This chapter covers the following topics:

- **Introduction to ATM-Based MPLS VPNs**—Service providers and carriers that currently provide ATM and Frame Relay services can utilize their existing ATM infrastructure to provide managed VPN services using MPLS. IP transport over ATM networks requires a complex hierarchy of translation protocols to map IP addresses and routing into ATM addressing and routing. MPLS eliminates complexity by mapping IP addressing and routing information directly into ATM switching tables.
- **MPLS and Tag Switching Terminology**—Tag Switching was Cisco's prestandard offering. Cisco has taken the initiative to be fully standards-compliant with respect to MPLS and has migrated many Tag Switching procedures and formats to MPLS standards. IOS supports MPLS commands as well as Tag Switching commands. This section compares MPLS terminology with that used in Tag Switching.
- **Packet-Based MPLS over ATM**—MPLS networks can use conventional ATM switches as a migration step in introducing MPLS to an existing ATM network. They can also be used to backhaul traffic when the access device (CE router) is remote from the Edge LSR, to tunnel through ATM switches between an Edge LSR and an ATM LSR, and to tunnel through ATM switches between ATM LSRs.
- **ATM-Based MPLS**—This section discusses the operation of MPLS over native ATM using ATM LSRs as Provider Edge (PE) routers and WAN switched ATM LSRs as Provider (P) routers. Such an infrastructure offers service providers the QoS levels guaranteed by ATM core networks and completely alleviates the scalability problem posed by the ATM overlay model.
- **Cell Interleaving**—This section discusses the challenges posed by label VC allocation over ATM for multiple sources transmitting data to the same destination. This section also explains how MPLS supports switches that do not have VC merge capability.
- **VC Merge**—This section discusses how VC merge allows ATM LSRs to transmit cells coming from different VCIs over the same outgoing VCI toward the same destination. This helps reduce the number of Label Virtual Circuits required in the MPLS network.
- **Label Virtual Circuits**—ATM Virtual Circuits (VCs) established for MPLS are called Label Virtual Circuits (LVCs). This section discusses how ATM MPLS uses the VCI fields of a few separate VPIs to carry labels. Each label on a link corresponds to a different LVC.
- **Label Switch Controllers**—The Label Switch Controller (LSC) manages the control and forwarding component of the ATM LSR. This section discusses how the ATM LSR differs from an ordinary ATM switch in the way connections are set up.
- **Virtual Switch Interface**—A Virtual Switch Interface (VSI) provides a standard interface so that a resource in the BPX switch can be controlled by an external controller other than the built-in BPX controller card.
- **IP+ATM**—IP+ATM capability of the ATM LSRs can be used to simultaneously provide MPLS service and traditional ATM switching. This is an extremely attractive proposition for service providers that would like to retain their ATM or Frame Relay installed base as well as expand their portfolio to include MPLS-based IP VPN services.
- **Packet-Based MPLS over ATM VPNs**—This section discusses the configuration of packet-based MPLS over ATM virtual circuits.
- **Case Study of a Packet-Based MPLS over ATM VPN**—This section discusses a service provider offering Layer 3 IP VPN services across its MPLS backbone.
- **ATM-Based MPLS VPNs**—This section discusses ATM-based MPLS Virtual Private Networks.
- **Case Study of an ATM-Based MPLS VPN**—This section discusses a service provider offering Layer 3 IP VPN services over its MPLS-enabled ATM backbone.

CHAPTER 6

ATM-Based MPLS VPNs

Introduction to ATM-Based MPLS VPNs

Service providers that currently operate ATM or Frame Relay networks over an ATM backbone can leverage the benefits provided by Multiprotocol Label Switching (MPLS). From a cost perspective, enormous savings can be realized if you do not have to build an MPLS network from the ground up. Service providers and carriers that currently provide ATM and Frame Relay services can utilize their existing infrastructure to provide managed Virtual Private Network (VPN) services using MPLS. This is possible if the ATM switches are MPLS-aware. For non-MPLS ATM switches, MPLS can be configured on MPLS-aware routers, and the underlying ATM Virtual Circuits (VCs) will be considered ATM links.

Deploying MPLS IP VPNs over an MPLS-aware ATM backbone has huge advantages, in the sense that VPN customers can be provisioned with hard QoS guarantees, similar to those found with ATM. If the ATM backbone is being used as an ISP backbone, MPLS provides immediate value by enabling traffic engineering of traffic flows over underutilized paths, thereby optimizing link usage within the network. MPLS solutions give ATM networks the ability to intelligently see IP application traffic as distinct from ATM or Frame Relay traffic. By harnessing the attributes of both IP and ATM, service providers can provision intranet or extranet VPNs.

Without MPLS, IP transport over ATM networks requires a complex hierarchy of translation protocols to map IP addresses and routing into ATM addressing and routing. MPLS eliminates complexity by mapping IP addressing and routing information directly into ATM switching tables. The MPLS label-swapping paradigm is the same mechanism that ATM switches use to forward ATM cells. This solution has the added benefit of allowing service providers to continue offering their current Frame Relay, leased-line, and ATM services portfolio while allowing them to offer differentiated business-quality IP services.

Service providers or carriers that currently operate a Cisco Stratacom-based BPX or MGX network can benefit greatly from the design principles and case study implementations presented in this chapter. The VPN feature of MPLS allows a service provider network to deploy scalable IPv4 Layer 3 VPN backbone services. These services can be deployed over a Layer 3 routed backbone or over an MPLS-aware ATM backbone. This chapter explains deployment over an ATM backbone.

MPLS and Tag Switching Terminology

Cisco has taken the initiative to be fully standards-compliant with respect to MPLS and has migrated many Tag Switching procedures and formats to MPLS standards. IOS supports **mpls** commands as well as **tag-switching** commands. Table 6-1 compares MPLS terminology with that used in Tag Switching.

Table 6-1 *Tag Switching and MPLS Terminology*

Tag Switching Terminology	MPLS Terminology
Tag Switching	MPLS (Multiprotocol Label Switching)
Tag (item or packet)	Label
TDP (Tag Distribution Protocol)	LDP (Label Distribution Protocol)
Tag-switched	Label-switched
TFIB (Tag Forwarding Information Base)	LFIB (Label Forwarding Information Base)
TSR (Tag-Switched Router)	LSR (Label-Switched Router)
TSC (Tag Switch Controller)	LSC (Label Switch Controller)
ATM TSR (ATM Tag-Switched Router)	ATM LSR (ATM Label-Switched Router)
TVC (Tag Virtual Circuit)	LVC (Label Virtual Circuit)
TSP (Tag Switch Path)	LSP (Label-Switch-Path)
XTag ATM (extended Tag ATM port)	XmplsATM (extended MPLS ATM port)

NOTE Cisco TDP (Tag Distribution Protocol) and LDP (MPLS Label Distribution Protocol) are nearly identical in function but use incompatible message formats and some different procedures. Cisco is changing from TDP to a fully-compliant LDP.

MPLS Elements

This section defines multiple MPLS elements. Figure 6-1 illustrates the MPLS elements in a network environment.

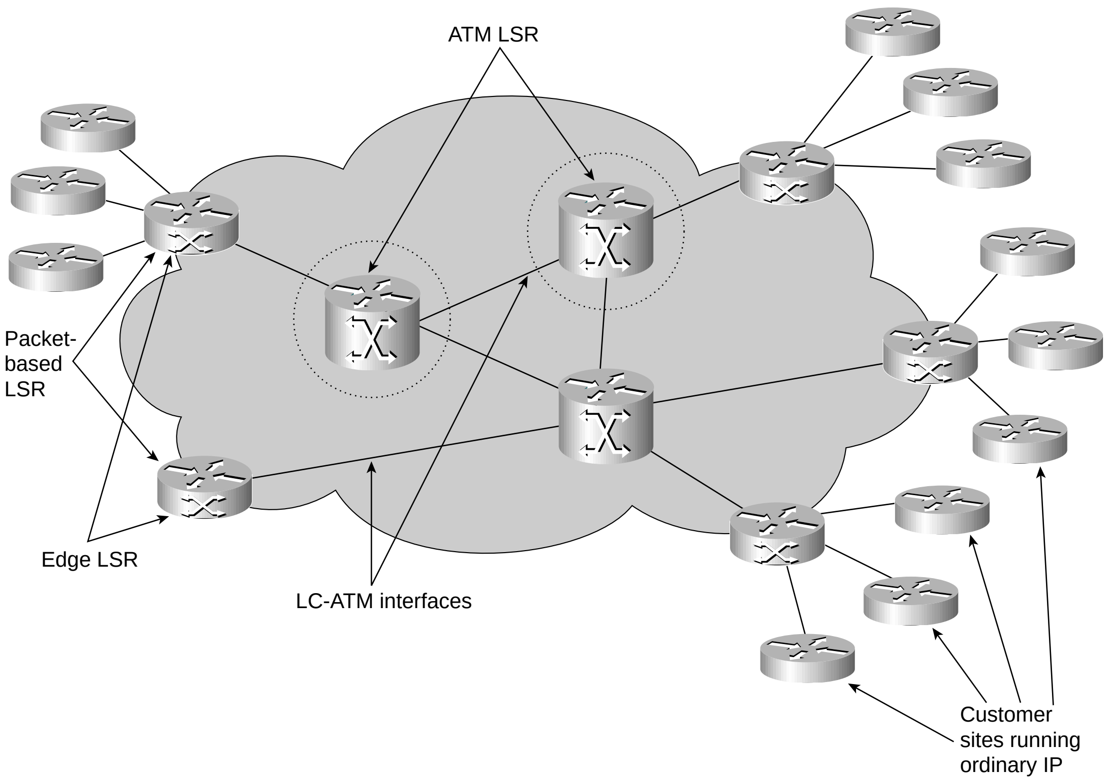
The MPLS elements are as follows:

- **Label-Switched Router (LSR)**—A device that implements the MPLS control and forwarding components as already described.
- **Label-Controlled ATM interface (LC-ATM interface)**—An ATM interface controlled by the MPLS control component. Cells traversing such an interface carry labels in the VCI field of a user-selected range of VPIs. The control component could be integrated in the switch or on an outside controller.

- **ATM LSR**—An LSR based on an ATM switch. It has LC-ATM interfaces.
- **Packet-based LSR**—An LSR that forwards complete packets between its interfaces. A packet-based LSR can have zero or more LC-ATM interfaces. Packet-based LSRs typically consist of MPLS software running on ordinary router platforms, such as the Cisco 3600, 4700, 7200, or 7500 series. Sometimes there are some hardware features specifically for MPLS, as on the Cisco 12000 series.
- **ATM Edge LSR**—A packet-based LSR that is connected to the ATM-LSR cloud via LC-ATM interfaces. The function of the ATM Edge LSR is to add labels to unlabeled packets and to strip labels from labeled packets.

NOTE Edge LSRs are part of the same service provider network as ATM-LSRs. Edge LSRs are not intended to be customer premises equipment (CPE) or customer-located equipment.

Figure 6-1 *MPLS Network Elements*



Packet-Based MPLS over ATM

The operation of MPLS over ATM Private Virtual Circuits (PVCs) results in an overlay model. MPLS is configured on ATM routers, which perform Provider (P) and Provider Edge (PE) router functionality. This model does not realize the full advantages of the underlying ATM QoS. However, for service providers that are running core ATM networks with non-MPLS ATM switches, MPLS can still be deployed to create VPNs or leverage the advantages of traffic engineering.

Service providers can run MPLS in an ATM overlay mode during the transition from an IP over ATM overlay model to a full IP+ATM MPLS model.

The ATM overlay model requires $((n)(n-1))/2$ PVCs in order to form a full mesh, where n is the number of routers in the core. Each router peers directly with $(n-1)$ routers and ends up with $(n-1)$ adjacencies. The amount of link-state routing information that is transmitted in the event of a topology change in the core can be as much as n^4.

This leads to scalability issues with respect to adding routers to the core, because routing traffic in itself can overwhelm the core routers.

Other scalability issues are involved with the ATM overlay model. There are limitations on the number of virtual circuits that can be deployed over a physical interface due to limitations on the switch resources required to support the large number of VCs per interface. Also, if a physical links goes down, it takes a large number of VCs down with it.

MPLS networks can use conventional ATM switches as a migration step to introduce MPLS to an existing ATM network. They can also be used to backhaul traffic when the access device (CE router) is remote from the Edge LSR, to tunnel through ATM switches between an Edge LSR and an ATM LSR, and to tunnel through ATM switches between ATM LSRs.

The label allocation scheme uses independent mode. The LDP relationship is established with unsolicited downstream label distribution. MPLS requires ATM adaptation Layer 5 SNAP (AAL5SNAP) encapsulation in order to run over ATM PVCs. The MPLS over ATM encapsulation technique is shown in Figure 6-2.

