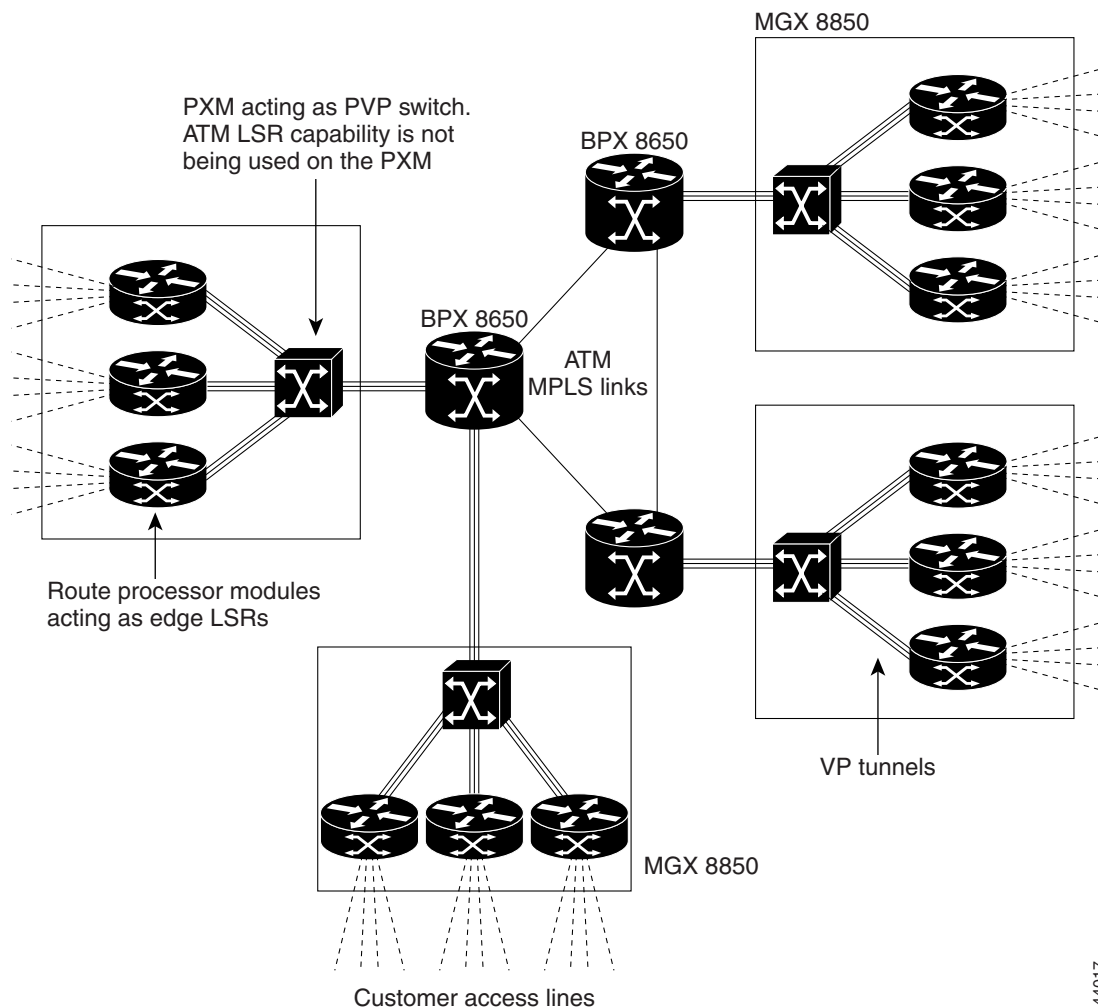
In the current release, the MGX 8850 does not support Label Switch Controller (LSC) function. When the MGX 8850 does support LSC function, the switch with an LSC will be able to set up LVCs directly on ATM trunks. Until then, LVCs must be set up within Permanent Virtual Paths (PVPs). These PVPs act as a type of "virtual trunk," and are otherwise known as MPLS VP Tunnels.

ATM MPLS, by way of VP Tunnels, can work with RPMs in two different ways:

- It is possible to use RPMs as Edge LSRs linked by PVPs, without using ATM MPLS on the core switches, as illustrated in Figure 7-1. However, there is little advantage to the use of PVPs in meshed MPLS configurations, and the use of PVCs is simpler.

- A more usual use of ATM MPLS is when the core ATM switches support ATM MPLS, such as in BPX 8650 switches. This is illustrated in [Figure 7-3](#).

Figure 7-3 MPLS for the MGX 8850 with an ATM MPLS Network Core, Using no LSC on the MGX 8850



44017

In any of these network configurations, the RPM also supports an MPLS feature, MPLS Virtual Private Networks (VPNs). In MPLS VPN operation, the RPM will act as a Provider Edge (PE) router. PE router function is a combination of MPLS Edge LSR function and the use of the Border Gateway Protocol (BGP) v4 with Multiprotocol Extensions to carry routing information for the VPNs.

MPLS in an MGX 8850 Switch

On the MGX 8850 platform, MPLS provides an IP solution without the cost of Layer 2 management. In contrast to IP over ATM, MPLS reduces the customer's network management. Also, MPLS provides the same level of privacy as does Frame Relay or ATM.

For a description of how the RPM acts as an Edge LSR to support MPLS feeder functionality, refer to the section, [““System Block Diagram” section on page 7-5](#).

Features

The RPM supports the following features for MPLS:

- Edge LSR functionality in the RPM.
- Ability to run MPLS traffic over a PVC between RPM Edge LSRs or between an RPM and routers such as the Cisco 7500.
- Ability to run MPLS traffic over a PVP between RPM Edge LSRs or from an RPM Edge LSR to a BPX 8650 switch with a Label Switch Controller (LSC).
- Ability to run packet-based MPLS traffic over the RPM Ethernet and Fast Ethernet port adaptor ports.
- MPLS PVC or PVP connections limits that fall within the established connection limits for the software release.

These connection limits stem from the MGX 8850 platform, not the MPLS feature. However, if the platform imposes the limit, the MPLS feature does not support any capacity beyond them.

- MPLS VPN.
- Protocol support provided by IGP-OSPF, RIP, EIGRP, IS-IS.
- VPN addition provided by BGP, RIPv2, static routes for PE-CE links.
- Ability to support ~2000 interface descriptor blocks (IDBs) for the RPM-PR and ~700 IDBs for the RPM/B.



Note If an interface does not contain any sub-interfaces, then it constitutes one sub-interface for the purpose of this limit.

- RPM switch interface can support 3,840 VCs (PVCs and/or LVCs) and 255 PVPs.



Note Due to the limit of 2000 sub-interfaces, the total of PVCs is limited to less than 3840 in some configurations, and it is not always possible to use all 255 PVPs.

- Packet-based MPLS over PVCs.
- ATM MPLS over PVPs.

The RPM also can support ATM MPLS without the use of PVPs. This feature will be supported once the MGX 8850 has LSC support.

Limitations

There may be no automatic switchovers in an MGX 8850 switch or node

System Block Diagram

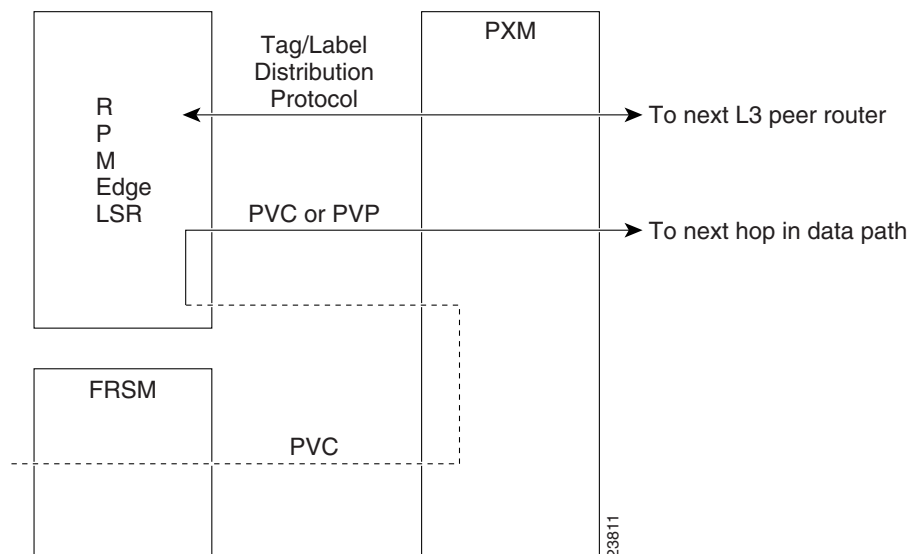
The RPM acts as an Edge LSR, with user data entering an FRSM card, flowing on a PVC to an RPM, to a PVP connection or PVC (see [Figure 7-4](#)), and on to the next hop in the data path. In this example, the FRSM associates the data with a PVC. The other end of the PVC terminates at the RPM switch port.

The RPM receives the packets and optionally provides Layer 3 services. Then, based on the Layer 3 destination address, the RPM forwards the packet to a PVP or PVC.

In the case where a PVP is used, the Edge LSR uses the VCI field in the ATM cell header for the MPLS label. The VPI value is specified statically when the PVP is provisioned.

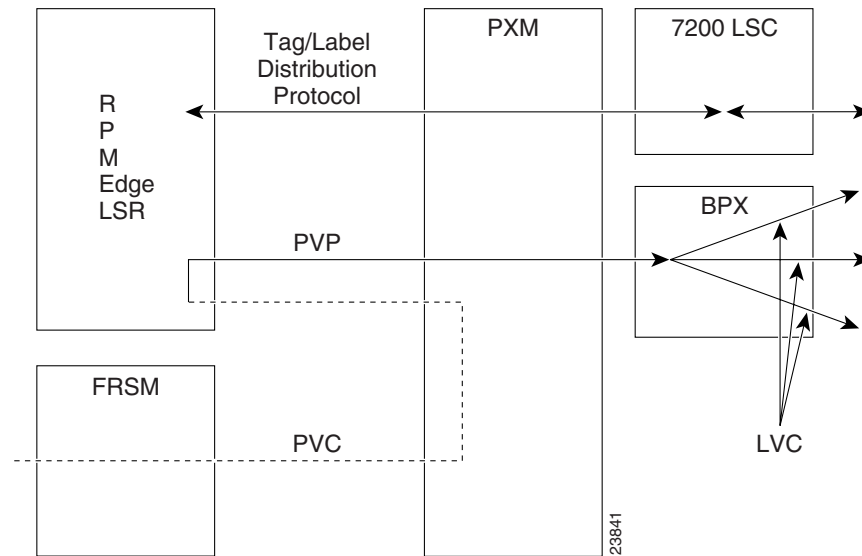
In the case where a PVC is used, the Edge LSR labels the packet and then segments it into ATM cells. The VPI/VCI values are specified statically when the PVC is provisioned. Therefore, the label exists only in the payload of the ATM cell.