Figure 6-2 *Packet-Based MPLS over ATM Encapsulation Technique*

ATM-Based MPLS

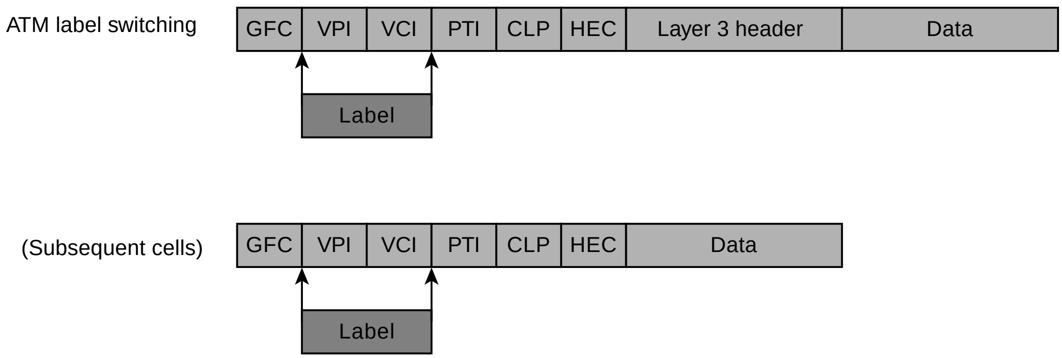
The operation of MPLS over native ATM using ATM LSRs as Provider Edge (PE) routers and WAN switched ATM LSRs as Provider (P) routers offers service providers the QoS levels guaranteed by ATM core networks and completely alleviates the scalability problem posed by the ATM overlay model.

Forwarding Component

The VPI/VCI pair identifies the ATM virtual circuit and is local to an interface. The VPI/VCI value of the incoming cell is used to look up the outgoing interface and outgoing VPI/VCI value. In an ATM environment, the label-switching forwarding function is carried out similarly to normal switching. The label information needed for label switching is carried in the VCI field within one or a small number of VPs. The labels are actually the VCIs. 16 bits in the VCI field of the ATM UNI or NNI header permits 65,536 unique label values for a single VPI.

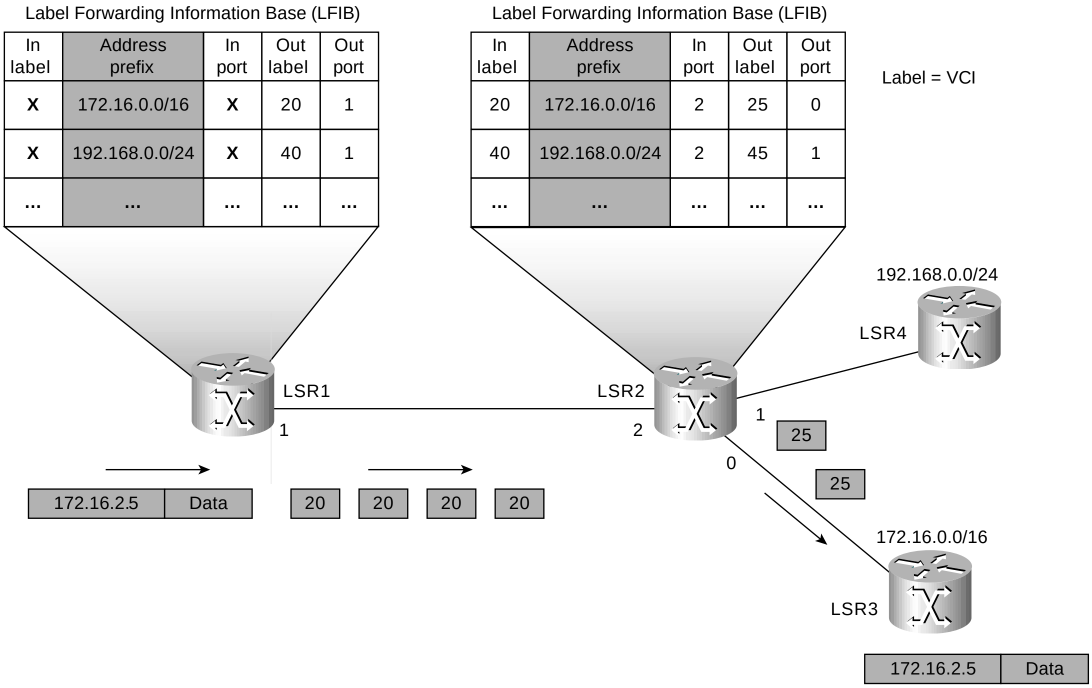
In order to run MPLS, the top label of the label stack is translated into a VCI or VPI/VCI value. The label allocation and distribution procedures are modified so that the ATM LSR looks up the VCI or VPI/VCI label value and determines the outgoing interface and outgoing label. The MPLS ATM encapsulation technique is shown in Figure 6-3.

Figure 6-3 *ATM MPLS Encapsulation Technique*



The ATM LSR is controlled by a routing engine such as a 7500 or 7200 in case of a BPX or the RPM in case of the MGX. In Figure 6-4, an unlabeled IP packet with a destination address of 172.16.2.5 arrives at Edge LSR1. LSR1 looks into its label forwarding information base (LFIB) and matches the destination with prefix 172.16.0.0/16 and a label value of 40. LSR1 sends an ATM adaptation Layer 5 (AAL5) frame as a sequence of cells on VCI 40. LSR2, which is an ATM LSR controlled by a Label Switch Controller (LSC), performs a normal switching operation by switching cells coming on interface 2/VCI 40 to interface 0/VCI 50.

Figure 6-4 *LFIB in an ATM MPLS Environment*



Control Component

The control component of MPLS consists of link-state IP routing protocols such as Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS) running in conjunction with MPLS label allocation and maintenance procedures. The control component is responsible for setting up label forwarding paths along IP routes. The control component also maintains accuracy for the paths, because network topologies are prone to change.

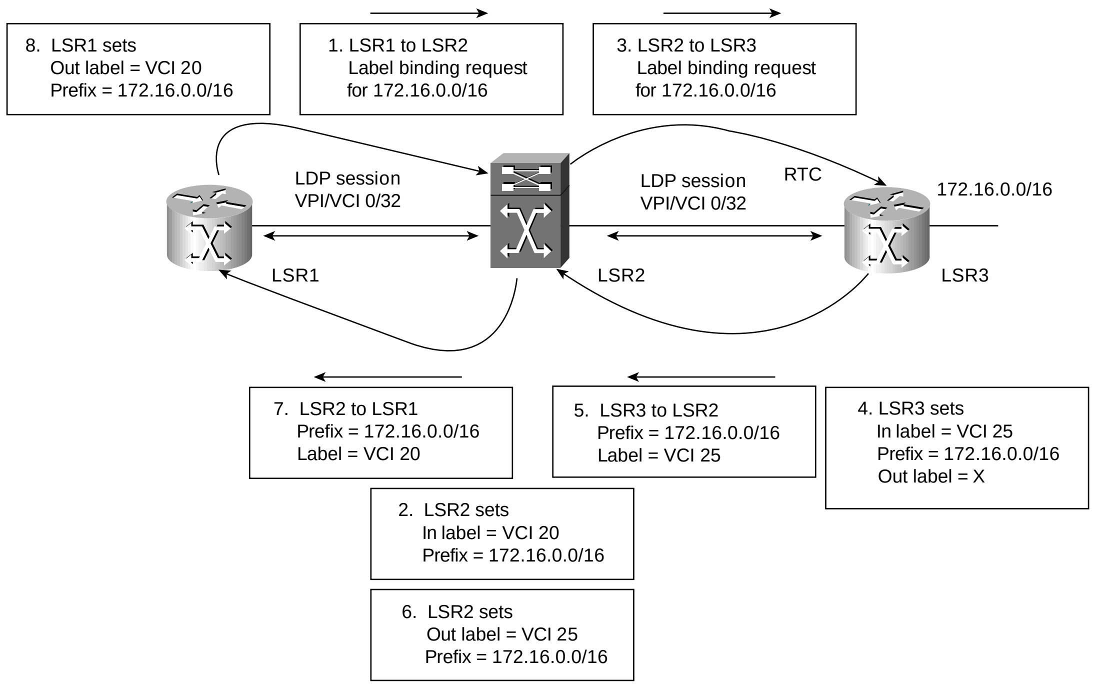
ATM LSRs use the downstream-on-demand allocating mechanism. Each ATM-LSR maintains an LFIB that contains a list of all IP routes that the ATM LSR uses. This function is handled by the routing engine function, which is either embedded in the switch (RPM in case of the MGX) or run on an outside controller (LSC in case of the BPX). For each route in its Forwarding Information Base (FIB), the edge ATM LSR identifies the next hop for a route. It then issues a request via LDP to the next hop for a label binding for that route.

When the next-hop ATM LSR receives the route, it allocates a label and creates an entry in its LFIB with the incoming label changed to the allocated outgoing label. The next action depends on whether the label allocation is in independent mode or ordered mode. In independent mode, it immediately returns the binding between the incoming label and the route to the LSR that sent the request. However, this might mean that it cannot immediately forward labeled packets that arrive, because the ATM-LSR might not yet have an outgoing label/VCI for the route. In ordered mode, it does not immediately return the binding, but waits until it has an outgoing label.

In ordered mode, the next-hop LSR sends a new binding request to its next hop, and the process repeats until the destination ATM Edge LSR is reached. This returns a label binding to the previous ATM-LSR, causing it to return a label binding, and so on, until the label bindings along the path are established.

Figure 6-5 illustrates ordered behavior. ATM edge LSR1 is an IP routing peer to ATM-LSR2. In turn, ATM-LSR2 is an IP routing peer to ATM-LSR3. LSR1-LSR2 and LSR2-LSR3 exchange IP routing updates over VPI/VCI 0/32.

Figure 6-5 *Ordered Mode Downstream-on-Demand Label Allocation*



The following are the steps that occur in Figure 6-5:

1 LSR1 sends a label binding request toward LSR2 in order to bind prefix 172.16.0.0/16 to a specific VCI.

2. LSR2 allocates VCI 20 and creates an entry in its LFIB with VCI 20 as the incoming label.
3. LSR2 sends a bind request toward LSR3.
4. LSR3 issues VCI 25 as a label.
5. LSR3 sends a reply to LSR2 with the binding between prefix 172.16.0.0/16 and the VCI 25 label.
6. LSR2 sets the outgoing label to VCI 25. This information is now used by LSR2 to switch cells coming on VCI 20 to VCI 25.
7. LSR2 sends a reply to LSR1 with the binding between prefix 172.16.0.0/16 and VCI 20.
8. LSR1 creates an entry in its LFIB and sets the outgoing label to VCI 20.

Independent mode operation is similar to that shown in Figure 6-5, except that Steps 7 and 8 might occur concurrently with Step 3. In independent mode, the LSR that initiated the request receives the binding information, creates an entry in its LFIB, and sets the outgoing label in the entry to the value received from the next hop. The next-hop ATM LSR repeats the process, sending a binding request to its next hop. The process continues until all label bindings along the path are allocated.

In optimistic mode, the LSR that initiated the request receives the binding information, creates an entry in its LFIB, and sets the outgoing label in the entry to the value received from the next hop. The next-hop ATM LSR then repeats the process, sending a binding request to its next hop. The process continues until all label bindings along the path are allocated.

In conservative mode, the next-hop LSR sends a new binding request to its next hop, and the process repeats until the destination ATM Edge LSR is reached. It then returns a label binding to the previous ATM-LSR, causing it to return a label binding, and so on, until all the label bindings along the path are established.

Cell Interleaving

Label VC allocation over ATM for multiple sources transmitting data to the same destination causes a few challenges. An ATM LSR that receives binding requests from different upstream neighbors toward the same prefix has to request multiple outbound labels from its downstream neighbor. If the ATM LSR allocates only one outgoing VCI, cells from different AAL5 frames are potentially interleaved and dropped at the receiving end. Allocating different outbound VCIs for the same destination ensures that cells are received in order. This setup is illustrated in Figure 6-6.