Figure 7-4 Edge LSR Function, Mesh Configuration



In [Figure 7-5](#), an MPLS-enabled core is shown. The LSC creates cross-connects in an ATM switch so that labeled packets can be forwarded through the switch. The BPX 8650 switch and the RPM Edge LSR use LDP/ TDP to identify labels. Because the connection between the BPX 8650 switch and the Edge LSR is a PVP, the VPI is static. This is an MPLS VP tunnel and therefore the VCI field is used as the label. The LSC establishes cross-connects in the BPX 8650 switch so VC connections in the PVP are broken out and individually switched. Such label forwarding is not possible when a PVC is used, because the label does not exist in the ATM cell header.

Figure 7-5 MGX 8850 Edge LSR with MPLS-Enabled BPX Core



VPN Overview

Virtual Private Networks (VPNs) provide the appearance, functions, and usefulness of a dedicated private network. The VPN feature for MPLS allows a Cisco IOS network to deploy scalable IPv4 Layer 3 VPN backbone service with private addressing, controlled access, and service-level guarantees between sites.

VPNs are supported by service provider networks over which labeled packets are forwarded from RPM Edge LSRs to other RPM Edge LSRs. A VPN service creates multiple private network environments within the public infrastructure. Service providers can use VPNs to target a given clientele and deliver individualized private network services to that clientele in a secure IP environment by using the public infrastructure.

Requirements

The requirements for an effective VPN are

- **Privacy**—All IP VPN services offer privacy over a shared (public) network infrastructure, the most well known solution of which is an encrypted tunnel. An IP VPN service must offer private addressing, where addresses within a customer private network do not need to be globally unique.
- **Scalability**—IP VPN services must scale to serve hundreds of thousands of sites and users. An IP VPN service should also serve as a management tool for service providers to control access to services, such as closed user groups for data and voice services. Controlled access places performance limits upon authorized programs, processes, or other systems in a network.
- **Flexibility**—IP VPN services must accommodate any-to-any traffic patterns and be able to accept new sites quickly, connect users over different media, and meet transport and bandwidth requirements of new intranet applications.

- **Predictable Performance**—Intranet applications supported by an IP an IP VPN service require different classes of service. The service level performance between customer sites must be guaranteed. Examples include widespread connectivity required by remote access for mobile users and sustained performance required by interactive intranet applications in branch offices.

MPLS VPN Features

Beyond the functions of an IP VPN, the VPN features for MPLS allow a Cisco IOS network to deploy the following scalable IPv4 Layer 3 VPN backbone services:

- **Connectionless Service**—MPLS VPNs are connectionless. They are less complex because they do not require tunnels or encryption to ensure network privacy.
- **Centralized Service**—VPNs in Layer 3 privately connect users to intranet services and allow flexible delivery of customized services to the user group represented by a VPN. VPNs deliver IP services such as multicast, QoS, and telephony support within a VPN, and centralized services like content and web hosting. Combinations of services can be customized for individual customers.
- **Scalability**—MPLS based VPNs use Layer 3 connectionless architecture and are highly scalable.
- **Security**—MPLS VPNs provide the same security level as connection-based VPNs. Packets from one VPN cannot accidentally go to another VPN. At the edge of a provider network, incoming packets go to the correct VPN. On the backbone, VPN traffic remains separate.

**Note**

Spoofing of a PER is nearly impossible because incoming packets are IP packets and must be received on an interface or subinterface uniquely identified with a VPN tag.

- **Easy to Create**—MPLS VPNs are connectionless. It is easy to add sites to intranets and extranets and to form closed user groups. A given site can have multiple memberships.
- **Flexible Addressing**—MPLS VPNs provide a public and private view of addresses, enabling customers to use their own unregistered or private addresses. Customers can freely communicate across a public IP network without network address translation (NAT).
- **Straightforward Migration**—MPLS VPNs can be built over multiple network architectures, including IP, ATM, Frame Relay, and hybrid networks. There is no requirement to support MPLS on the customer edge (CE) router.

Supported Platforms

All Cisco routers from the 3600 series up support VPNs, as do the MGX 8850 node with RPM, the 6400 series, and several other devices. Any LSR-capable platform can serve in the backbone. In addition to devices already mentioned, the LS 1010 ATM Switch, 8540 MSR, and the BPX 8650 switch are supported. Non-MPLS capable switches can also be used, as they can carry MPLS over PVCs or PVPs.

How VPNs Work

Each VPN is associated with one or more VPN routing/forwarding instances (VRFs), which defines a VPN at a customer site attached to a PE router. A VRF table contains the following information.

- IP routing table
- Derived Cisco Express Forwarding table
- Set of interfaces that use the forwarding table
- Set of rules and routing protocol variables that determine what goes into the forwarding table

VPNs for MPLS

A customer site can be a member of multiple VPNs. However, a site can be associated with only one VRF. A customer site's VRF contains all routes available to the site from the associated VPNs.

The IP routing table and CEF table for each VRF store packet forwarding information. (Together, these tables are analogous to the forwarding information base [FIB] used in MPLS.) A logically separate set of routing and CEF tables is constructed for each VRF. These tables prevent packets from being forwarded outside a VPN and prevent packets outside a VPN from being forwarded to a router within the VPN.

VPN Route-Target Communities and Export and Import Lists

The distribution of VPN routing information is controlled through the use of VPN route-target communities, implemented by Border Gateway Protocol (BGP) extended communities. Distribution works as follows:

- When a VPN route is injected into BGP, it is associated with a list of VPN route-target communities. This list is set through an export list associated with the VRF from which the route was learned.
- Associated with each VRF is an import list of route-target communities, which defines values to be verified by the VRF table before a route is deemed eligible for import into the VPN routing instance. For example, if a given VRF's import list includes community-distinguishers A, B, and C, then any VPN route carrying A, B, or C is imported into the VRF.

iBGP Distribution of VPN Routing Information

A LER learns an IP prefix from a CE router through static configuration, a BGP session, RIP, or OSPF. The LER then generates a VPN-IPv4 (vpnv4) prefix by linking an 8-byte route distinguisher to the IP prefix. The VPN-IPv4 address uniquely identifies hosts within each VPN site, even if the site uses globally non-unique (unregistered private) IP addresses. The route distinguisher used to create the VPN-IPv4 prefix is specified by a configuration command on the LER.

BGP uses VPN-IPv4 addresses to distribute network reachability information for each VPN within a service provider network. In building and maintaining routing tables, BGP sends routing messages *within* (interior BGP or iBGP) or *between* IP domains (exterior BGP or eBGP).

BGP propagates vpnv4 information using BGP multiprotocol extensions for handling extended addresses. Refer to RFC 2283, *Multiprotocol Extensions for BGP-4*. BGP propagates reachability information (expressed as VPN-IPv4 addresses) among PE routers; reachability information for a given VPN is propagated only to members of that VPN. BGP multiprotocol extensions identify valid recipients of VPN routing information.

Label Forwarding

Based on the routing information stored in each VRF's IP routing and CEF tables, MPLS uses extended VPN-IPv4 addresses to forward packets to their destinations.

To achieve this, an MPLS label is associated with each customer route. The PE router assigns the route originator's label and directs data packets to the correct CE router. Tag forwarding across the provider backbone is based on dynamic IP paths or Traffic Engineered paths.

A customer data packet has two levels of labels attached when it is forwarded across the backbone.

- The top label directs the packet to the correct PE router.
- The second label indicates how that PE router should forward the packet.

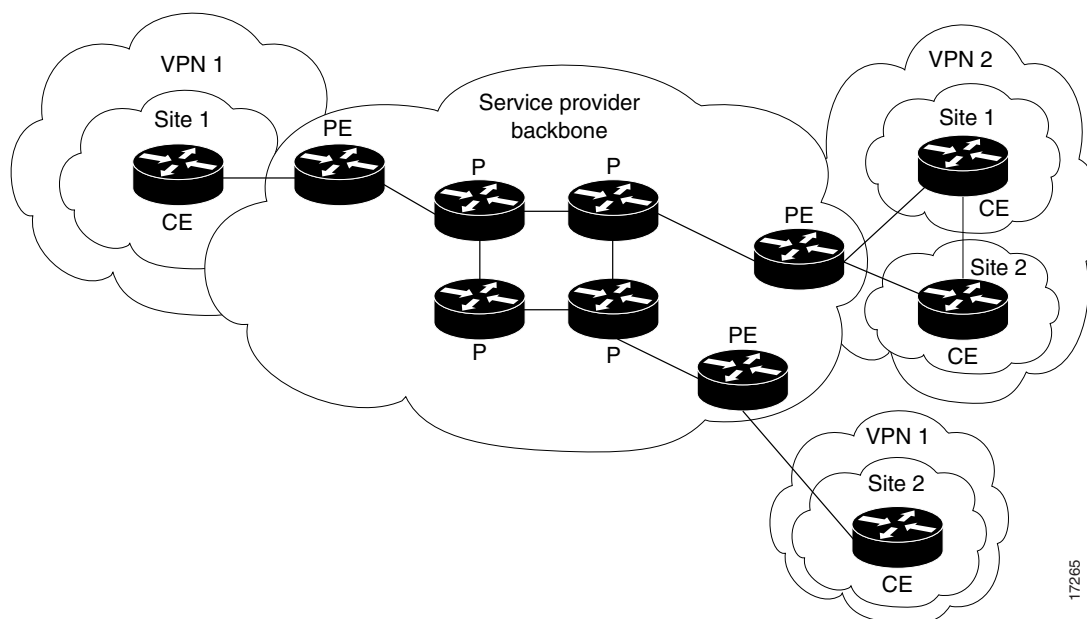
The PE router associates each CE router with a forwarding table that contains only the set of routes that are available to that CE router.

Example VPN Topologies

A VPN contains customer devices attached to CE routers. These customer devices use the VPN to exchange data. Only the PE routers are aware of the VPN.

An example of a VPN with a service provider (P) backbone network, service PE routers, and CE routers is shown in [Figure 7-6](#).

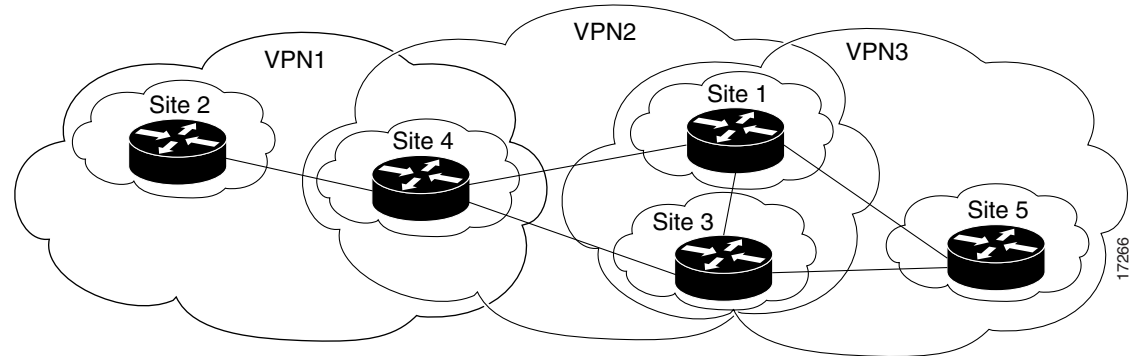
Figure 7-6 VPN with a Service Provider (P) Backbone Network



17265

Three VPNs communicating with five customer sites are shown in [Figure 7-7](#). Notice that sites 1, 3, and 4 are members of two VPNs.

Figure 7-7 VPNs Communicate with Customer Sites



Configuring a VPN

This section explains how to configure the RPM for VPN operation, including listing the prerequisites for VPN configuration and the configuration procedures.

Prerequisites for VPN Operation

The network must be running the following Cisco IOS services before you can configure VPN operation:

- CEF switching in every tag-enabled router
- MPLS connectivity among all provider edge (PE) routers with VPN service or MPLS in all provider backbone (P) routers
- MPLS with VPN code in all provider routers with a VPN PE routers
- BGP in all routers providing a VPN service

Complete the following tasks before you configure VPN operation:

- Turn on Cisco Express Forwarding (CEF).
- Configure MPLS.
- Turn on BGP between provider routers for distribution of VPN routing information.

Configuring VPN Operation

The following sections describe how to configure routing protocols and create VPFs for a VPN. The commands used in the tasks are described in more detail in the “[VPN Overview](#)” section. Perform the following four tasks to configure and verify VPNs in your network:

1. Configure VRFs and associate interfaces with VRFs.
2. Configure BGP between provider routers for distribution of VPN routing information.

3. Configure import and export routes to control the distribution of routing information.
4. Verify VPN operation.

Configuring VRFs

To create a VRF, perform the following steps on the provider edge router:

- Step 1** Enter VRF configuration mode and specify the VRF to which subsequent commands apply.

```
RPM(config)# ip vrf vrf-name
```

- Step 2** Define the instance by assigning a name and an 8-byte route distinguisher.

```
RPM(config-vrf)# rd route-distinguisher
```

- Step 3** Associate interfaces with the VRF.

```
RPM(config-if)# ip vrf forwarding vrf-name
```

- Step 4** If BGP is used between the PE and a VRF CE, configure BGP parameters for the VRF CE session.

```
RPM(config)# address-family ipv4 vrf name
RPM(config-af)# aggregate-address
RPM(config-af)# auto-summary
RPM(config-af)# default-information originate
RPM(config-af)# default-metric ...
RPM(config-af)# distance ...
RPM(config-af)# distribute-list ...
RPM(config-af)# network ...
RPM(config-af)# neighbor ...
RPM(config-af)# redistribute ...
RPM(config-af)# synchronization
RPM(config-af)# table-map...
```



Note To ensure that addresses learned from CE routers via BGP are properly treated as VPN IPv4 addresses on a PE router, enter the command **no bgp default ipv4-activate** before configuring any CE neighbors. See Step 2 and Step 3 in the next section, “[Configuring BGP](#)” below.

- Step 5** If RIP is used between the PE and VRF CEs, configure RIP parameters (in a VRF address-family submode).



Note The default for auto-summary and synchronization in VRF address-family submode is off.

```
RPM(config)# address-family ipv4 vrf name
RPM(config-af)# auto-summary
RPM(config-af)# default-information originate
RPM(config-af)# default-metric ...
RPM(config-af)# distance ...
RPM(config-af)# network ...
RPM(config-af)# offset-list ...
RPM(config-af)# redistribute ...
```

- Step 6** Exit from the address family config mode.

```
config-af# exit-address-family
```

Step 7 Configure static routes for the VRF.

```
RPM(config)# ip route [vrf vrf-name] destination <interface> ip_address
```

Configuring BGP

To configure router address families, define sessions, and set global variables for routing protocols, perform the following steps with the PE router in configuration mode.

Step 1 Configure BGP address families.

```
RPM(config)# address-family {ipv4 | vpnv4}[unicast | multicast]
```

Step 2 Define BGP sessions.

```
RPM(config-af)# neighbor address | peer-group} remote-as as-number
RPM(config-af)# neighbor address | peer-group} update-source interface
RPM(config-af)# neighbor peer-group peer-group
RPM(config-af)# neighbor address peer-group peer-group
```

Step 3 Activate a BGP session by entering the **no bgp default ipv4-activate** to prevent automatic advertisement of **address family IPv4** for every neighbor. This command is required on a PE that establishes BGP sessions with CE routers. To enable advertisement of IPv4 prefixes for a particular neighbor, enter address-family mode for IPv4 then use the **neighbor...activate** command for the neighbor.

```
RPM(config)# no bgp default ipv4-activate
```

For a particular address family, use **neighbor... activate**.

```
RPM(config-af)# [no] neighbor address |peer-group} activate
```

Step 4 Enter optional BGP global commands that affect all address families.

```
RPM(config)# bgp always-compare-med
RPM(config)# bgp bestpath ...
RPM(config)# bgp client-to-client reflection
RPM(config)# bgp cluster-id ...
RPM(config)# bgp confederation ...
RPM(config)# bgp default local-preference ...
RPM(config)# bgp deterministic-med ...
RPM(config)# bgp fast-external-fallover ...
RPM(config)# bgp log-neighbor-changes
RPM(config)# bgp redistribute-internal
RPM(config)# bgp router-id ...
RPM(config)# timers bgp ...
```

Step 5 Enter the BGP configuration commands for address family IPv4.

All BGP configuration commands supported in previous versions of IOS are valid for address family IPv4 unicast. These commands affect either all IPv4 instances or the default IPv4 routing table. For backward compatibility, these commands can be entered in either router config mode or in address family mode for ipv4 unicast. See Step 3 for information on the command **no bgp default ipv4-activate**.

```
RPM(config)# bgp ...
```

Step 6 Enter the BGP configuration commands for address family VPNv4.

```
RPM(config-af)# bgp dampening ...
RPM(config-af)# neighbor ...
RPM(config-af)# neighbor address | peer-group}activate
```

- Step 7** To configure iBGP to exchange VPNv4 Network Layer Reachability Information (NLRI) (between PE router and route reflector or between PE routers), first define an iBGP BGP session.



Note To ensure that VPN packets are properly tag forwarded between the PE routers, specify loopback addresses for the neighbor address and the update-source interface.

```
RPM(config)# neighbor address remote-as as-number
RPM(config)# neighbor address update-source interface
```

- Step 8** Activate the advertisement of VPNv4 NLRI.

```
RPM(config)# address-family vpnv4
RPM(config-af)# neighbor address activate
```

Configure Import and Export Routes

To configure VRF route target extended communities and import route maps, perform the following steps with the PE router in configuration mode.

- Step 1** Enter VRF configuration mode and specify a VRF.

```
RPM(config)# ip vrf vrf-name
```

- Step 2** Import routing information from the specified extended community.

```
RPM(config-vrf)# route-target import community-distinguisher
```

- Step 3** Export routing information to the specified extended community.

```
RPM(config-vrf)# route-target export community-distinguisher
```

- Step 4** Associate the specified route map with the VRF being configured.

```
RPM(config-vrf)# import map route-map
```

Checking the VRFs

Perform the following steps to verify VPN configuration:

- Step 1** Display the set of defined VRFs and the interfaces associated with each one.

```
RPM# show ip vrf
```

- Step 2** Display detailed information about configured VRFs, including the import and export community lists.

```
RPM# show ip vrf detail
```

- Step 3** Display the IP routing table for VRF.

```
RPM# show ip route vrf vrf-name
```

- Step 4** Display the routing protocol information associated with a VRF.

```
RPM# show ip protocols vrf vrf-name
```

- Step 5** Display the CEF forwarding table associated with a VRF.

```
RPM# show ip cef vrf vrf-name
```

- Step 6** Display the VRF table associated with an interface. Use either of the following commands:

```
RPM# show ip interface interface-number
RPM# show cef interface interface-number
```

- Step 7** Display VPNv4 NLRI information. The keyword **all** displays the entire database. The keyword **rd** displays NLRIs that match the specified route distinguisher. The keyword **vrf** displays NLRIs with the specified VRF. Add the keyword **tags** after any of the other keywords and arguments to list the tags distributed with the VPNv4 NLRIs.

```
RPM# show ip bgp vpnv4 all [tags]
RPM# show ip bgp vpnv4 rd route-distinguisher [tags]
RPM# show ip bgp vpnv4 vrf vrf-name [tags]
```

- Step 8** Display tag forwarding entries that correspond to VRF routes advertised by this router.

```
RPM# show tag-switching forwarding vrf vrf-name [prefix mask/length] [detail]
```

- Step 9** You can also use **ping** or **traceroute**.

```
RPM# ping vrf vpn 1.1.1.1
```

where 1.1.1.1 is the destination address.

- Step 10** Enter the following **telnet** command to check the VRFs.

```
telnet 1.1.1.1 /vrf vpn
```

Configuring an ATM PVC for Carrying Packet-Based MPLS

This section describes how to add PVC connections for packet-based MPLS connectivity in a network core, using the command line interface of IOS. With packet-based MPLS over PVCs, a PVC is created to every other RPM in the network. In this simple example, there are just two RPMs, both in the same MGX 8850, so there is one PVC. The PVC connects the RPMs through the Cellbus. Therefore, for each connection from an RPM to another MPLS device over a PVC, you must create:

- A subinterface connected to the PVC, with MPLS enabled.
- An ATM PVC between the RPM and the cellbus.

From either a control terminal connected to the RPM or through the MGX 8850 switch, go to the IOS command line interface, then perform the following steps (unless they have already been done):

- Step 1** Enable Cisco Express Forwarding (CEF).
- Step 2** Specify a routing protocol.
- Step 3** Specify the area in which the RPM can communicate.
- Step 4** Configure the subinterface.
- Step 5** Create an ATM PVC so the RPM can communicate with the cellbus.
- Step 6** Ping the far end router to verify connectivity.

Step 7 Add VPN configuration (this is described later).

The following items must be the same on the RPMs at both ends.

- Routing protocol (preferably OSPF or IS-IS)
- Routing area (specified by a network area number)

All configuration tasks require you to enable configuration mode, as follows.

```
TErpm4>
TErpm4>ena
Password:
```

To establish user-connections for MPLS edge routing, perform the following steps:

Step 1 Enter the password.