Figure 6-6 *Cell Interleaving*

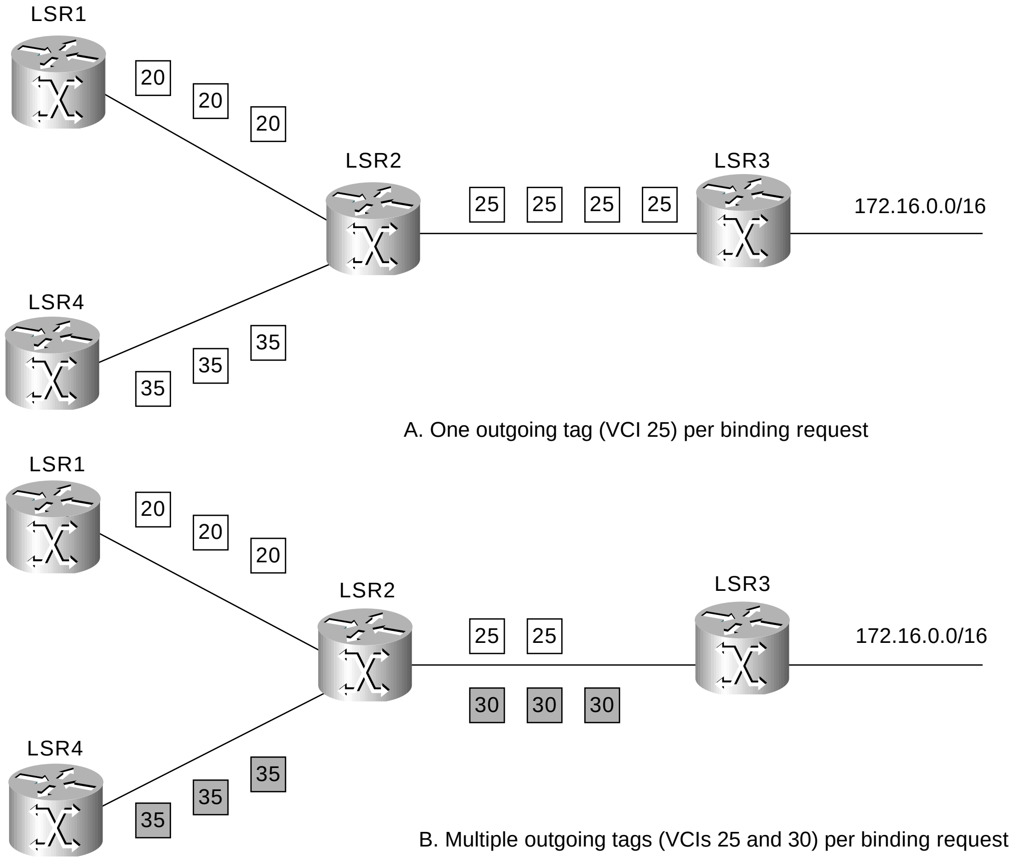


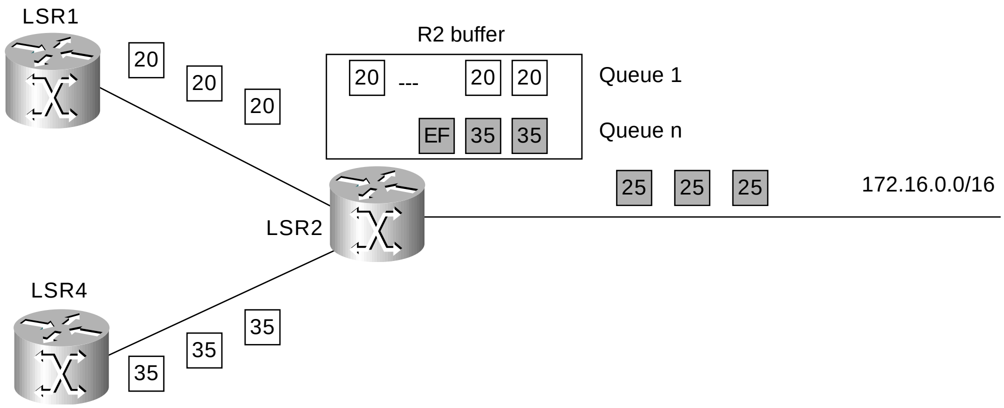
Figure 6-6, topology A shows a hypothetical situation. LSR2 has received two different binding requests for prefix 172.16.0.0/16 from LSR1 and LSR4. LSR2 logically creates two entries in its LFIB and assigns incoming labels for each request. In this example, LSR2 has assigned VCI 20 for LSR1 and VCI 35 for LSR4. In case LSR2 does not already have an outbound label for the prefix, LSR2 sends a binding request toward LSR3 and gets VCI 25 assigned as an outbound label. As a result, cells arriving from LSR1 and LSR4 on VCIs 20 and 35 are sent over VCI 50 and potentially get interleaved, causing AAL5 frames to be discarded.

Topology B shows the same scenario, with the difference that LSR2 requests two outgoing labels for prefix 172.16.0.0/16. LSR2 is assigned two VCIs, 20 and 30. Cells from LSR1 are switched using cross-connect (20, 25), and cells from LSR4 are switched using cross-connect (35, 30). As such, complete noninterleaved AAL5 frames are received at the destination. This example explains how MPLS supports switches that do not have VC merge capability.

VC Merge

VC merge, illustrated in Figure 6-7, allows ATM LSRs to transmit cells coming from different VCIs over the same outgoing VCI toward the same destination. This helps reduce the number of Label Virtual Circuits (LVCs) required in the MPLS network. In other words, it allows multipoint-to-point connections to be implemented by queuing complete AAL5 frames in input buffers until the end of frame is received. The cells from the same AAL5 frame are all transmitted before cells from any other frame are sent. This setup requires more buffering capabilities inside the switch, but no more buffering than is required in IP networks. The small additional delay caused by VC merge is of little concern, because VC merge is designed for IP traffic and does not need to be used for delay-sensitive traffic. IP traffic has good delay tolerance compared to other traffic that might be carried on an ATM network.

Figure 6-7 *VC Merge*



In Figure 6-7, LSR1 and LSR4 are sending traffic toward prefix 172.16.0.0/16. LSR2 has a single outbound VCI 25 bound to that prefix. Cells coming over VCIs 20 and 35 are buffered in separate queues of LSR2 until complete AAL5 frames have been formed. In this example, an end of frame has been detected over VCI 35, and the complete frame has been transmitted over VCI 25. An end of frame has not been detected for cells coming over VCI 20, so these cells are held back in the input buffer, solving the cell interleave problem and minimizing VC usage.

Label Virtual Circuits

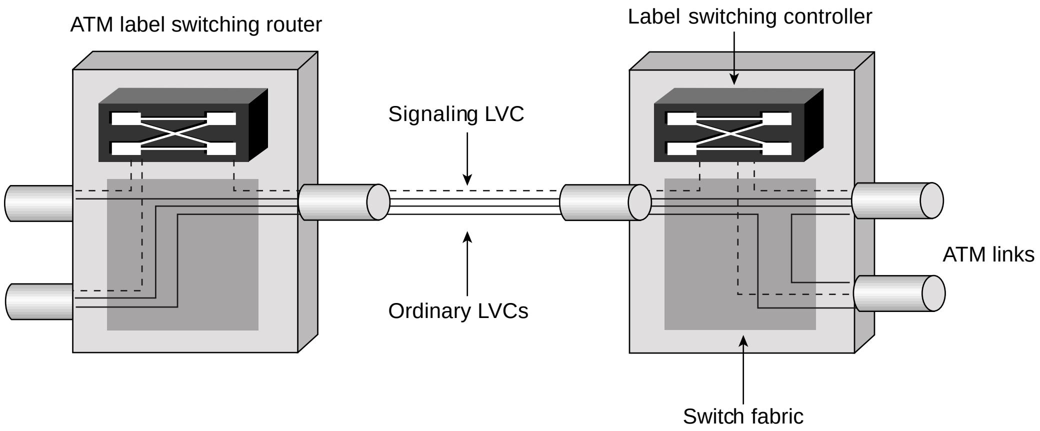
ATM Virtual Circuits established for MPLS are called Label Virtual Circuits. A link between two ATM LSRs, or between an ATM Edge LSR and an ATM LSR, is an ordinary

ATM link. Because ATM MPLS uses the VCI fields of a few separate VPIs to carry a label, each label on a link corresponds to a different LVC. LVCs are neither switched virtual circuits (SVCs) nor permanent virtual circuits (PVCs). They are set up using LDP instead of ATM Forum signaling protocols. LVCs, PVCs, and SVCs may all be used on the same link; however, they use different parts of the VPI/VCI space.

As illustrated in Figure 6-8, at least two distinct types of LVCs are used on each link:

- Signaling LVC
- Ordinary LVC

Figure 6-8 *Label Virtual Circuits*



Signaling LVC

This VC carries IP packets that are reassembled and examined at each ATM LSR. It carries routing information such as MP-BGP, OSPF, IS-IS, and LDP. It might also be used to carry management traffic, such as Simple Network Management Protocol (SNMP) traffic or Internet Control Message Protocol (ICMP) traffic. By default, this VC has VPI and VCI 0/32, which can be reconfigured if desired.

Ordinary LVC

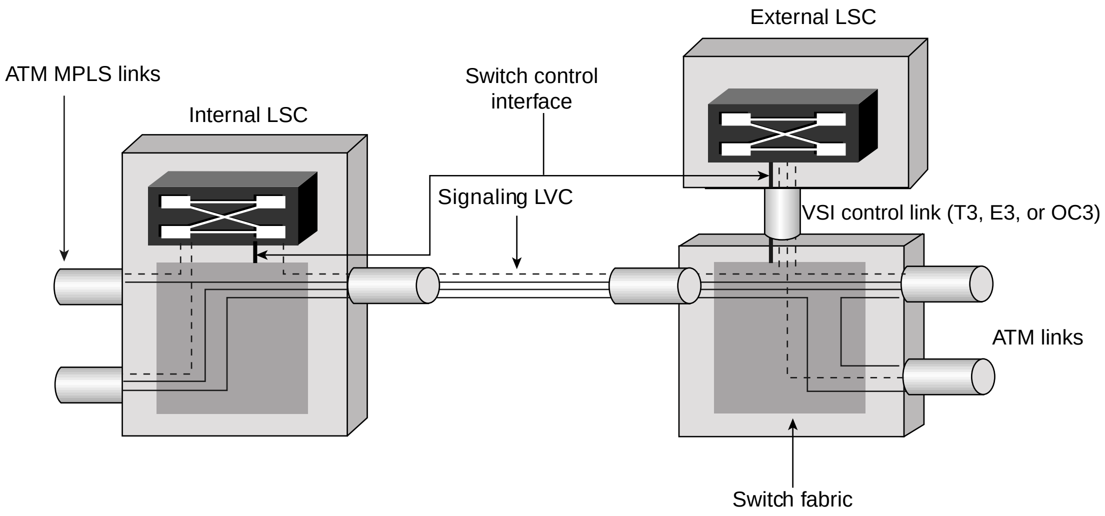
This LVC carries label-switched data. Packets on ordinary LVCs are cross-connected by ATM LSRs without being reassembled. On each link, all ordinary LVCs are within the same VP or a small set of VPs.

Label Switch Controllers

The Label Switch Controller (LSC) manages the control and forwarding component of the ATM LSR. The ATM LSR differs from an ordinary ATM switch in the way connections are set up. Normally an ATM connection is set up by control software running a connection routing protocol such as PNNI or automatic routing management. The LSC is a part of an ATM LSR that runs an IP routing protocol such as OSPF or IS-IS.

In addition to MPLS software, the IP routing software of the LSC maintains knowledge of the MPLS network topology. Using this information, LDP establishes labels (such as VCs) on links connected to the ATM LSR. When the LSC has established incoming and outgoing labels for the same route in its LFIB, it then instructs the switch fabric to set up a connection with the parameters (incoming interface, incoming label VCI, outgoing interface, and outgoing label VCI). Figure 6-9 shows possible locations for the LSC.

Figure 6-9 *Label Switch Controller*



Label Switch Controller Implementation

The LSC can be implemented in a variety of ways. Smaller switches use integrated LSC software, and larger ATM switches use router blades or external router-based LSC architectures.

Integrated LSC software

The LSC could be implemented as integrated software within the ATM switch and might run on the main control card. An example of this kind of implementation is found in the

LightStream 1010, Catalyst 5500, Catalyst 6500, or 8500 ATM-LSR. The LSC software runs on the ATM switch processor, which is the main ATM control card of these units.

Internal LSC

The software runs on an ATM switch shelf card separate from the main switch control card. In the MGX 8800, a route processor module (RPM) card in the switch is used as the LSC. The LSC function is supported on the universal router module (URM) in the IGX 8400 series.

External LSC

The LSC may also be a separate piece of external hardware. The Cisco BPX 8650 ATM LSR switch consists of a BPX 8600 ATM switch shelf and an LSC based on a Cisco 7200 series router. The LSC and switch are interconnected by a switch control link. For the BPX 8650, the switch control link is an ATM link. This link is used in a different way with the other ATM interfaces. On the LSR, it is used to connect the signaling LVCs from all other interfaces on the switch to the LSC, but it does not often carry any data. A similar architecture is also supported in the IGX 8400 series.

An LSC sets up connections in the switch fabric by way of a switch control interface. In the case of the LightStream 1010, Catalyst 5500, Catalyst 6500, or Catalyst 8500, this interface is an internal interface within switch IOS software. In the case of the BPX 8650, IGX 8400 series, and MGX 8850, a switch control interface is used. It is either an external interface or a channel between two cards in the switch.

Figure 6-9 shows how an LSC is connected in a BPX 8650 or IGX 8400 series switch. The physical connection between the LSC and the BPX or IGX series ATM switch shelf is the virtual switch interface (VSI) control link, which is an ATM T3/E3 or OC3 link.

The external LSC model has the advantage of separating the services into logical entities, each having a road map that does not interfere with the other.

If an external router controls a BPX 8650 switch running PNNI and SVC services, an IP MPLS upgrade can be performed without disturbing the operation of the PNNI, PVC, and SVC services. In the WAN space, this is an attractive functionality.