The prompt changes to include a pound sign (#). At this level, you can show aspects of the RPM per the IOS commands and list the configuration commands, but you cannot actually execute configuration commands.

```
TErpm4#
```

Step 2 Enter configuration mode and include a specification for the configuration source. The source is terminal, memory, or network. The syntax and resulting display are,

```
TErpm4#config term
Enter configuration commands, one per line. End with CNTL/Z.
TErpm4 (config) #
```

Step 3 If this is not done, activate support for CEF, as follows.

```
TErpm4 (config) #ip cef
```

Step 4 If necessary, configure resource partitioning for the RPM. You can enter the **rpmrscprtn** command at any point before adding the user-connection, however, its executive level makes resource partitioning more convenient before you enter the subinterface level.

```
rpmrscprtn <partition_type> <percent_ingressBW> <percent_egressBW> <min_vpi> <max_vpi>
<min_vci> <max_vci> <max_lcns>
```

- *partition_type* identifies the network controller. Choices are PAR, MPLS, and PNNI.
- *percent_ingressBW* is the percent of available bandwidth for the ingress direction (towards the cellbus).
- *percent_egressBW* is the percent of available bandwidth for the egress direction (away from the cellbus)
- *min_vpi* is the minimum vpi within the range 0–255. (Note that once you set the minimum vpi, it is the lower limit of the range for maximum vpi.)
- *max_vpi* is the maximum vpi within the range 0–255.
- *min_vci* is the minimum vci. The vci range is 0–3840.
- *max_vci* is the maximum vci. The vci range is 0–3840.
- *max_lcns* is the maximum number of LCNs for this controller. The range is 0–4080.

For example,

```
TErpm4 (config) #rpmrscprtn par 100 100 10 10 10 3000 2000
```


**Note**

The MGX 8850 CLI has the partitioning ability to support MPLS (or “tag”) control of the MGX resources. The “tag” partition will be required once the MGX 8850 has support for an RPM acting as a Label Switch Controller (LSC). LSC support is not yet available, so the “tag” partition is not used. In the current example, MPLS connectivity is provided by way of PVCs. PVCs use resources from the Portable AutoRoute (PAR) partition if PNNI is not used, or from the PNNI partition if PNNI is used. (Technically, an SPVC and not a PVC is used with PNNI).

Step 5 Enter the **router** command to specify the routing protocol for the RPM,.

```
TERpm4(config)#router <protocol> <PID>
```

- *protocol* is any routing protocol such as OSPF, EIGRP, RIP, and IS-IS.
- *PID* is a process identification number.

The recommended protocols are OSPF or IS-IS. The PID can be any integer in the range 1–65535.

**Note**

A router can communicate with only those routers that have the same protocol unless you execute the optional **redistribute** command to specify protocol translation (see the IOS command reference for details).

Step 6 Enter the **network** command to specify a *network area*.

MPLSA network area is a number that an *entity* must have in common with another entity for communication to occur. The entity is defined by an IP address or a subnet mask (or both). The syntax is:

```
RPM_name(config)#network <IP address> <mask> <area> <area_number>
```

- **area** indicates the area number follows
- *area_number* is a number you associate with the IP address and mask. The range for *area_number* is 1–65535.

For example:

```
TERpm4(config-subif)#network 212.212.212.0 0.0.0.255 area 200
```

For each of the preceding lines, each entity can communicate with any other suitable entity if the network area number is the same. For example, the first example line assigns the network area number 200 to the current node, so this node can communicate with other nodes that have a network area number of 200.

Step 7 Enter the **interface** command to specify a subinterface on the RPM.

If the subinterface does not exist, the first-time specification of a subinterface *creates* the interface. Thereafter, when you provide the number of the subinterface to **interface**, you *enter* that interface. The syntax is:

```
RPM_name(config)#interface switch subinterface_number link
```

- **switch** indicates an interface between the RPM and the cellbus.
- *subinterface_number* indicates the shelf number (always “1” because the node has one cellbus), (and a subinterface number in the range 1–700 for RPM/B and 1–2000 for RPM-PR. Note the slash and the period in this parameter. Specify **point-to-point** subinterface.

For example:

```
TERpm4(config)#interface switch 1.200 point-to-point
```

- **switch** is an extension indicating the RPM's ATM interface.
- cellbus *interface number* is always "1" because currently only one cellbus slave exists on the RPM.
- *subinterface* is 200. The range for subinterfaces is 1–700 for RPM/B and 1–2000 for RPM-PR. You must also configure the subinterface to be point-to-point, multipoint, or MPLS. With point-to-point or MPLS, each subinterface has one PVC. With multipoint, a subinterface has one IP address but multiple connections to one or more RPMs.

Step 8 Assign an IP address to the subinterface.

Normally, in an ATM MPLS network, subinterfaces in the network are not numbered, but are unnumbered. It is less important to do this in a packet-based MPLS network (that is, one using PVCs), but it still may be done, and it does lead to the simplest configuration. If an unnumbered interface is used, the interface address is taken to be the address of a loopback interface on the router. This procedure minimizes the number of addresses and routes in the network, and minimizes the number of MPLS labels used.

To do this, enter the following command,

```
RPM(config-subif)#ip unnumbered loopback0
```

Alternatively, an IP address could be assigned by using an **ip address** command.

Step 9 Configure the PVC. Each subinterface must have a connection to the cellbus. The command is **atm PVC**:

```
atm pvc vcd vpi aal-encap [[midlow midhigh] [peak average burst] [oam seconds] [inarp minutes]
```

For example,

```
rpm(config-subif)#atm PVC 10 0 10 aal5snap 100000 50000 10000 inarp 10 oam ?  
<0-600> OAM loopback frequency(seconds)
```

- *vcd* is a user-specified connection identifier with significance on the local RPM only. It refers to the connection whether the connection is a VCC or VPC. You can specify any integer, but using the same value as the vci (or vpi when applicable) helps to avoid possible confusion.
- *vpi* or ATM network virtual path identifier of this PVC is always **0** (for a VCC).
- *vci* or ATM network virtual channel identifier is **10**.
- *aal-encap* encapsulation is aal5snap.
- *peak cell rate* is 100000.
- *average cell rate* is 50000.
- *maximum burst size* is 10000.
- *inarp* is enabled by its inclusion on the line. This parameter enables periodic reporting on the presence of each far-end interface by way of an IP address. If you enable inarp, the parameter that follows it is the time between reports. The number of minutes between inarp reports is 10 in the current example.
- *oam cell circulation* is enabled. The current example shows the device that requests information about the next parameter. For this parameter, the range for the intervals between OAM cell transmission is 0–600 seconds.

Step 10 Enable MPLS on this subinterface.

Execute the following command.

```
RPM(config-subif)#tag-switching ip
```

This command enables MPLS using the Tag Distribution Protocol for label distribution.

Step 11 Specify an IP address for each subinterface on the RPM.

```
TErpm4(config-subif)#ip addr 162.32.188.17 255.255.255.0
TErpm4(config-subif)#exit
TErpm4(config)#
```

Step 12 Add a connection between the local and remote RPMs through **addcon**. The syntax is:

```
addcon <connection_type> <switch> <switch.subinterface> <local_vci> <rslot> <switch>
<remote_vpi> <remote_vci> [master local]
```

- *connection_type* is either **vcc** or **vpc**. For this connection, type **vcc**.
- **switch** indicates this device is an RPM and that the virtual switch identification follows.
- *switch.subinterface* identifies the virtual switch interface. The *slot* identifies the RPM slot. The *switch* is always 1. The *subinterface* is a number in the range 1–700 for RPM/B and 1–2000 for RPM-PR.
- **rslot** indicates that the remote slot number follows.
- remote *switch* number is always 1.
- *remote_vpi* of 0 for a VCC where the remote end is another RPM.
- *remote_vci* is the remote VCI. It may take any value, and is not necessarily the same as the *local_vci*.
- **master local** identifies the local end of the connection as the master. If the other end is the master, do not enter either **master** or **local** for this parameter.

For example,

```
TErpm4(config-subif)#addcon vcc switch 1/1.200 1012 rslot 10 1 0 1012 master local
```

Step 13 Add the connection at the other endpoint. Where appropriate, use identical parameters and unique parameters (such as **master** connection).

Step 14 Verify that the connection exists by using the **dspscons**, **dspchans**, **tstcon**, **tstconseg**, **dspslotcon**, or **dspportcon** commands on the PXM and **show atm pvc**, **show atm vc**, **show switch conn**, and the **ping** commands on the RPM.

Step 15 Add more connections as needed.



Note Remember that, for each packet-based MPLS connection over a PVC created with **addcon**, you must first specify a subinterface and add an ATM PVC between the RPM and the cellbus.

The IOS-version of the **addcon** command adds a local user-connection between the RPM and another RPM in the MGX 8850 switch. For a three-segment connection in an ATM or Frame Relay network, you must also add a segment between the edges of the cloud. Through the CLI of the BPX 8600 series switch, you enter the switch-software version of **addcon**.

Three-Segment Connections

A three-segment connection consists of a local connection between an RPM and PXM at each of two MGX 8850 switches and a connection between two endpoints in an ATM or Frame Relay cloud.

Configuring a Virtual Path Tunnel

This section describes how to add PVP connections for ATM MPLS connectivity in a network core, using the command line interface of IOS. With ATM MPLS using PVPs, a PVP is created from the RPM to the nearest ATM-LSR, for example, a BPX 8650. The ATM-LSR then connects to the rest of the ATM MPLS network.

The RPM terminates the PVP on a subinterface configured for ATM MPLS. The ATM-LSR must terminate the PVP on a virtual trunk interface. On the BPX 8650, virtual trunk interfaces must be on UNI or NNI ports, and must not be on feeder trunks. This setup creates additional requirements on the MGX 8850. An MGX 8850 would normally be connected to the adjacent node by a link configured as a trunk or feeder trunk, but one additional link is required to use MPLS ATM MPLS over PVPs, for a total of two links.

**Note**

If it is essential to use a single link, for example, if an MGX 8850 system with a PXM-1 card is connected to a BPX 8650 with a single OC12/STM-4 feeder trunk, then it is still possible to use the BPX 8650 as an ATM-LSR as a work-around. Contact TAC if you need to do this.

Therefore, for each MPLS connection from an RPM to an ATM-LSR over a PVP, you must create:

- A UNI or NNI link to the ATM-LSR, if it not already present. (This is typically either a PXM-1 link, or an AXSM link in the case of PXM-45 system).
- An ATM PVP between the link and the RPM.
- A subinterface connected to the PVP, with MPLS enabled.

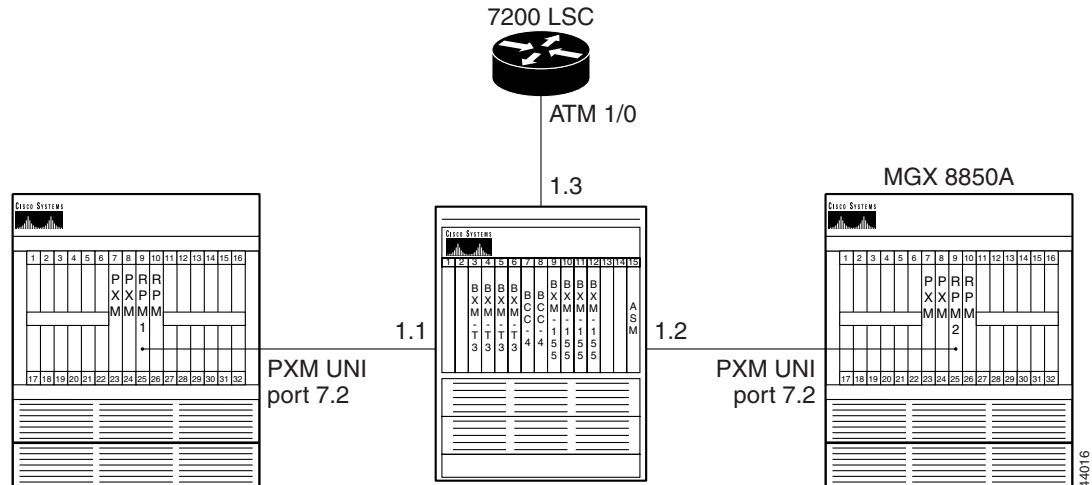
From either a control terminal connected to the RPM or through the MGX 8850 switch, go to the IOS command line interface, then perform the following tasks (skip any completed task):

1. Enable Cisco Express Forwarding (CEF).
2. Specify a routing protocol.
3. Specify the area in which the RPM can communicate.
4. Configure the subinterface.
5. Create an ATM PVP.
6. Ping the far end router to verify connectivity.
7. Add VPN configuration (this is described later).

The following items must be the same on the RPMs at both ends:

- Routing protocol (preferably OSPF or IS-IS).
- Routing area (specified by a network area number).

Figure 7-8 Configuring a Virtual Path Tunnel



This example shows there are two MGX 8850 nodes and a BPX 8650. Each MGX 8850 has a PXM-1 card supporting 4xSTM-1 links. The UNI link between each MGX 8850 and BPX 8650 is link number 2 on the PXM-1 card, which is in slot 7. Each link connects to a 4-port BXM card in the BPX 8650 switch, and the links are 1.1 and 1.2, for example, card 1, ports 1 and 2. The BPX 8650 also has a 7200 Label Switch Controller connected to port 1.3. The MGX 8850 may also have a feeder trunk to the BPX 8650, which would be in slot 1 of the PXM-1 card. (The feeder trunk is not used by MPLS in this example, and is not shown.)

The following example describes the RPM configuration in detail, and also summarizes the required PXM-1, BPX 8650 and LSC configuration.

PXM-1 Configuration Summary

If the UNI link to the BPX has not already been created, create it now. A typical configuration is as follows. (For more information on configuring PXM-1, refer to the *Cisco MGX 8850 Multiservice Switch Installation and Configuration* guide, Chapter 6, “Card and Service Configuration.”) The configurations of both PXM-1 cards in the example are identical.

Step 1 Enter the **addln** command to enable the link.

```
addln 7.2
```

Step 2 Enter the **addport** command to configure a logical port on the link.

```
addport 2 2 100 0 4095
```

The logical port number is “2”, which can be different from the physical line number, 2, if required. This command gives full control of the port to Portable AutoRoute, so that a PVP may be set up on it. The **addport** command also has an “MPLS” option, but that is not used with PVPs. It will be used once the MGX 8850 has LSC support.

Step 3 Enter the **cnfatmln** command to declare the port as a UNI.

```
cnfatmln 2 2
```

The first “2” is the physical line number, the second “2” indicates “UNI.”

RPM1 Configuration

To create a VP tunnel, enter the following steps.

- Step 1** Enter configuration mode, as follows.

```
TErpm4>
TErpm4>ena
Password:
```

Enter the password.

The prompt subsequently includes a pound sign (#). At this command level, you can show aspects of the RPM per the IOS commands and list configuration commands, while not actually entering them.

```
TErpm4#
```

- Step 2** Enter configuration mode and include a specification for the configuration source. The source is terminal, memory, or network. The syntax and resulting display are:

```
TErpm4#config term
Enter configuration commands, one per line. End with CNTL/Z.
TErpm4 (config)#
```

- Step 3** If you have not already indicated the switch-level support for Cisco Express Forwarding, do so now by entering the **ip cef** command.

```
TErpm4 (config)#ip cef
```

- Step 4** Create the VP-tunnel by entering the **tag-switching** command. Note that the subinterfaces you subsequently create for this tunnel must be of the MPLS switching type (rather than point-to-point or multipoint). The syntax is

```
RPM_name(config-subif)#tag-switching atm vp-tunnel <VP-tunnel_number> [vci-range]
```

- *VP-tunnel_number* is the vpi for the tunnel. The tunnel vpi must be the same at both ends of the tunnel.
- *vci-range* is from 33 to 1000.

For example,

```
TErpm4 (config-subif)#tag-switch ATM VP-tunnel 50
```

- Step 5** If you have not already specified the router protocol for the RPM, do so now by entering the **router** command.

```
RPM_name (config)#router <protocol> <PID>
```

- *protocol* is any router protocol such as OSPF, EIGRP, RIP, or ISIS.
- *PID* is a process identification number. Typically, OSPF is the protocol. The PID can be any integer you want.

- Step 6** Enter the **network** command to specify a network area, which is a number that a communications *entity* must have in common with another entity for communication to occur. The entity is defined by an IP address or a subnet mask (or both).

```
RPM_name(config)#network <IP address> <mask> <area> <area_number>
```

- **area** indicates that an area number follows.
- *area_number* is a number you associate with the IP address and mask. The range for *area_number* is 1–65535.

For example,

```
TErpm4(config)#network 10.1.0.0 0.0.255.255 area 0
```

In the preceding example, the command assigns the network area number 200 to the current node, so this node can communicate with other nodes that have a network area number of 200.

- Step 7** Enter the **interface** command to specify a subinterface on the RPM. If the subinterface does not already exist, the first-time specification of a subinterface *creates* the interface. Thereafter, when you provide the number of the subinterface to **interface**, you enter that interface. The syntax is

```
RPM_name(config)#interface switch shelf.subinterface_number link
```

- **switch** indicates an interface between the RPM and the cellbus.
- *shelf.subinterface_number* indicates the RPM shelf number (always “1” because the RPM connects to only one cellbus)
- *subinterface number* is in the range 1–700 for the RPM/B and 1–2000 for the RPM-PR. Note the slash and the period in this parameter.
- *link* is **point-to-point**, **MPLS**, or **multipoint**. You must specify point-to-point.

For example,

```
TErpm4(config)# interface switch/1.200 MPLS
```

- **switch** is an extension indicating an RPM.
- cellbus *interface number* is always 1 because only one cellbus exists on the switch
- *subinterface* is 200, which is in the range of 1–700 for the RPM/B and 1–2000 for the RPM-PR. Note the slash and the period in this parameter.
- *link* is **MPLS**. To create a VP-tunnel, the controller for the subinterface must be MPLS. Note also that, with MPLS or point-to-point, each subinterface has one PVC. With multipoint, a subinterface has one IP address but multiple connections that can terminate on one or more RPMs.

- Step 8** Enter the **atm PVC** command to configure each subinterface connection to the cellbus.

```
atm PVC vcd vpi aal-encap [[midlow midhigh] [peak average burst] [oam seconds] [inarp minutes]
```

For example,

```
rpm(config-subif)#atm PVC 50 50 0 aal5snap vbr-nrt 10000 10000 64000
<0-600> OAM loopback frequency(seconds)
<cr>
```

- *vcd* is **10**. The VCD is a user-specified connection identifier with significance only on the local RPM. It refers to the connection whether the connection is a VCC or VPC. You can specify any integer, but using the same value as the VPI (or VPI when applicable) helps to avoid possible confusion.
- *vci* of **0**, with a *vpi* of **50**, configures a PVP. Another value may be chosen if desired.
- *vbr-nrt 10000 10000 64000* configures the VP for **CBR** shaping with a peak rate of **10** Mb/s.
- *aal-encap* encapsulation is **aal5snap**.

- *peak cell rate* is **100000**.
- *average cell rate* is **5000**.
- *maximum burst size* is **10000**.
- **inarp** enables inarp, which enables periodic reporting on the presence of far-end interfaces by way of IP addresses. If you enable inarp, the next parameter is the time between reports.
- *minutes between inarp reports* is **10**.
- **oam** cell circulation is enabled. This example shows the prompt that requests information about the next parameter. For this parameter, the range for the intervals between OAM cell transmission is 0–600 seconds.

Step 9 Enter the **ip addr** command to specify an IP address for each subinterface on the RPM.

```
TErpm4(config-subif)#ip addr 162.32.188.17 255.255.255.0
```

Step 10 Exit the current level of configuration.

```
TErpm4(config-subif)#exit
```

Step 11 If necessary, **cc** to the card to configure a resource partition and enter the **rpmrscprtn** command.

Resource partitions are not a subinterface-level entity. Therefore, exit the subinterface if necessary (as in Step 10).

```
rpmrscprtn <partition_type> <percent_ingressBW> <percent_egressBW> <min_vpi> <max_vpi>  
<min_vci> <max_vci> <max_lcns>
```

For example

```
TErpm4(config)#rpmrscprtn par 100 100 10 10 10 3000 2000
```

- *partition_type* identifies the network controller. Choices are PAR, MPLS, and PNNI.
- *percent_ingressBW* is the percent of available bandwidth for the ingress direction (towards the cellbus).
- *percent_egressBW* is the percent of available bandwidth for the egress direction (away from the cellbus).
- *min_vpi* is the minimum VPI within the range 0–255. (Note that once you set the minimum vpi, it is the lower limit of the range for maximum VPI.)
- *max_vpi* is the maximum VPI, within the range 0–255.
- *min_vci* is the minimum VCI. The VCI range is 0–3840.
- *max_vci* is the maximum VCI. The VCI range is 0–3840.
- *max_lcns* is the maximum number of LCNs for this controller. The range is 0–4080.

Step 12 Enter the **addcon** command to add a VP tunnel between the local and remote RPMs.

```
addcon <connection_type> switch <slot/switch.subinterface> <local_vpi> rslot <slot>  
<switch> <remote_vpi> [master local]
```

- *connection_type* is either **vcc** or **vpc**. For this connection, type **vpc**.
- **switch** indicates this device is an RPM and that the virtual switch identification follows.
- *slot/switch.subinterface* identifies the virtual switch interface. The *switch* is always **1**.
- **rslot** indicates that the remote slot number follows.
- *slot* is the slot number of the remote RPM.
- Remote *switch* number is always **1**.

- *remote_vpi* is the remote VPI and must be the same as the *local_vpi*.



Note The keywords **master local** let you identify the local end of the connection as the master. If the other end is the master, enter nothing (do not enter either **master** or **local**).

- Step 13** Add the connection at the other endpoint. Where appropriate, use identical parameters such as the VPI or unique parameters such as the connection mastership specification.
- Step 14** Verify that the connection exists by entering the **dspcons** command.
- Step 15** Continue to add more VP-tunnels as needed.