LSC Control of a Switch

The ATM interface on the LSC must be configured as an LSC interface, and a trunk on the ATM switch must be enabled as a control interface. The data connections between the LSC and switch shelf consist of two sets of VCs: signaling LVCs and switch-control VCs.

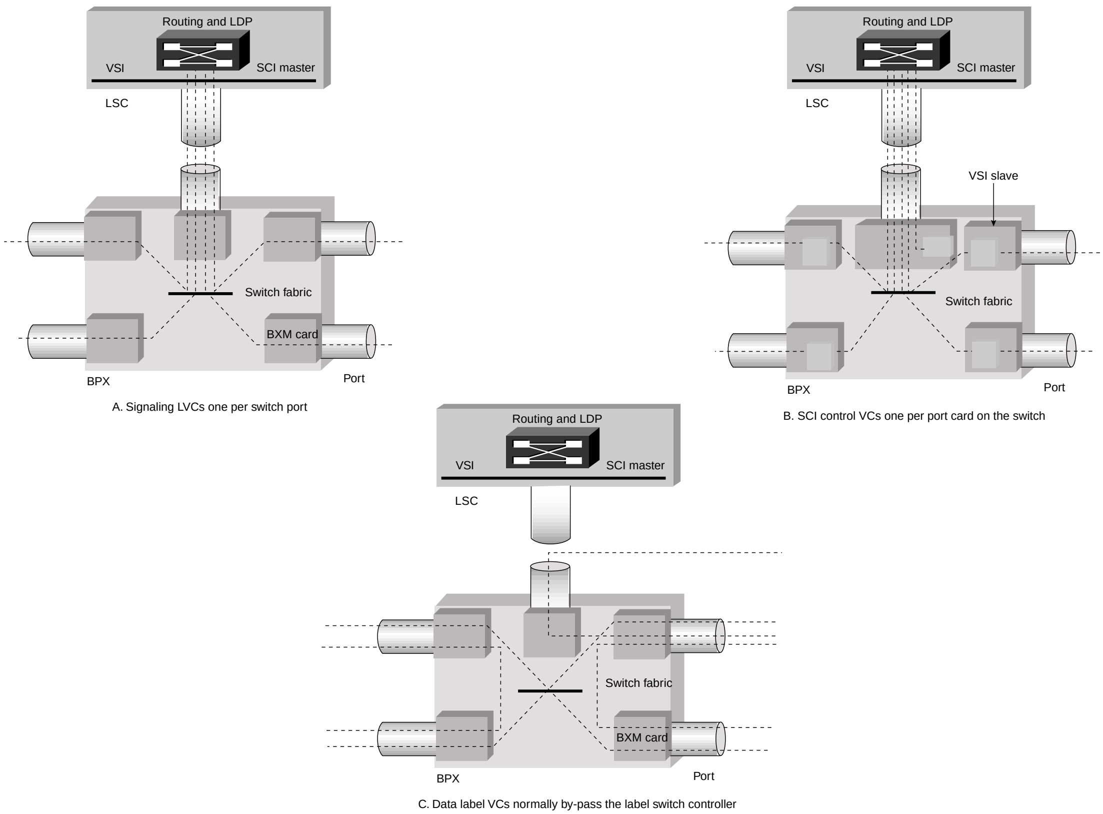
Signaling LVCs

The signaling LVCs from every interface of the ATM switch must be connected through to the LSC, as shown in Figure 6-9. The signaling LVC on each interface is on VPI and VCI (0, 32) by default but is generally cross-connected to a different VCI on the switch control link. This VCI is selected by the LSC software, which requests the setup of the cross-connects as part of its initialization.

Switch-Control VCs

The LSC uses an interface control protocol to discover the switch's port configuration and make switch connections. This protocol operates using VCs connected to each port card, called the external control VCs, as shown in Figure 6-10. In the BPX 8650, there may be up to 12 of these, one for each BXM port card in the BPX8650. In the IGX 8400 series, there may be up to 30, one per UXM card. The external control VCs are set up automatically if external control is enabled.

Figure 6-10 *LSC Signaling and Control Virtual Circuits*

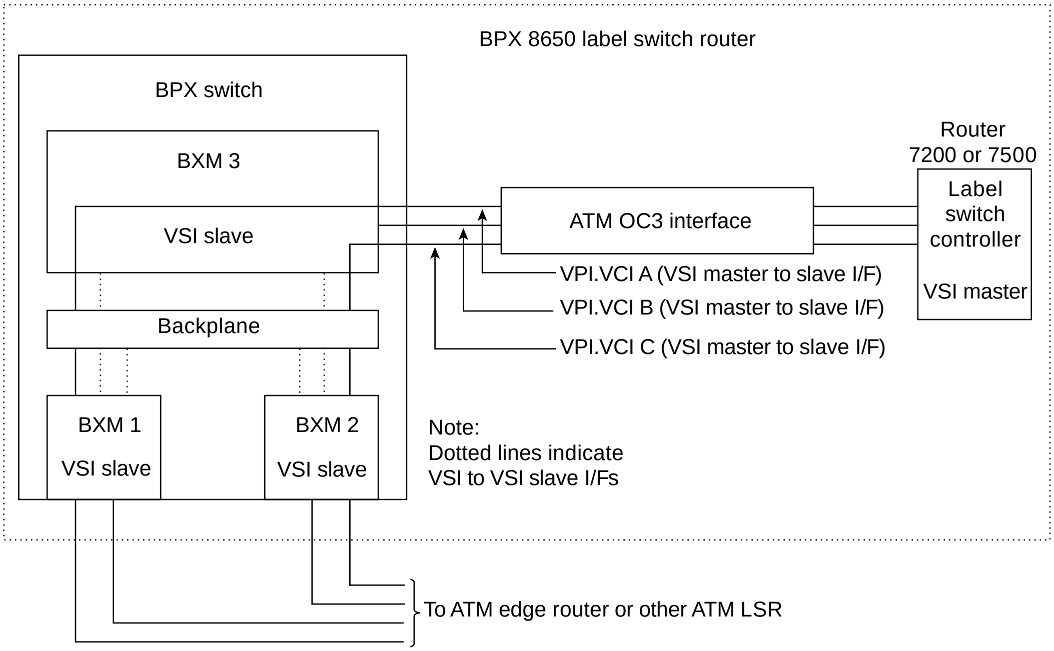


Using the infrastructure of signaling LVCs and external control VCs, the LSC can establish label bindings with the neighboring ATM Edge LSRs and consequently request the setup of LVC cross-connects in the switch. Most data LVCs bypass the LSC.

Virtual Switch Interface

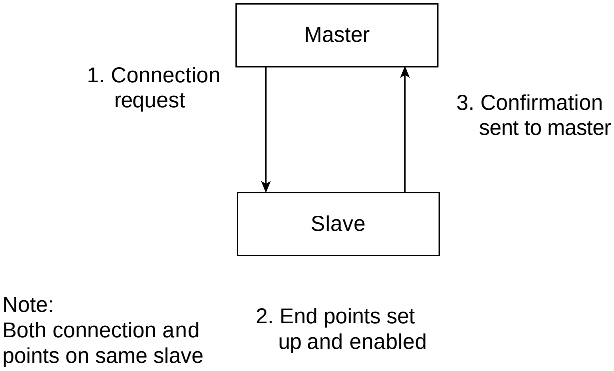
A Virtual Switch Interface (VSI) provides a standard interface so that an external controller other than the built-in BPX controller card can control a resource in the BPX switch. External controllers such as the LSC are usually implemented as an external 7200 or 7500 router. A schematic of such an implementation is shown in Figure 6-11.

Figure 6-11 *Virtual Switch Interface*



A distributed slave model is used to implement VSI in a BPX. Each BXM in a BPX switch is a VSI slave and communicates with the controller and other slaves if needed when processing VSI commands. The VSI master (LSC) sends a VSI message to one slave. Depending on the command, the slave handles the command entirely by itself or communicates with a remote slave to complete the command. For example, a command to obtain configuration information would be completed by one slave only, as shown in Figure 6-12.

Figure 6-12 *Connection Setup with Endpoints on the Same VSI Slave*



However, a command for connection setup would require the local slave in turn to communicate with a remote slave in order to set up both endpoints of the connection. This is demonstrated in Figure 6-13.

Figure 6-13 *Connection Setup with Endpoints on Different VSI Slaves*

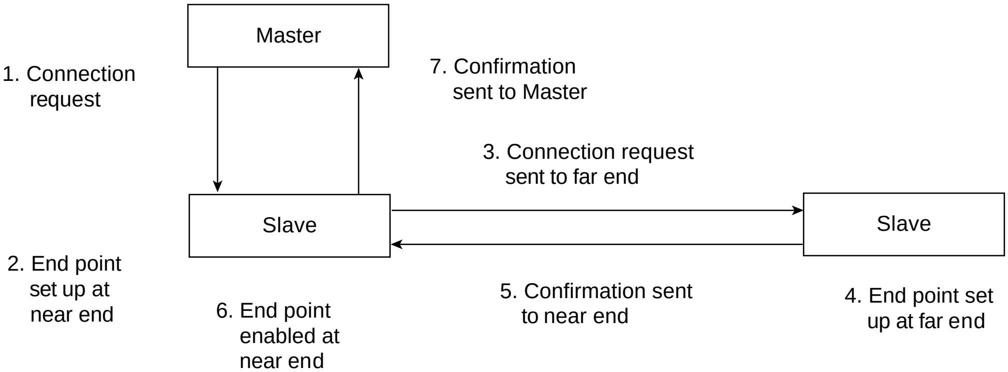
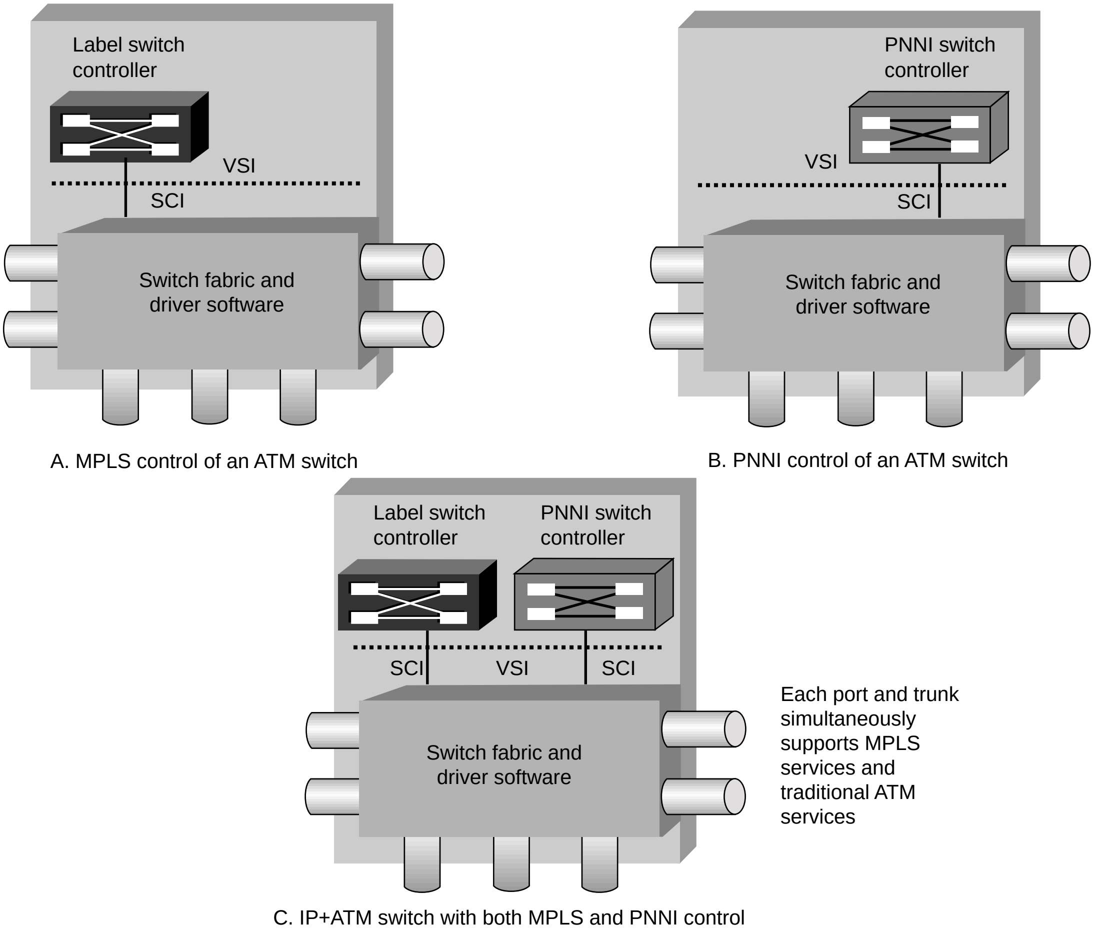


Figure 6-12 shows a simplified example of a connection setup with endpoints on the same VSI slave, and Figure 6-13 shows a connection setup with endpoints on different VSI slaves.