The IOS version of the **addconn** command adds a local user-connection between the RPM and another endpoint in the MGX 8850 switch. For a three-segment connection in an ATM or Frame Relay network, you must also add a middle segment between the edges of the cloud. Through the CLI of the BPX 8600 series switch, you would enter the switch-software version of **addconn**.

The following example shows a typical configuration

```
---
hostname RPM1
!
ip cef
!
interface loopback0
ip address 10.1.1.1 255.255.255.255
!
interface Switch1
no ip address
!
interface Switch1.1 tag-switching
ip unnumbered Loopback0
atm pvc 50 50 0 aal5snap vbr-nrt 10000 10000 64000
tag-switching atm vp-tunnel 50
tag-switching ip
!
router ospf 1
network 10.1.0.0 0.0.255.255 area 0
!
rpmrscprtn par 100 100 1 100 10 4000 3800
addconn vpc switch 1/1.1 50 rslot 0 1 50 master local
!
---
```

RPM2 Configuration

The configuration of the second RPM should be the same as that of the first RPM, except that the loopback address should be 10.1.1.3 not 10.1.1.1, as shown in the following example.

```
---
hostname RPM2
!
ip cef
!
interface loopback0
ip address 10.1.1.3 255.255.255.255
!
!
```

```

interface Switch1
no ip address
!
interface Switch1.1 tag-switching
ip unnumbered Loopback0
atm pvc 50 50 0 aal5snap vbr-nrt 10000 10000 64000
tag-switching atm vp-tunnel 50
tag-switching ip
!
rpmrscprtn par 100 100 1 100 10 4000 3800
addconn vpc switch 1/1.1 50 rslot 0 1 50 master local
!
router ospf 1
network 10.1.0.0 0.0.255.255 area 0
---
```

BCC Configuration

The following paragraphs show a summary of the commands required on the BCC CLI and the BPX 8650 node. For more details of the BPX8650 and LSC configuration, refer to the *Cisco MPLS Controller Software Configuration Guide*.

The PVPs terminate on VSI Virtual Trunk interfaces on ports 1.1 and 1.2. In this example, only one virtual trunk is needed on each interface, and it is numbered 1. If required, there could be more than one virtual trunk per interface, with any numbers in the range 1 to 31.

```

uptrk 1.1.1
uptrk 1.2.1
```

The LSC is connected to a trunk, and not a virtual trunk, as shown in the following example.

```

uptrk 1.3
```

The VSI virtual trunk interfaces must now have virtual trunk parameters configured with the **cnftrk** command. Most of the **cnftrk** parameters are relevant only for virtual trunks carrying PVCs, not MPLS LVCs, and can therefore be left at their default values. There are the following exceptions.

- Trunk bandwidth must be set correctly.
 - 26,000 cells per second or about 10 Mb per second in this example
- The appropriate VPI must be chosen
 - 50 in this example

```

cnftrk 1.1.1 26000 N 1000 7F V,...,RT-VBR N TERRESTRIAL 10 0 0 N N N CBR 50 0
cnftrk 1.2.1 26000 N 1000 7F V,...,RT-VBR N TERRESTRIAL 10 0 0 N N N CBR 50 0
```

Enter the **cnfrsrc** command to assign resources for MPLS use on the subinterfaces and interfaces. On all interfaces, MPLS is guaranteed to be able to use 1000 LVCs, and permitted to use up to 4000 if resources are spare. The PVP endpoints use a VPI of 50, but the link to the LSC has the range of VPIs 2-10 assigned to MPLS. The PVPs are configured for a bandwidth of 26000 cells per second (about 10mb/s), but 260,000 cells per second are guaranteed for MPLS on the link to the LSC.

```

cnfrsrc 1.1.1 256 0 1 e 1000 4000 50 50 26000 26000
cnfrsrc 1.2.1 256 0 1 e 1000 4000 50 50 26000 26000
cnfrsrc 1.3 256 0 1 e 1000 4000 2 10 260000 260000
```

Enter the **addshelf** command to add the LSC.

```

addshelf 1.3 vsi 1 1
```

LSC Configuration

A typical LSC configuration is shown in the following example. For more details on configuring an LSC, refer to the *Cisco MPLS Controller Software Configuration Guide*.

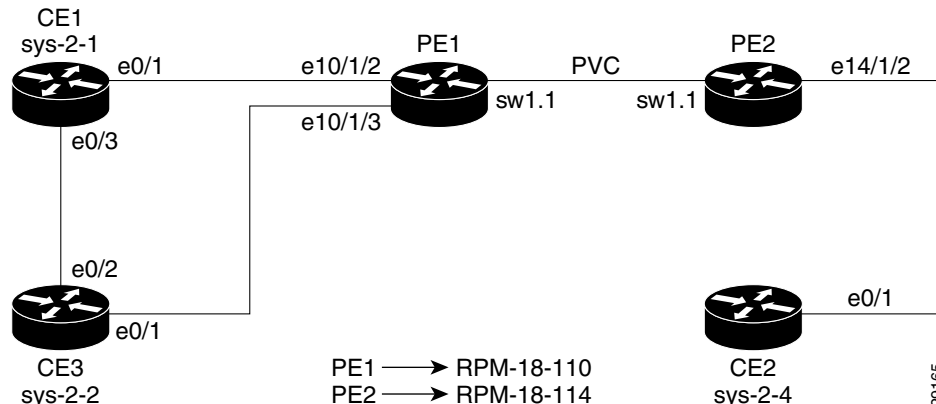
In the LSC, it is not necessary to specify that the interfaces 1.1.1 and 1.1.2 are “VP tunnels.” The LSC automatically detects this from information passed to the LSC by the BPX. The LSC also automatically detects the valid VPI, 50, because this was configured with the **cnfrsrc** commands on the BPX.

```
ip cef switch
!
interface lo0
ip address 10.1.1.2 255.255.255.255
!
interface atm1/0
no ip address
tag-control-protocol vsi
!
interface XTagATM111
extended-port atm1/0 bpx 1.1.1
ip unnumbered loopback0
tag-switching ip
!
interface XTagATM121
extended-port atm1/0 bpx 1.2.1
ip unnumbered loopback0
tag-switching ip
!
router ospf 1
network 10.1.0.0 0.0.255.255 area 0
```

RPM Configuration Examples for MPLS-based Virtual Private Networks

The examples in this section use the same configuration, with two PEs and a single VPN containing 3 CEs. (Many more VPNs could be added if needed.) For simplicity, both RPMs are in the same MGX 8850 shelf, and are linked by a PVC. Any other linking of RPMs can also be used.

In the following example, CE1 and CE2 have a fairly sophisticated configuration, with a “back door” link between each other, which is a connection that is not part of the VPN. CE3 has a simpler configuration that easier to implement and therefore, more commonly used.

Figure 7-9 VPN Configuration

The CEs are connected to the MGX 8850 by Frame Relay PVCs. In this example, the three CEs are connected to a single FRSM card in slot 18 of the MGX 8850. Each CE is connected to a separate T1 port in a FRSM-8T1 card.

FRSM Configuration

The following example shows a typical FRSM configuration. For more details, see the “Frame over ATM Example” section on page 6-18 of this manual. For a full description of Service Module configuration options, refer to the *Cisco MGX 8850 Multiservice Switch Installation and Configuration, Release 1.1.31*, Chapter 6, “Card and Service Configuration.”

Step 1 Enter the **addln** command to configure the first line.

```
Example.1.18.FRSM.a> addln 1
```

Step 2 Enter the **addport** command to configure a Frame Relay logical port on the line.

```
Example.1.18.FRSM.a> addport 1 1 2 1 24 1
```

This command configures a Frame Relay data link using the whole T1 rate, for example, all 24x64kb/s time slots on the link.

- **1**—define this to be logical port 1 (could use another number in range 1-192)
- **1**—physical line is 1
- **2**—DSO speed is chosen to be 64k per T1 time slot
- **1**—beginning time slot is chosen to be 1
- **24**—last time slot is 24
- **1**—this is a Frame Relay port (not FUNI or frame forwarding)

Step 3 Enter the **xcnfport** command to configure LMI on the line.

```
Example.1.18.FRSM.a> xcnfport -pt 11 -sig 3
```

- **pt 1**—logical port is 1, as defined above
- **sig 3**—specifies StrataLMI

- Step 4** Enter the **addcon** command to configure a service-interworking or network-interworking channel from the FRSM port to PE1.

```
Example.1.18.FRSM.a> addcon 1 123 1536000 3 2 1 2 Example.9.1.0.101
```

- **1**—port number as defined above
- **123**—choose DLCI 123 on the Frame Relay data link
- **1536000**—committed information rate is 1536000bits/s, for example, full T1 rate
- **3**—channel type is service interworking
- **2**—connection use VBR-rt queueing into the cellbus
- **1**—connection uses Portable AutoRoute PVC routing, not PNNI
- **2**—this end of the connection is the slave end, (which means that the RPM is the master)

```
Example.9.1.0.101
```

The other end of the PVC connection is in the same node (called “Example”), slot 9 (which is an RPM, PE1), port 1 (must be 1 for an RPM, it is the Cellbus interface), VPI 0 (must be 0 for a PVC on the RPM), and the VCI is chosen to be 101.

The second customer link also has one PVC, DLCI 123. This is cross-connected to a different VCI on PE1, namely 102.

```
addln 2
addport 2 2 2 1 24 1
xcnfport -pt 2 -sig 3
addcon 2 123 1536000 3 2 1 2 Example.9.1.0.102
```

The third customer link again has one PVC, DLCI 123. This customer link connects to PE2 in slot 11, using VCI 101.

```
addln 3
addport 3 3 2 1 24 1
xcnfport -pt 3 -sig 3
addcon 3 123 1536000 3 2 1 2 Example.11.1.0.101
```

Two PE–Three CE Configuration BGP-connected CE routers

The following example shows the most sophisticated and flexible way of setting up routing between the CE and PE routers, which is to use BGP between them. (This is distinct from the normal use of BGP in a VPN backbone.)

CE1 Configuration

This is an example of the CE1 configuration.

```
hostname CE1
!
interface Ethernet0
ip address 172.16.1.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0
no ip address
encapsulation frame-relay
```

```

service module t1 clock source line
!
interface Serial0.1 point-to-point
frame-relay interface-dlci 123
ip address 10.101.1.2 255.255.255.252
no ip directed-broadcast
!
router ospf 100
redistribute bgp 101
network 172.16.0.0 0.0.255.255 area 100
!
router bgp 101
redistribute ospf 100
no synchronization
network 172.16.0.0
neighbor 10.101.1.1 remote-as 100
!

```

CE2 Configuration

This is an example of the CE2 configuration.

```

hostname CE2
!
interface Ethernet0
ip address 172.16.1.2 255.255.255.0
no ip directed-broadcast
!
interface Serial0
no ip address
encapsulation frame-relay
service module t1 clock source line
!
interface Serial0.1 point-to-point
frame-relay interface-dlci 123

ip address 10.101.2.2 255.255.255.252
no ip directed-broadcast
!
router ospf 100
redistribute bgp 101
network 172.16.0.0 0.0.255.255 area 100
!
router bgp 101
redistribute ospf 100
no synchronization
network 172.16.0.0
neighbor 10.101.2.1 remote-as 100
!