IP+ATM

The LSC can be added to an ATM switch to give it MPLS capability. The IP+ATM capability of the ATM LSRs can be used to simultaneously provide MPLS service and traditional ATM switching. Figure 6-14, Part A shows an ATM switch with an MPLS LSC. Part B of the figure shows a conventional ATM switch under the control of a PNNI controller. As shown in Part C of the figure, IP+ATM switches allow an LSC and a PNNI controller to be simultaneously connected to the same switch. In other words, the same switch can support both optimized IP services using MPLS and conventional ATM services using PNNI.

Figure 6-14 *IP+ATM Capability*



Because IP+ATM switches directly support both MPLS and PNNI services, IP+ATM networks offer both native IP services and native ATM services. An IP+ATM network physically consists of ordinary ATM switches and links. As part of the network's initial configuration, the operator assigns resources of the ATM network to PNNI and MPLS. The resources involved include:

- Bandwidth on links
- VPI/VCI space on links
- VC connection table spaces
- Traffic management

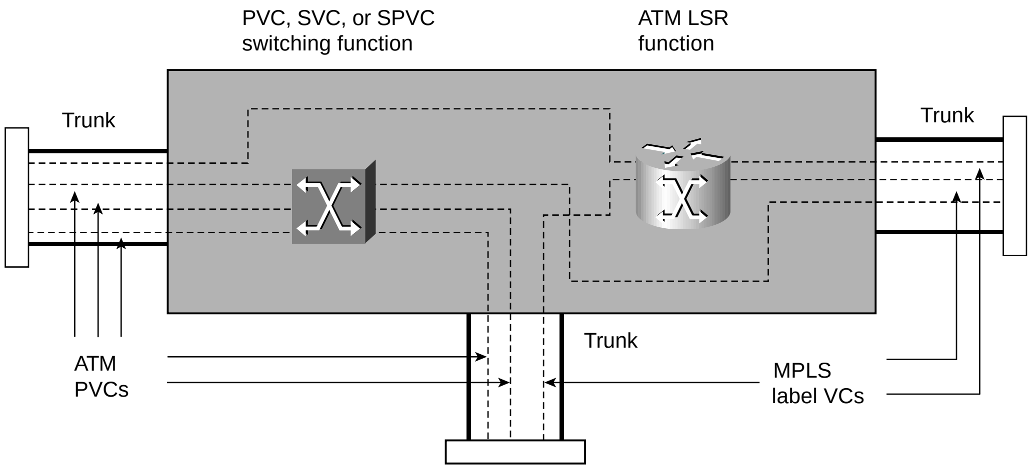
The partitioning of resources is quite flexible, with arbitrary divisions of resources between the different control planes. Partitioning can involve giving fixed allocation of resources to the control planes or can involve a pool of link bandwidth or connection table spaces shared between the control planes. Furthermore, the concept of having controllers independently control a switch extends to more than two controllers. Cisco IP+ATM switches can support four or more control planes.

Structure of an IP+ATM Switch

The concept of an IP+ATM switch is shown in Figure 6-15. A single switch contains two logically separate switches:

- An MPLS ATM LSR optimized for IP transport
- A traditional ATM PVC/SVC switch

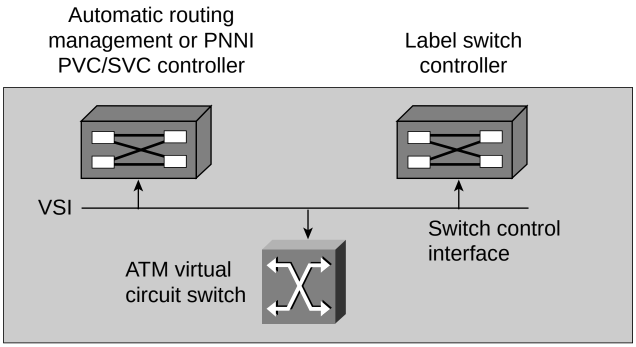
Figure 6-15 *Logical View of an IP+ATM Switch*



Each trunk can support PVCs, SVCs, soft PVCs, and MPLS LVCs.

Although an IP+ATM switch contains logically separate switches, it is physically one switch. However, it contains two or more separate sets of control software. One set of software controls ATM forum PVCs, SVCs, soft PVCs, and Automatic Routing Management, and the other set controls MPLS. These controllers act independently, allowing the single physical switch to act as two or more virtual switches. In switches such as the BPX 8650 and MGX 8850, this independent control is implemented by using the VSI. The VSI allows two or more separate controllers to independently control a single switch, as shown in Figure 6-16.

Figure 6-16 *VSI Controllers*



The MPLS control software is implemented in the LSC. Other VSI controllers may be software running on the switch control card. In the case of the BPX 8650 and MGX 8850, AutoRoute software, which controls PVCs, runs on the switch control card. PNNI control may be added to the BPX 8650 as a separate controller on the Service Expansion Shelf (SES). The LS1010 and 8540 MSR implement functionality similar to the VSI using internal software interfaces.

To ensure that the control planes can act independently, the VSI slave processes in the switch must allocate resources to the different control planes (MPLS or PNNI). In the BPX 8650, resources for AutoRoute PVCs are reserved in a similar way.

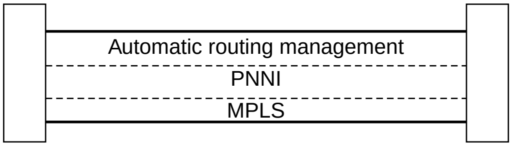
The resources partitioned in the different control planes include the following:

- **VPI/VCI space on trunks**—Each control plane gets a range of VPIs to use.
- **Bandwidth**—Each control plane is guaranteed a certain bandwidth for connection admission control (CAC) purposes. With soft partitioning, a pool of bandwidth can be shared between control planes for CAC purposes. Even with hard partitioning, spare bandwidth unused by a control plane is available on a cell-by-cell basis to other control planes.

- **Traffic queues**—MPLS traffic gets different traffic queues on the switch than the PVC and SVC traffic. This means that MPLS traffic can be handled by queues that directly support the MPLS classes of service (CoSs). The alternative is manually configured translations to ATM forum service types as used in IP over ATM implementations of QoS.

The configuration process for IP+ATM switches includes the assignment of the preceding resources to the different control planes. This involves creating different *partitions* of link resources for the different control planes, as shown in Figure 6-17.

Figure 6-17 *Partitioning of Resources on a Trunk*



IP+ATM Networks

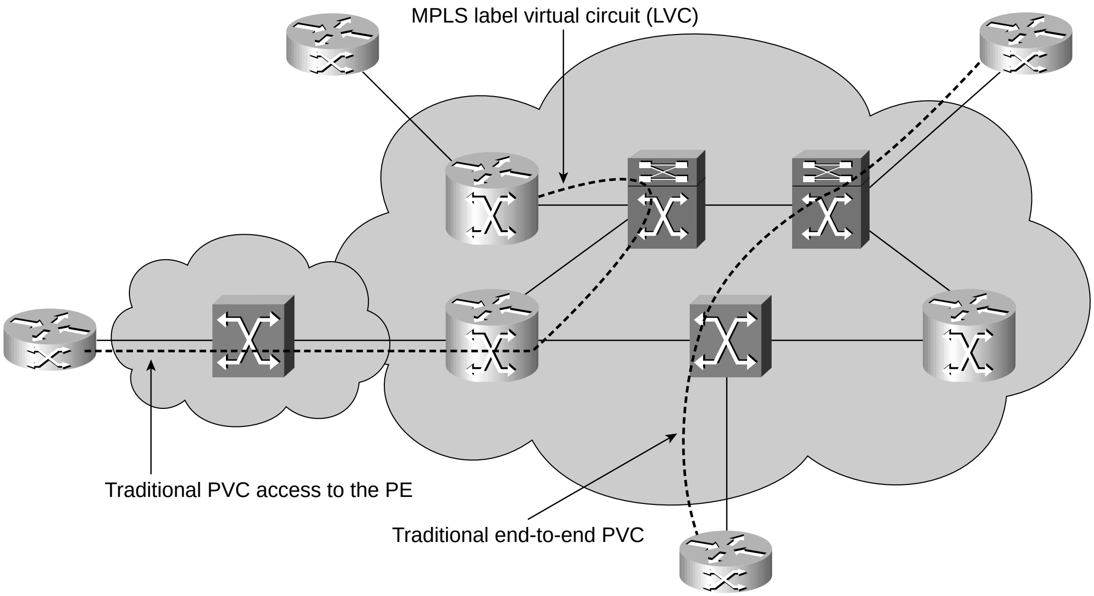
IP+ATM can be used to offer MPLS services, along with PVC and SVC services, on the same network. This means that some switches in the network act as both ATM LSRs and traditional ATM switches, as shown in Figure 6-18.

Traditional ATM services can also be used in conjunction with an MPLS service. Figure 6-18 shows the use of a PVC to connect ordinary IP traffic from a customer site to an ATM Edge LSR. A PVC used in this fashion is called an *MPLS Access PVC*.

Other PVCs are *traditional PVCs* that are part of a traditional end-to-end PVC service. The traffic from the Edge LSR can then be fed back through the ATM LSR function in the same switch that supports the MPLS access PVC, or alternatively through a different switch. In any case, the end-to-end data path for IP traffic can include both MPLS access PVCs and MPLS LVCs.

An integrated IP+ATM edge switch, such as the MGX 8850 or Cisco 6400, supports the ATM LSR function, as well as traditional access switch and PVC switching functions. In addition, the Edge LSR function is integrated into the device. In the MGX 8850, the routing function is supported by RPMs. Node Route Processor (NRP) modules are used in the Cisco 6400. Each RPM or NRP acts as an Edge LSR. In the MGX 8850, one of the RPMs simultaneously acts as an LSC and an Edge LSR.

Figure 6-18 *An IP+ATM Network*



Packet-Based MPLS over ATM VPNs

Service providers that run IP over core ATM networks normally run an IP over ATM model. The ATM forum PVCs constitute the link layer for IP. The *router on a stick* model is still widely used. These ATM switches might or might not be MPLS-aware. LVCs are not built in such environments.

A migration plan is required to fully convert such networks to MPLS. Typically, the first step is to upgrade the router IOS to a stable version that supports MPLS and to configure packet-based MPLS over certain parts of the network. The second migration step entails cutting over the MPLS-capable ATM switches to ATM MPLS and running ATM MPLS. ATM forum PVCs can coexist with MPLS LVCs. This greatly helps with the migration. This step also entails cutting over the IP-based Frame Relay or ATM customers to managed IP MPLS VPN service. This section deals with the configuration of packet-based MPLS over an ATM backbone.

NOTE The configuration of packet-based MPLS using ATM is similar to the VPN configuration discussed in Chapter 5, "Packet-Based MPLS VPNs." The basic steps are outlined in this chapter. This is followed by a case study.

Configuring Packet-Based MPLS over ATM VPNs

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

- MPLS in provider backbone routers
- MPLS with VPN code in provider routers with VPN edge service (PE) routers
- BGP in all routers providing a VPN service
- CEF switching in every MPLS-enabled router
- CoS feature (optional)

Configuration of PE routers

You must perform the following tasks on the PE router to configure and verify MPLS VPN operation:

- Configure your ATM interfaces and IGP
- Define your VPNs
- Configure PE to PE routing sessions
- Configure PE to CE routing sessions. There are four ways to do this:
  - — Static PE to CE routing configuration
  - — RIPv2 PE to CE routing configuration
  - — BGP4 PE to CE routing configuration
  - — OSPF PE to CE routing configuration

Configuration of CE Routers

CE routers can be configured with one of four options:

- Static routing
- RIPv2 routing
- BGP4 routing
- OSPF routing

The PE router must be configured using the same routing protocol as the CE router.