```

CE3 Configuration

This is an example of the CE3 configuration.

```

hostname CE3
!
interface Ethernet0
ip address 172.16.3.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0
no ip address

```

```
encapsulation frame-relay
service module t1 clock source line
!
interface Serial0.1 point-to-point
frame-relay interface-dlci 123

ip address 10.101.3.2 255.255.255.252
no ip directed-broadcast
!
router ospf 100
redistribute bgp 101
network 172.16.0.0 0.0.255.255 area 100
!
router bgp 101
no synchronization
redistribute ospf 100
network 172.16.0.0
neighbor 10.101.3.1 remote-as 100
!
```

PE1 Configuration

This is an example of the PE1 configuration.

```
hostname RPM1
!
ip vrf vpn1! Define a VRF for VPN 1
rd 100:1
route-target export 100:1
route-target import 100:1
!
ip cef
!
interface loopback0
ip address 10.1.1.1 255.255.255.255
!
interface Switch1
no ip address
!
interface Switch1.1 tag-switching
ip unnumbered Loopback0
atm pvc 50 50 0 aal5snap vbr-nrt 10000 10000 64000
tag-switching atm vp-tunnel 50
tag-switching ip
!
router ospf 1
network 10.1.0.0 0.0.255.255 area 0
interface Switch1.101
ip address 10.101.1.1 255.255.255.252
pvc 0/101
encap aal5snap
vbr-nrt 1500 1500 64000
ip vrf forwarding vpn1
!
interface Switch1.102
ip address 10.101.2.1 255.255.255.252
pvc 0/102
encap aal5snap
vbr-nrt 1500 1500 64000
ip vrf forwarding vpn1
!
router bgp 100
no synchronization
```

```

no bgp default ipv4-unicast
neighbor 212.212.212.2 remote-as 100 !define BGP connection to PE2 for IPv4
neighbor 212.212.212.2 update-source Loopback0
!
address-family vpnv4!define BGP connection to PE2 for VPN routes
neighbor 212.212.212.2 activate
neighbor 212.212.212.2 send-community extended
exit-address-family
!
address-family ipv4 vrf vpn1! Define BGP PE-CE sessions for vpn1
neighbor 10.101.1.2 remote-as 101
neighbor 10.101.1.2 activate
neighbor 10.101.2.2 remote-as 101
neighbor 10.101.2.2 activate
no auto-summary
no synchronization
exit-address-family
!
rpmrscprtn par 100 100 1 100 10 4000 3800
addconn vpc switch 1/1.1 50 rslot 0 1 50 master local
addcon vcc Switch1.101 101 rslot 18 1 0 123 master local
addcon vcc Switch1.102 102 rslot 18 2 0 123 master local
!

```

PE2 Configuration

This is an example of the PE2 configuration.

```

hostname RPM1
!
ip vrf vpn1! Define a VRF for VPN 1
rd 100:1
route-target export 100:1
route-target import 100:1
!
ip cef
!
interface loopback0
ip address 10.1.1.1 255.255.255.255
!
interface Switch1
no ip address
!
interface Switch1.1 tag-switching
ip unnumbered Loopback0
atm pvc 50 50 0 aal5snap vbr-nrt 10000 10000 64000
tag-switching atm vp-tunnel 50
tag-switching ip
!
router ospf 1
network 10.1.0.0 0.0.255.255 area 0

interface Switch1.101
ip address 10.101.3.1 255.255.255.252
pvc 0/101
encap aal5snap
vbr-nrt 1500 1500 64000
ip vrf forwarding vpn1
!
router bgp 100
no synchronization
no bgp default ipv4-unicast
neighbor 212.212.212.1 remote-as 100!define BGP connection to PE1 for IPv4

```



```

neighbor 212.212.212.1 update-source Loopback0
!
address-family vpnv4!define BGP connection to PE1 for VPN routes
neighbor 212.212.212.1 activate
neighbor 212.212.212.1 send-community extended
exit-address-family
!
address-family ipv4 vrf vpn1! Define BGP PE-CE3 session for vpn1
neighbor 10.101.3.2 remote-as 101
neighbor 10.101.3.2 activate
no auto-summary
no synchronization
exit-address-family
!
rpmrscprtn par 100 100 1 100 10 4000 3800
addconn vpc switch 1/1.1 50 rslot 0 1 50 master local
addcon vcc Switch1.101 101 rslot 18 1 0 123 master local
!

```

Two PE–Three CE Configuration, RIP-connected CE Routers

This is similar to the previous examples except that RIP is used between CE and PE routers. The use of RIP is not recommended when there are “back-door” routes between customer sites, for example, links between CEs, except through the VPN network.

CE1 Configuration

This is an example of the CE1 configuration.

```

hostname CE1
!
interface Ethernet0
ip address 172.16.1.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0
no ip address
encapsulation frame-relay
service module t1 clock source line
!
interface Serial0.1 point-to-point
frame-relay interface-dlci 123

ip address 10.101.1.2 255.255.255.252
no ip directed-broadcast
!
router rip
version 2
network 10.0.0.0
network 172.16.0.0
no auto-summary
!

```

CE2 Configuration

This is an example of the CE2 configuration.

```

hostname CE2
!

```

```

interface Ethernet0
ip address 172.16.2.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0
no ip address
encapsulation frame-relay
service module t1 clock source line
!
interface Serial0.1 point-to-point
frame-relay interface-dlci 123

ip address 10.101.2.2 255.255.255.252
no ip directed-broadcast
!
router rip
version 2
network 10.0.0.0
network 172.16.0.0
no auto-summary
!

```

CE3 Configuration

This is an example of the CE3 configuration.

```

hostname CE3
!
interface Ethernet0
ip address 172.16.3.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0
no ip address
encapsulation frame-relay
service module t1 clock source line
!
interface Serial0.1 point-to-point
frame-relay interface-dlci 123

ip address 10.101.3.2 255.255.255.252
no ip directed-broadcast
!
router rip
version 2
network 10.0.0.0
network 172.16.0.0
no auto-summary
!

```

PE1 Configuration

This is an example of the PE1 configuration.

```

hostname RPM1
!
ip vrf vpn1! Define a VRF for VPN 1
rd 100:1
route-target export 100:1
route-target import 100:1
!
ip cef

```

```

!
interface loopback0
ip address 10.1.1.1 255.255.255.255
!
interface Switch1
no ip address
!
interface Switch1.1 tag-switching
ip unnumbered Loopback0
atm pvc 50 50 0 aal5snap vbr-nrt 10000 10000 64000
tag-switching atm vp-tunnel 50
tag-switching ip
!
router ospf 1
network 10.1.0.0 0.0.255.255 area 0
interface Switch1.101
ip address 10.101.1.1 255.255.255.252
pvc 0/101
encap aal5snap
vbr-nrt 1500 1500 64000
ip vrf forwarding vpn1
!
interface Switch1.102
ip address 10.101.2.1 255.255.255.252
pvc 0/102
encap aal5snap
vbr-nrt 1500 1500 64000
ip vrf forwarding vpn1
!
router bgp 100
no synchronization
no bgp default ipv4-unicast!define BGP connection to PE2 for IPv4
neighbor 212.212.212.2 remote-as 100
neighbor 212.212.212.2 update-source Loopback0
!
address-family vpnv4!define BGP connection to PE2 for VPN routes
neighbor 212.212.212.2 activate
neighbor 212.212.212.2 send-community extended
exit-address-family
!
address-family ipv4 vrf vpn1!logical connection to vrf1
redistribute static
no auto-summary
exit-address-family
!
ip route vrf vpn1 172.16.1.0 255.255.255.0 Switch1.101
ip route vrf vpn1 172.16.2.0 255.255.255.0 Switch1.102
!
rpmrscprtn par 100 100 1 100 10 4000 3800
addconn vpc switch 1/1.1 50 rslot 0 1 50 master local
addcon vcc Switch1.102 102 rslot 18 2 0 123 master local
addcon vcc Switch1.101 101 rslot 18 1 0 123 master local
!

```

PE2 Configuration

This is an example of the PE2 configuration.

```

hostname RPM1
!
ip vrf vpn1! Define a VRF for VPN 1
rd 100:1
route-target export 100:1

```

```

route-target import 100:1
!
ip cef
!
interface loopback0
ip address 10.1.1.1 255.255.255.255
!
interface Switch1
no ip address
!
interface Switch1.1 tag-switching
ip unnumbered Loopback0
atm pvc 50 50 0 aal5snap vbr-nrt 10000 10000 64000
tag-switching atm vp-tunnel 50
tag-switching ip
!
router ospf 1
network 10.1.0.0 0.0.255.255 area 0
interface Switch1.101
ip address 10.101.3.1 255.255.255.252
pvc 0/101
encap aal5snap
vbr-nrt 1500 1500 64000
!
router bgp 100
no synchronization
no bgp default ipv4-unicast
neighbor 212.212.212.1 remote-as 100!define BGP connection to PE1 for IPv4
neighbor 212.212.212.1 update-source Loopback0
!
address-family vpnv4!define BGP connection to PE1 for VPN routes
neighbor 212.212.212.1 activate
neighbor 212.212.212.1 send-community extended
exit-address-family
!
address-family ipv4 vrf vpn1!logical connection to vrf1
redistribute static
no auto-summary
exit-address-family
!
ip route vrf vpn1 172.16.3.0 255.255.255.0 Switch1.101
!
rpmrscprtn par 100 100 1 100 10 4000 3800
addconn vpc switch 1/1.1 50 rslot 0 1 50 master local
addcon vcc Switch1.101 101 rslot 18 1 0 123 master local
!

```

Two PE–Three CE Configuration, Static routing to CE Routers

This is similar to the RIP example except that Static routing is used between CE and PE routers. This results in the simplest but least flexible configuration.

CE1 Configuration

This is an example of the CE1 configuration.

```

hostname CE1
!
interface Ethernet0
ip address 172.16.1.1 255.255.255.0

```

```
no ip directed-broadcast
!
interface Serial0
no ip address
encapsulation frame-relay
service module t1 clock source line
!
interface Serial0.1 point-to-point
frame-relay interface-dlci 123

ip address 10.101.1.2 255.255.255.252
no ip directed-broadcast
!
ip route 172.16.1.0 255.255.255.0 Ethernet0
ip route 0.0.0.0 0.0.0.0 Serial0!default route:serial link
!
```

CE2 Configuration

This is an example of the CE2 configuration.

```
hostname CE2
!
interface Ethernet0
ip address 172.16.2.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0
no ip address
encapsulation frame-relay
service module t1 clock source line
!
interface Serial0.1 point-to-point
frame-relay interface-dlci 123

ip address 10.101.1.2 255.255.255.252
no ip directed-broadcast
!
ip route 172.16.2.0 255.255.255.0 Ethernet0
ip route 0.0.0.0 0.0.0.0 Serial0!default route:serial link
!
```

CE3 Configuration

This is an example of the CE3 configuration.

```
hostname CE3
!
interface Ethernet0
ip address 172.16.3.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0
no ip address
encapsulation frame-relay
service module t1 clock source line
!
interface Serial0.1 point-to-point
frame-relay interface-dlci 123

ip address 10.101.1.2 255.255.255.252
no ip directed-broadcast
```

```

!
ip route 172.16.3.0 255.255.255.0 Ethernet0
ip route 0.0.0.0 0.0.0.0 Serial0!default route:serial link
!

```

PE1 Configuration

This is an example of the PE1 configuration.

```

rpm-18-110#sho run
Building configuration...

Current configuration:
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname rpm-18-110
!
boot system tftp mpls/12.0/rpm-js-mz.120-5.T.bin 3.3.0.1
no logging console
!
!
!
!
!
clock timezone EST -5
clock summer-time EDT recurring
ip subnet-zero
no ip domain-lookup
ip host ios-lab-fw 3.3.0.1
!
!
ip vrf vpn1
  rd 100:1
  route-target export 100:1
  route-target import 100:1
ip cef
cns event-service server
!
process-max-time 200
!
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
 no ip directed-broadcast
!
interface Loopback1
 no ip address
 no ip directed-broadcast
!
interface Ethernet1/1
 ip address 3.3.18.110 255.255.0.0
 no ip directed-broadcast
 no ip mroute-cache
 no keepalive
!
interface Ethernet1/2
 ip vrf forwarding vpn1
 ip address 50.0.0.2 255.0.0.0

```

```
no ip directed-broadcast
no ip mroute-cache
no keepalive
tag-switching ip
!
interface Ethernet1/3
bandwidth 100
ip vrf forwarding vpn1
ip address 51.0.0.1 255.0.0.0
no ip directed-broadcast
no ip mroute-cache
shutdown
tag-switching ip
no fair-queue
!
interface Ethernet1/4
no ip address
no ip directed-broadcast
no ip mroute-cache
no keepalive
!
interface FastEthernet2/1
no ip address
no ip directed-broadcast
no ip mroute-cache
!
interface Switch1
no ip address
no ip directed-broadcast
no atm ilmi-keepalive
!
interface Switch1.1 tag-switching
ip unnumbered Loopback0
no ip directed-broadcast
atm pvc 50 50 0 aal5snap
tag-switching atm vp-tunnel 50
tag-switching ip
!
interface Switch1.2 tag-switching
ip unnumbered Loopback0
no ip directed-broadcast
atm pvc 30 30 0 aal5snap
tag-switching atm vp-tunnel 30
tag-switching ip
!
interface Switch1.3 tag-switching
ip unnumbered Loopback0
no ip directed-broadcast
atm pvc 60 60 0 aal5snap
tag-switching atm vp-tunnel 60
tag-switching ip
!
router ospf 100
passive-interface Ethernet1/2
passive-interface Ethernet1/3
network 11.0.0.0 0.255.255.255 area 100
network 50.0.0.0 0.255.255.255 area 100
network 51.0.0.0 0.255.255.255 area 100
!
router bgp 100
no synchronization
no bgp default ipv4-unicast
neighbor 10.10.10.10 remote-as 100
neighbor 10.10.10.10 update-source Loopback0
```

```

!
address-family ipv4 vrf vpn1
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family
!
address-family vpnv4
neighbor 10.10.10.10 activate
neighbor 10.10.10.10 send-community extended
exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
ip route vrf vpn1 12.0.0.0 255.0.0.0 Ethernet1/2 50.0.0.1
ip route vrf vpn1 13.0.0.0 255.0.0.0 Ethernet1/3 51.0.0.2
ip route vrf vpn1 50.0.0.0 255.0.0.0 Ethernet1/2 50.0.0.1
ip route vrf vpn1 51.0.0.0 255.0.0.0 Ethernet1/3 51.0.0.2
ip route vrf vpn1 52.0.0.0 255.0.0.0 Ethernet1/2 50.0.0.1
ip route vrf vpn1 52.0.0.0 255.0.0.0 Ethernet1/3 51.0.0.2
no ip http server
!
!
line con 0
exec-timeout 0 0
transport input none
line aux 0
line vty 0 4
password lab
login
!
exception core-file mpls/mgx/dumps/rpm-18-110.core
rpmrscprtn PAR 100 100 0 255 0 3840 4080
addcon vpc switch 1.2 30 rslot 0 3 30 master local
addcon vpc switch 1.1 50 rslot 14 1 50
addcon vpc switch 1.3 60 rslot 0 4 60 master local
end

rpm-18-110#
rpm-18-110#

```

PE2 Configuration

This is an example of the PE2 configuration.

```

rpm-18-114#sho run
Building configuration...

Current configuration:
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname rpm-18-114
!
boot system tftp mpls/12.0/rpm-js-mz.120-5.T.bin 3.3.0.1
no logging console
!

```



```
!  
!  
clock timezone EST -5  
clock summer-time EDT recurring  
ip subnet-zero  
no ip domain-lookup  
ip host ios-lab-fw 3.3.0.1  
!  
!  
ip vrf vpn1  
  rd 100:1  
  route-target export 100:1  
  route-target import 100:1  
ip cef  
cns event-service server  
!  
!  
process-max-time 200  
!  
interface Loopback0  
  ip address 10.10.10.10 255.255.255.255  
  no ip directed-broadcast  
!  
interface Loopback1  
  no ip address  
  no ip directed-broadcast  
!  
interface Ethernet1/1  
  ip address 3.3.18.114 255.255.0.0  
  no ip directed-broadcast  
  no ip mroute-cache  
  no keepalive  
!  
interface Ethernet1/2  
  bandwidth 100  
  ip vrf forwarding vpn1  
  ip address 53.0.0.1 255.0.0.0  
  no ip directed-broadcast  
  no ip mroute-cache  
  tag-switching ip  
  no fair-queue  
!  
interface Ethernet1/3  
  no ip address  
  no ip directed-broadcast  
  no ip mroute-cache  
!  
interface Ethernet1/4  
  bandwidth 100  
  no ip address  
  no ip directed-broadcast  
  no ip mroute-cache  
  no fair-queue  
!  
interface Switch1  
  no ip address  
  no ip directed-broadcast  
  no atm ilmi-keepalive  
!  
interface Switch1.1 tag-switching  
  ip unnumbered Loopback0  
  no ip directed-broadcast  
  atm pvc 50 50 0 aal5snap  
  tag-switching atm vp-tunnel 50
```

```

tag-switching ip
!
interface Switch1.2 tag-switching
 ip unnumbered Loopback0
 no ip directed-broadcast
 atm pvc 40 40 0 aal5snap
 tag-switching atm vp-tunnel 40
 tag-switching ip
!
interface Switch1.3 tag-switching
 ip unnumbered Loopback0
 no ip directed-broadcast
 atm pvc 20 20 0 aal5snap
 tag-switching atm vp-tunnel 20
 tag-switching ip
!
router ospf 100
 passive-interface Ethernet1/2
 network 10.0.0.0 0.255.255.255 area 100
 network 53.0.0.0 0.255.255.255 area 100
!
router bgp 100
 no synchronization
 no bgp default ipv4-unicast
 neighbor 11.11.11.11 remote-as 100
 neighbor 11.11.11.11 update-source Loopback0
!
 address-family ipv4 vrf vpn1
  redistribute connected
  redistribute static
 no auto-summary
 no synchronization
 exit-address-family
!
 address-family vpnv4
  neighbor 11.11.11.11 activate
  neighbor 11.11.11.11 send-community extended
 exit-address-family
!
 ip default-gateway 3.3.0.1
 no ip classless
 ip route vrf vpn1 14.0.0.0 255.0.0.0 Ethernet1/2 53.0.0.2
 ip route vrf vpn1 53.0.0.0 255.0.0.0 Ethernet1/2 53.0.0.2
 no ip http server
!
 line con 0
  exec-timeout 0 0
  transport input none
 line aux 0
 line vty 0 4
  password lab
  login
!
 exception core-file mpls/mgx/dumps/rpm-18-114.core
 rpmrscprtn PAR 100 100 0 255 0 3840 4080
 addcon vpc switch 1.3 20 rslot 0 2 20 master local
 addcon vpc switch 1.2 40 rslot 0 1 40
 addcon vpc switch 1.1 50 rslot 10 1 50 master local
end

rpm-18-114#

```

Support for LDP

MPLS label distribution protocol (LDP) allows the construction of highly scalable and flexible IP Virtual Private Networks (VPNs) that support multiple levels of services. LDP provides a standard methodology for hop-by-hop, or dynamic label, distribution in an MPLS network by assigning labels to routes that have been chosen by the underlying Interior Gateway Protocol (IGP) routing protocols. The resulting labeled paths, called label switch paths or LSPs, forward label traffic across an MPLS backbone to particular destinations. These capabilities enable service providers to implement Cisco's MPLS-based IP VPNs and IP+ATM services across multivendor MPLS networks.

LDP is a superset of Cisco's prestandard Tag Distribution Protocol (TDP), which also supports MPLS forwarding along normally routed paths. For those features that LDP and TDP share in common, the pattern of protocol exchanges between network routing platforms is identical. The differences between LDP and TDP for those features supported by both protocols are largely embedded in their respective implementation details, such as the encoding of protocol messages, for example.

This release of IOS, which supports both the LDP and TDP protocols, provides the means for transitioning an existing network from a TDP operating environment to an LDP operating environment. Thus, you can run LDP and TDP simultaneously on any given router platform. The routing protocol that you select can be configured on a per-interface basis for directly- connected neighbors and on a per-session basis for nondirectly connected (targeted) neighbors. In addition, a label switch path (LSP) across an MPLS network can be supported by LDP on some hops and by TDP on other hops.

For more information, including configuration tasks, transitioning a network from TDP to LDP, and command reference, see “MPLS Label Distribution Protocol” in the Cisco IOS Release 12.2T documentation at

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t2/ldp_221t.htm

**Note**

There is no CWM support planned for LDP or TDP.

Support for Multi-VC on the RPM-PR

This feature enables support for initiation of Multiple label switched paths (LSPs) per destination on the RPM. Different label switched paths are established for different class of services. This feature enables interface level queueing rather than per-vc level on the RPM based on MPLS class of service policy.

MPLS quality of service (QoS) functionality enables network administrators to satisfy a wide range of requirements in transmitting IP packets through an MPLS-enabled network. The three primary MPLS QoS offerings made available to customers are:

- Packet classification
- Congestion avoidance
- Congestion management

For more information, see “MPLS QoS Multi-VC Mode for PA-A3” in the Cisco IOS Release 12.2T documentation at

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t2/cos1221t.htm>

**Note**

There is no CWM support for Multi-LVC.