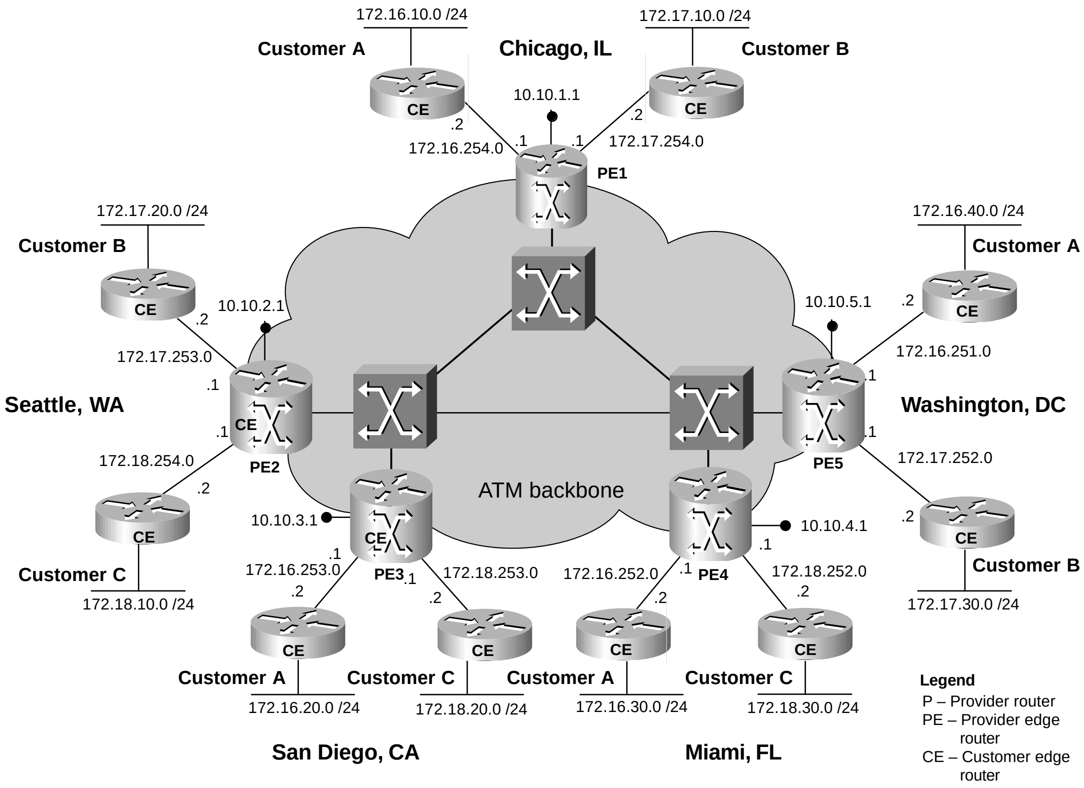
Configuration of P Routers

The provider core routers (P routers) are LSRs that participate in the IGP routing protocol, such as OSPF or IS-IS. However, they do not take part in the multiprotocol IBGP process like the PEs would. In a packet-based ATM configuration, the P router concept is optional, and PE routers can communicate directly over ATM forum PVCs using a combination of the IGP and IBGP.

Case Study of a Packet-Based MPLS over ATM VPN

Consider the service provider shown in Figure 6-19. It has points of presence (PoPs) in Chicago, Seattle, San Diego, Miami, and Washington. The service provider can offer Layer 3 IP VPN services across its MPLS backbone. The service provider offers MPLS VPN services to three customers—A, B, and C. Backbone ATM switches have replaced the core P routers. The customers are each operating a single VPN.

Figure 6-19 *Case Study: Packet-Based MPLS over ATM VPN Configuration*



The service provider has provisioned the VPNs as shown in Table 6-2.

Table 6-2 *Case Study: VPN Deployment (PoP)*

	Customer A	Customer B	Customer C
Chicago, IL	VPN A	VPN B	
Seattle, WA		VPN B	VPN C
San Diego, CA	VPN A		VPN C
Miami, FL	VPN A		VPN C
Washington, DC	VPN A	VPN B	

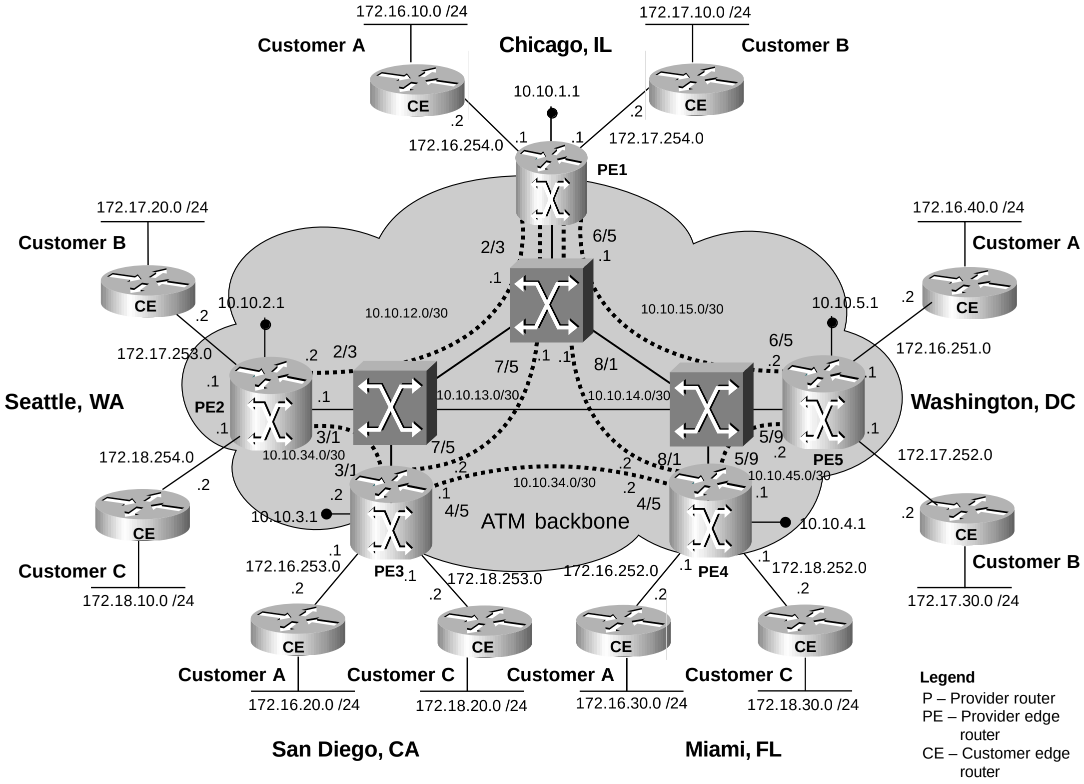
The customer edge (CE) routers connect to the service provider's edge (PE) routers as shown in Table 6-3.

Table 6-3 *Case Study: PE Deployment*

	Customer A	Customer B	Customer C
Chicago, IL	PE1	PE1	
Seattle, WA		PE2	PE2
San Diego, CA	PE3		PE3
Miami, FL	PE4		PE4
Washington, DC	PE5	PE5	

Layer 3 Interior Gateway Routing protocols run across the backbone normally provide any-to-any connectivity for IBGP peers to communicate with each other, even though the IBGP peers (PE routers) are not directly connected to each other. Refer to Figure 6-20 for the PVC configuration in this case study.

Figure 6-20 *Case Study: PVC Configuration*



The IP addressing scheme used in the case study is explained in Table 6-4.

Table 6-4 *Case Study: VPN IP Address Architecture*

	PE Router	**CE WAN Subnet**	**CE LAN Subnet**
Customer A			
Chicago	10.10.1.1/32	172.16.254.0/24	172.16.10.0/24
Seattle	No presence	No presence	No presence
San Diego	10.10.3.1/32	172.16.253.0/24	172.16.20.0/24
Miami	10.10.4.1/32	172.16.252.0/24	172.16.30.0/24
Washington	10.10.5.1/32	172.16.251.0/24	172/16.40.0/24

continues

Table 6-4 *Case Study: VPN IP Address Architecture (Continued)*

	PE Router	CE WAN Subnet	CE LAN Subnet
Customer B			
Chicago	10.10.1.1/32	172.17.254.0/24	172.17.10.0/24
Seattle	10.10.2.1/32	172.17.253.0/24	172.17.20.0/24
San Diego	No presence	No presence	No presence
Miami	No presence	No presence	No presence
Washington	10.10.5.1/32	172.17.252.0/24	172/17.30.0/24
Customer C			
Chicago	No presence	No presence	No presence
Seattle	10.10.2.1/32	172.18.254.0/24	172.18.10.0/24
San Diego	10.10.3.1/32	172.18.253.0/24	172.18.20.0/24
Miami	10.10.4.1/32	172.18.252.0/24	172.18.30.0/24
Washington	No presence	No presence	No presence

Provider Edge Configuration

The PE router configurations for the ATM interfaces are shown in the following sections. The detailed configurations are similar to the PE configurations in Chapter 5. The ATM PVCs provide Layer 2 connectivity between the various PE routers. The PE routers run an IGP and multiprotocol IBGP within the cloud.

Chicago Configuration

Refer to Figure 6-20 for the Chicago area connectivity and addressing information. Chicago uses VPI/VCI 2/3 as a virtual circuit to establish Layer 2 connectivity with Seattle, VPI/VCI 7/5 to establish Layer 2 connectivity with San Diego, VPI/VCI 8/1 to establish Layer 2 connectivity with Miami, and VPI/VCI 6/5 to establish Layer 2 connectivity with Washington.

Chicago PE Configuration

The following is the Chicago PE configuration:

```
!
interface atm1/0/0
 no ip address
!
interface atm1/0/0.1 point-to-point
```

```
 description atm pvc to Seattle
 ip address 10.10.12.1 255.255.255.252
 pvc 2/3
 encapsulation aal5snap
 tag-switching ip
!
interface atm1/0/0.2 point-to-point
 description atm pvc to San Diego
 ip address 10.10.13.1 255.255.255.252
 pvc 7/5
 encapsulation aal5snap
 tag-switching ip
!
interface atm1/0/1
 no ip address
!
interface atm1/0/1.1 point-to-point
 description atm pvc to Miami
 ip 10.10.14.1 255.255.255.252
 pvc 8/1
 encapsulation aal5snap
 tag-switching ip
!
interface atm1/0/1.2 point-to-point
 description atm pvc to Washington
 ip 10.10.15.1 255.255.255.252
 pvc 6/5
 encapsulation aal5snap
 tag-switching ip
!
```

Seattle Configuration

Refer to Figure 6-20 for the Seattle area connectivity and addressing information. Seattle uses VPI/VCI 2/3 as a virtual circuit to establish Layer 2 connectivity with Chicago and VPI/VCI 3/1 to establish Layer 2 connectivity with San Diego.

Seattle PE Configuration

The following is the Seattle PE configuration:

```
!
interface atm1/0/0
 no ip address
!
interface atm1/0/0.1 point-to-point
 description atm pvc to Chicago
 ip address 10.10.12.2 255.255.255.252
 pvc 2/3
```

continues

```
 encapsulation aal5snap
 tag-switching ip
!
interface atm1/0/0.2 point-to-point
 description atm pvc to Miami
 ip address 10.10.23.1 255.255.255.252
 pvc 3/1
 encapsulation aal5snap
 tag-switching ip
!
```

San Diego Configuration

Refer to Figure 6-20 for the San Diego area connectivity and addressing information. San Diego uses VPI/VCI 3/1 as a virtual circuit to establish Layer 2 connectivity with Seattle, VPI/VCI 7/5 to establish Layer 2 connectivity with Chicago, and VPI/VCI 4/5 to establish Layer 2 connectivity with Miami.

San Diego PE Configuration

The following is the San Diego PE configuration:

```
!
interface atm1/0/0
 no ip address
!
interface atm1/0/0.1 point-to-point
 description atm pvc to Seattle
 ip 10.10.23.2 255.255.255.252
 pvc 3/1
 encapsulation aal5snap
 tag-switching ip
!
interface atm1/0/0.2 point-to-point
 description atm pvc to Chicago
 ip 10.10.13.2 255.255.255.252
 pvc 7/5
 encapsulation aal5snap
 tag-switching ip
!
interface atm1/0/0.3 point-to-point
 description atm pvc to Miami
 ip 10.10.34.1 255.255.255.252
 pvc 4/5
 encapsulation aal5snap
 tag-switching ip
!
```

Miami Configuration

Refer to Figure 6-20 for the Miami area connectivity and addressing information. Miami uses VPI/VCI 4/5 as a virtual circuit to establish Layer 2 connectivity with San Diego, VPI/VCI 8/1 to establish Layer 2 connectivity with Chicago, and VPI/VCI 5/9 to establish Layer 2 connectivity with Washington.

Miami Configuration

The following is the Miami PE configuration:

```
!
interface atm1/0/0
 no ip address
!
interface atm1/0/0.1 point-to-point
 description atm pvc to San Diego
 ip 10.10.34.2 255.255.255.252
 pvc 4/5
 encapsulation aal5snap
 tag-switching ip
!
interface atm1/0/0.2 point-to-point
 description atm pvc to Chicago
 ip 10.10.14.2 255.255.255.252
 pvc 8/1
 encapsulation aal5snap
 tag-switching ip
!
interface atm1/0/0.3 point-to-point
 description atm pvc to Washington
 ip 10.10.45.1 255.255.255.252
 pvc 5/9
 encapsulation aal5snap
 tag-switching ip
!
```

Washington Configuration

Refer to Figure 6-20 for the Washington area connectivity and addressing information. Washington uses VPI/VCI 5/9 as a virtual circuit to establish Layer 2 connectivity with Miami and VPI/VCI 6/5 to establish Layer 2 connectivity with Chicago.

Washington PE Configuration

The following is the Washington PE configuration:

```
!
interface atm1/0/0
 no ip address
!
interface atm1/0/0.1 point-to-point
 description atm pvc to Miami
 ip 10.10.45.2 255.255.255.252
 pvc 5/9
 encapsulation aal5snap
 tag-switching ip
!
interface atm1/0/1
 no ip address
!
interface atm1/0/1.1 point-to-point
 description atm pvc to Chicago
 ip 10.10.15.2 255.255.255.252
 pvc 6/5
 encapsulation aal5snap
 tag-switching ip
!
```

ATM-Based MPLS VPNs

In an ATM-based MPLS configuration, ATM Forum PVCs are unnecessary. MPLS LVCs are automatically brought up after the switch has its resources partitioned for LVCs and other parameters have been appropriately set up.

The VPN feature of MPLS allows several sites to transparently interconnect through a service provider's network. One service provider network can support several different IP VPNs. Each of these VPNs appears to its users as a closed user group private network, separate from all other networks. Within a VPN, each site can send IP packets to any other site in the same VPN. Each VPN is associated with one or more VPN routing or forwarding instances (VRFs). A VRF consists of an IP routing table, a derived Cisco express forwarding (CEF) table, and a set of interfaces that use this forwarding table. The PE router maintains a separate routing and CEF table for each VRF. This prevents information from being sent outside the VPN and allows the same subnet to be used in several VPNs without causing duplicate IP address problems. The PE routers use Multiprotocol Interior Border Gateway Protocol (MP-IBGP) to distribute the VPN routing information using the BGP extended communities attributes.

NOTE The MPLS VPN integration function resides only on the PE routers of the MPLS network. The MPLS cloud is composed of MPLS-aware ATM switches such as the BPX 8650, MGX 8850, 8540 MSR, or LightStream 1010. The Cisco Stratacom BPX 8650 is the recommended platform for industrial-strength core ATM switched networks. The PE feeder nodes could be a combination of MGX 8850s, 8540 MSRs, LS1010, 10000, 12000 GSRs, 7500, or 7200 ATM routers.

NOTE If you plan to use the LightStream 1010 as an ATM MPLS switch, it is recommended that you use software version WA4.8d or higher.

Configuring ATM-Based MPLS VPNs

Your network must be running the following Cisco IOS and software services before you configure VPN operation:

- MPLS in provider backbone ATM switches
- MPLS with VPN code in provider routers with VPN edge service (PE) routers
- BGP in all routers providing a VPN service
- CEF switching in every MPLS-enabled router
- CoS feature (optional)

Configuration of PE Routers

The following tasks must be performed on the PE router to configure and verify MPLS VPN operation:

- Configure your ATM interfaces and IGP
- Define your VPNs
- Configure PE to PE routing sessions
- Configure PE to CE routing sessions. There are four ways to do so:
    - — Static PE to CE routing configuration
    - — RIPv2 PE to CE routing configuration
    - — BGP4 PE to CE routing configuration
    - — OSPF PE to CE routing configuration

Configuration of CE Routers

CE routers can be configured with one of four options:

- Static routing
- RIPv2 routing
- BGP4 routing
- OSPF routing

The PE router must be configured using the same routing protocol as the CE router.

Configuration of ATM MPLS Core Switches

The provider core devices are ATM LSRs that participate in the IGP routing protocol, such as OSPF or IS-IS. However, they do not take part in the multiprotocol IBGP process like the PEs would. In the ATM-based MPLS model, the P routers are implemented as LSCs working in conjunction with the ATM switches, providing Layer 3/MPLS control of the ATM switch using a VSI. This chapter focuses on the use of the BPX 8650 as an ATM MPLS core switch.

BPX 8650

The Cisco BPX 8600 series is a standards-based ATM switch with ATM and MPLS capabilities. The switch offers up to 20 Gbps of switching for multiple traffic types, data, voice, and video. The BPX 8600 series supports MPLS. This functionality can easily be added to any BPX switch already installed in the field with a software upgrade and the addition of a 7200 or 7500 LSC.

The BPX VSI allows controllers to set up and tear down virtual circuit converters through the BPX independent of the control protocol (PNNI, MPLS, SS7) and independent of whether the controller is internal or external to the chassis. Thus, the VSI allows multiple controllers to control the BPX. The controllers are optimized for the service to be delivered. The VSI manages the resource allocation, so the controllers are independent, and each service receives the QoS required.

The 7200 or 7500 LSC uses the VSI to provide MPLS control of the BPX. With a co-located Extended Services Processor (ESP), the BPX switch adds the capability to support ATM and Frame Relay SVCs and soft permanent virtual circuits (SPVCs). The ESP is an adjunct processor that is co-located with a BPX switch shelf. The ESP provides the signaling and Private Network-to-Network Interface (PNNI) routing for ATM and Frame Relay SVCs via BXM cards in the BPX switch and AUSM and FRSM cards in the MGX 8220. Frame Relay to ATM Interworking is performed in accordance with RFC 1490 and RFC 1483.

The BPX 8600 series switch is designed for high levels of reliability. All system components can be configured for 100-percent redundancy and are hot-swappable. Automatic Routing Management reroutes virtual circuits if a trunk fails. Software upgrades can be performed in the background, and the conversion to a new software release is achieved without interruption of traffic or loss of data. All broadband interfaces can be configured for 1:1 redundancy, and narrowband modules can be configured for 1:*n* redundancy.

The BPX 8600 series switch incorporates Stratacom technology implemented in custom silicon application-specific integrated circuits (ASICs) in the broadband switch modules (BXM). This supports traffic management, per-virtual-circuit queuing, CoS management, and multicasting. It also provides full virtual source/virtual destination (VS/VD) implementation of the ATM Forum's Traffic Management Specification V.4.0, as well as supporting explicit rate marking and explicit forward congestion indication (EFCI) tagging. The BPX supports Inverse ATM as well as virtual trunking, which provides the ability to define multiple trunks within a single physical trunk port interface. The configuration of the BPX 8650 and MGX 8850 core ATM MPLS switches is detailed in the following sections.

BPX 8650 Configuration

The BPX nodes need to be set up and configured in the ATM network, including links to other nodes, and so on. Following this, they may be configured for MPLS operation. In configuring the BPX nodes for operation, a virtual interface and associated partition are set up with the **cnfrsrc** command. The LSC is linked to the BPX with the **addshelf** command to allow the router's label switch controller function to control the MPLS operation of the BPX node. The partition's resources, such as bandwidth, VPI range, and the number of logical connection numbers (LCNs), may be distributed between the associated ports. The VPIs are of local significance, so they do not have to be the same for each port in a node, but it is generally convenient from a tracking standpoint to keep them the same for a given BPX node. In this example, it is assumed that a single external controller per node is supported, so the partition chosen is always 1. With the appropriate release of switch software, firmware, and IOS, service class templates are supported.

Step 1 Display the status of all cards as follows:

```
:dspcds
```

BXM cards that you are configuring should be *standby* or *active*. If they are not, perform a hard reset. For example, **resetcd 5 h** resets card 5.

Step 2 Check card connection capabilities using the **dspcd** command as follows:

```
:dspcd <card_number>
```

For example:

```
:dspcd 5
```

This elicits the following system response:

```
bpx2       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:10 EST

Detailed Card Display for BXM-155 in slot 5
```

continues

```
Status:              Active
Revision:            CD18
Serial Number:       783314
Fab Number:          28-2158-02
Queue Size:          228300
Support:             FST,4 Pts,OC3,Vc
Chnls:16320, PG[1}:7588, PG[2]:7588
PG [1} : 1,2,
PG [2] : 3,4,
Backcard Installed
  Type:          LM-BXM
  Revision:      BA
  Serial Number: 784533
  Supports: 4 pts, OC3, SMF Md


Last Command: dspcd 5


Next Command:
```

This example shows that ports 1 and 2 together have a total of 7588 connections or *channels* available for use. Ports 1 and 2 form a port group (PG). Similarly, ports 3 and 4 are a port group with a limit of 7588 connections.

NOTE The connections just shown are used for PVCs, VSI connections, and internal signaling. Unless there is a good reason to do otherwise, it is best to leave many of the LCNs as spares. LCN connections can be allocated to MPLS on each port using the **cnfrsrc** command.

Step 3 Enable the BXM trunk interfaces as follows:

```
:uptrk slot.port.[virtual trk]
```

A BXM interface is a *trunk* if it connects to another BPX, MGX, or MGX 8220 (AXIS) feeder. The VSI connection to an LSC is also a *trunk*.

NOTE The virtual trunk configuration is available from Release 9.2 onward. The **uptrk** command, for example, would be of the form **uptrk 9.1.1** for port interface 9.1, virtual trunk 1. Either ports or trunks can be active simultaneously on the same BXM.

The following is an example:

```
:uptrk 5.1
:uptrk 9.1
:uptrk 9.2
```

This elicits the following system response:

```
bpx2       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:20 EST

TRK       Type        Current Line Alarm Status          Other End
9.1       OC3         Clear  -  OK                       bpx1/9.1
9.2       OC3         Clear  -  OK                       bpx2/9.2
5.1       OC3         Clear  -  OK                       VSI(VSI)









Last Command: uptrk 9.2


Next Command:
```

In this example, trunk 5.1 is the link to the LSC controller, and trunks 9.1 and 9.2 are used as the broadband trunks to other BPX nodes on the network. Trunks 5.2 and 5.3 connect to PE routers and also need to be *upped*.

Step 4 Use the **cnfrsrc** command to configure partition resources for Automatic Routing Management PVCs or VSI-MPLS as follows:

```
:cnfrsrc <slot.port.{virtual trunk}> <maxpvclcns> <maxpvcbw> partitionID <e/d>
    <minvsilcns> <maxvsilcns> <vsistartvpi> <vsiendvpi> <vsiminbw> <vsimaxbw>
```

Table 6-5 describes the parameters used in the **cnfrsrc** command.

Table 6-5 **cnfrsrc** *Command Parameters*

Parameter	Description
slot.port	Specifies the slot and port number for the BXM card in the BPX.
virtual trunk	Specifies the virtual trunk number for the BXM slot and port in the BXM card.
maxpvclcns *a(x)*	The maximum number of LCNs allocated for AutoRoute PVCs for this port. For trunks, additional LCNs are allocated for AutoRoute that are not configurable. The **dspcd** *slot* command displays the maximum number of LCNs configurable via the **cnfrsrc** command for the given port. For trunks, configurable LCNs represent the LCNs remaining after the BCC has subtracted the additional LCNs needed. For a port card, a larger number is shown, as compared with a trunk card. Setting this field to 0 enables the configuration of all the configurable LCNs to the VSI. The variable *a(x)* is the mathematical representation of **maxpvclcns** in integer format.

continues

Table 6-5 **cnfrsrc** *Command Parameters (Continued)*

Parameter	Description
maxpvcbw	The maximum bandwidth of the port allocated for AutoRoute use.
partitionID	The partition number.
e/**d**	Enables or disables the VSI partition.
minvsilcns *n(x)*	The minimum number of LCNs guaranteed for this partition. The VSI controller guarantees at least this many connection endpoints in the partition, provided that there are sufficient free LCNs in the common pool to satisfy the request at the time the partition is added. When a new partition is added or the value is increased, existing connections might have depleted the common pool so that there are not enough free LCNs to satisfy the request. The BXM gives priority to the request when LCNs are freed. The net effect is that the partition might not receive all the guaranteed LCNs (min LCNs) until other LCNs are returned to the common pool. This value may not be decreased dynamically. All partitions in the same port group must be deleted first and reconfigured in order to reduce this value. The value may be increased dynamically. However, this might cause the deficit condition just described. The command-line interface warns the user when the action is invalid, except for the "deficit" condition. To avoid this deficit condition, which could occur with maximum LCN usage by a partition or partitions, it is recommended that you configure all partitions before adding connections. Also, it is recommended that you configure all partitions before adding a VSI controller via the **addshelf** command. The variable *n(x)* is the mathematical representation of **minvsilcns** in integer format.
maxvsilcns *m(x)*	The total number of LCNs that the partition is allowed for setting up connections. The min LCNs is included in this calculation. If max LCNs equals min LCNs, the max LCNs are guaranteed for the partition. Otherwise, max – min or sum(*m(x)*) – sum(*n(x)*). LCNs are allocated from the common pool on a FIFO basis. If the common pool is exhausted, new connection setup requests are rejected for the partition, even though the max LCNs has not been reached. This value may be increased dynamically when there are enough unallocated LCNs in the port group to satisfy the increase. The value may not be decreased dynamically. All partitions in the same port group must be deleted first and reconfigured in order to reduce this value. Different types of BXM cards support different maximums. If you enter a value greater than the allowed maximum, a message is displayed with the allowable maximum. The variable *m(x)* is the mathematical representation of **maxvsilcns** in integer format.

Table 6-5 **cnfrsrc** *Command Parameters (Continued)*

Parameter	Description
vsistartvpi	By default, the LSC (7200 or 7500 series router) uses either a starting VSI VPI of 1 or 2 for Tag Switching—whichever is available. If both are available, a starting VSI VPI of 1 is used. The VPI range should be 2–15 on a BPX 8620 VSI. The VSI range for Tag Switching on the BPX 8620 is configured as a VSI partition—usually VSI partition 1. VSI VPI 1 is reserved for AutoRoute PVCs, so the VSI partition for Tag Switching should start at VPI 2. If VPI 2 is not to be used, the Tag Switching VPI interface configuration command can be used on the TSC to override the defaults.
vsiendvpi	Two VPIs are sufficient for the current release, although it might be advisable to reserve a larger range of VPIs for later expansion, such as VPIs 2–15.
vsiminbw	The minimum port bandwidth allocated to this partition in cells/sec. (Multiply by 400 based on 55 bytes per ATM cell to get approximate bits/sec.)
vsimaxbw	The maximum port bandwidth allocated to this partition. This value is used for VSI Qbin bandwidth scaling.

When you add a trunk, the entire bandwidth is allocated to Automatic Routing Management. The **cnfrsrc** command is used to change the allocation to provide resources for a VSI on the BPX switch. The VSIs need to partition the resources between competing controllers: Automatic Routing Management, MPLS, and PNNI. You can have different types of controllers splitting up a partition's assets—for example, Automatic Routing Management and MPLS or Automatic Routing Management and PNNI, but not PNNI and MPLS.

On each interface (port or trunk) of the BXM cards used for label switching, bandwidth and connection resources must be divided between traditional PVC connections and label switching connections. On each interface, space for connections is divided between traditional BPX switch PVC connections and LVCs.

The traditional PVC connections are configured directly on the BPX platform, and label switching connections are set up by the LSC using the VSI. As with all ATM switches, the BPX switch supports up to a specified number of connections. On the BPX switch, the number of connections supported depends on the number of port/trunk cards installed. When configuring the port using the **cnfrsrc** command, the term Logical Connection Number (LCN) is used in place of connection.

Each BXM card supports up to 16,384 connections in total, including PVCs, label switching VSI connections, and connections used for internal signaling. On the BXM, the ports are grouped, and each port group has a certain number of connections allocated to it, as shown in Table 6-6. For label switching, connections are allocated to VSI partitions, which are used to support the LSC.

Table 6-6 *BXM Card Connection Allocation*

BXM Card	Port Group	Port Group Size	LCN Limit Per Port	Average Connections Per Port
8-T3/E3	1	8 ports	16k	2048
12-T3/E3	1	12 ports	16k	1365
4-OC3	2	2 ports	8k	4096
8-OC3	2	4 ports	8k	2048
1-OC12	1	1 port	16k	16384
2-OC12	2	1 port	8k	8192

The various configurations of BXM port resources for label switching are described in this section. The first allocation example uses default allocations. The second allocation example describes more rigorous allocations in which default allocations are not applicable.

Table 6-7 shows the default allocation of LCNs.

Table 6-7 *Default Allocation of LCNs*

Connection Type	Variable	Default Value	cnfrsrc Parameter
AutoRoute LCNs	*a(x)*	256	**maxpvclcns**
Minimum VSI LCNs for partition 1	*n(x)*	512	**minvsilcns**
Maximum VSI LCNs for partition 1	*m(x)*	16384	**maxvsilcns**

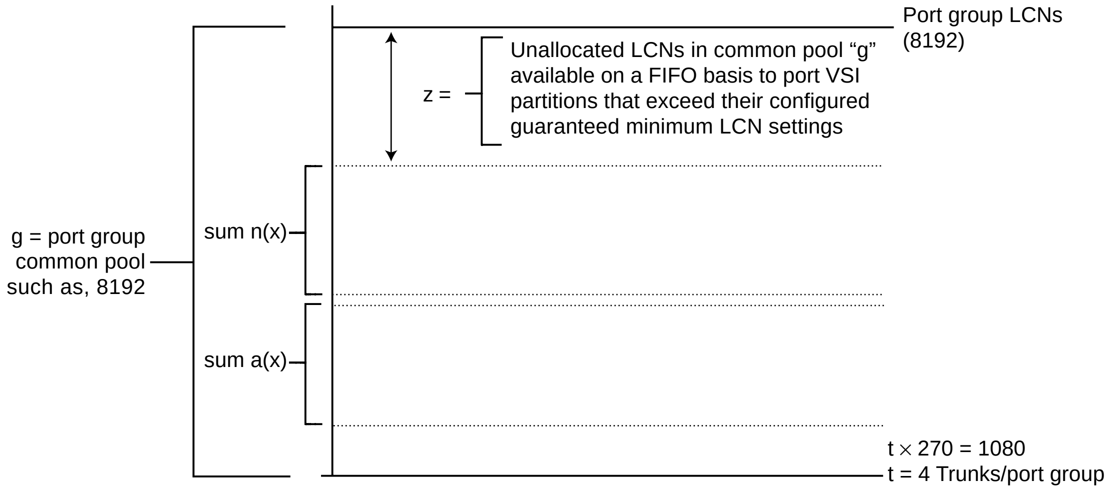
NOTE Different types of BXM cards support different maximums. If you enter a value greater than the allowed maximum, a message is displayed with the allowable maximum. The average *maximum* connections per port are shown in Table 6-7.

Rigorous allocations are possible when default values are not applicable. For example, the LCN allocations for a port group must satisfy the following equation:

$\boldsymbol{g}$ => sum($\boldsymbol{a}$($\boldsymbol{x}$)) + sum($\boldsymbol{n}$($\boldsymbol{x}$)) + $\boldsymbol{t} \times 270$
$\boldsymbol{g}$ = The total number of LCNs available to the port group
$\boldsymbol{a}(\boldsymbol{x})$ = AutoRoute LCNs
$\boldsymbol{n}(\boldsymbol{x})$ = The minimum number of guaranteed VSI LCNs
$\boldsymbol{t}$ = The number of ports in the port group that are configured as AutoRoute trunks

Figure 6-21 shows the relationship of these elements.

Figure 6-21 *Port VSI Partition LCN Allocation Elements*



The 270 value reflects the number of LCNs that are reserved on each AutoRoute trunk for internal purposes. If the port is configured in port rather than trunk mode, t = 0, and $t \times 270 = 0$.

Label switching can operate on a BXM card configured for either trunk (network) or port (service) mode. If a BXM card is configured for port (service) mode, all ports on the card are configured in port (service) mode. If a BXM card is configured for trunk (network) mode, all ports on the card are configured for trunk (network) mode. When the card is configured for *trunk* mode, the trunks reserve some connection bandwidth.

$\boldsymbol{z} = \boldsymbol{g} - \text{sum}(\boldsymbol{a}(\boldsymbol{x})) - \text{sum}(\boldsymbol{n}(\boldsymbol{x})) - \boldsymbol{t} \times 270$
$\boldsymbol{z}$ = The number of unallocated LCNs in the common pool of LCNs available for use by the port VSI partitions
$\boldsymbol{g}$ = The total number of LCNs available to the port group
$\boldsymbol{a}(\boldsymbol{x})$ = AutoRoute LCNs
$\boldsymbol{n}(\boldsymbol{x})$ = The minimum number of guaranteed VSI LCNs
$\boldsymbol{t}$ = The number of ports in the port group that are configured as AutoRoute trunks

NOTE For a BXM card with ports configured in *port* mode, t = 0, and the unallocated LCN equation becomes

$$\boldsymbol{z} = \boldsymbol{g} - \text{sum}(\boldsymbol{a}(\boldsymbol{x})) - \text{sum}(\boldsymbol{n}(\boldsymbol{x}))$$

When a port partition has exhausted its configured guaranteed LCNs (min LCNs), it may draw LCNs for new connections on a FIFO basis from the unallocated LCNs, z, until its maximum number of LCNs, m(x), is reached or the pool, z, is exhausted.

No limit is actually placed on what may be configured for m(x), although m(x) is effectively ignored if it's larger than z + n. The value m(x) is a nonguaranteed maximum value of connection spaces that may be used for a new connection or shared by a number of connections at a given time if a sufficient number of unallocated LCNs are available in z.

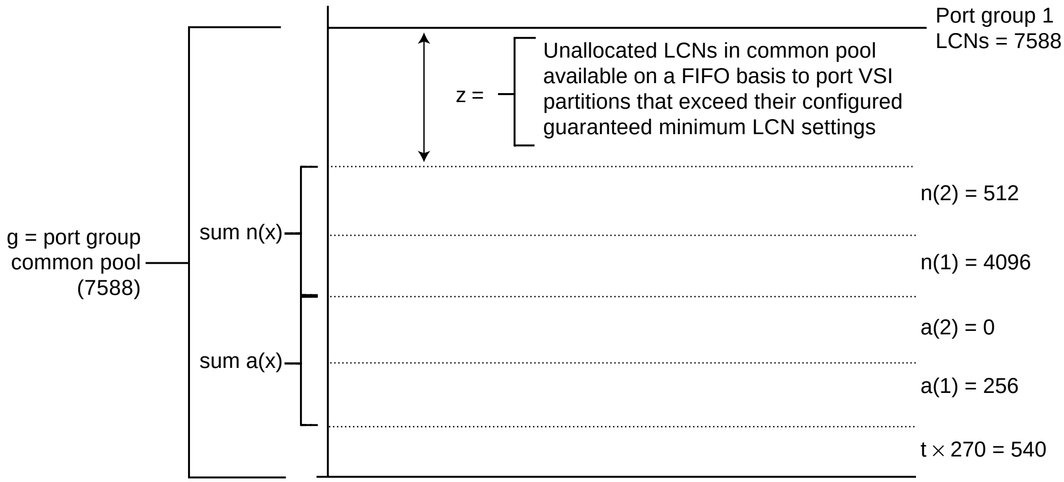
The following is an example of a four-port OC-3 BXM configured in trunk mode. This example is for a four-port OC-3 BXM configured for trunk mode with all ports configured as trunks. Table 6-8 lists the configured connection space (LCN) allocations for each port with respect to a(x), n(x), and m(x). It also shows the unallocated LCN pool, z for each port group, and the total common pool access, g. The total number of LCNs available to the port group is g = 7588 for the four-port OC-3 BXM card. This value is obtained from the **dspcd** command output. Also, the number of trunks per port group is t = 2, which gives you the value of t × 270 = 540.

Table 6-8 *Example of LCN Allocation*

Port(x)	a(x)	n(x)	m(x)	Unallocated LCNs z = g – sum(a(x)) – sum(n(x)) – t × 270	Total LCNS Available to Port VSI Partition = min(z + n(x), m(x))
Port Group 1					
1	256	4096	7588	2184	6280
2	0	512	1024	2184	1024
Sum for Port 1 to 2	256	4608	N/A	N/A	N/A
Port Group 2					
3	0	3200	7588	648	3848
4	0	3200	7588	648	3848
Sum for Port 3 to 4	0	6400	N/A	N/A	N/A

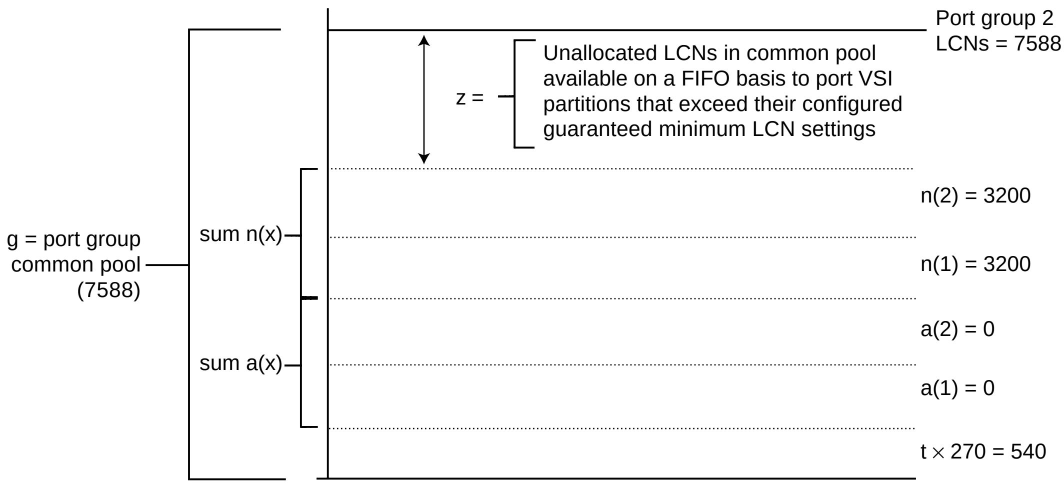
The port groups in the example are ports 1–2 and 3–4, and the maximum number of connection spaces (LCNs) per port group is 7588 for this four-port OC-3 BXM card. The allocations for the port groups are shown in Figure 6-22.

Figure 6-22 *LCN Allocations for Port Group 1*



As shown in Figure 6-22 and Figure 6-23, g is the total number of connection spaces (LCNs) available to port group 1–2 and is equal to 7588 LCNs in this example. To find the number of unallocated LCNs available for use by port partitions that exhaust their assigned number of LCNs, proceed as follows:

Figure 6-23 *LCN Allocations for Port Group 2*



From g, subtract the sum of the AutoRoute connections, a(x), and the sum of minimum guaranteed LCNs, n(x). Also, because the ports in this example are configured in trunk mode, 270 LCNs per port are subtracted from g. Because there are two ports, t equals 2 in the expression $t \times 270$. The unallocated LCNs in the pool (z) are available for use by ports 1–2 that exceed their minimum VSI LCN allocations n(x) for partition 1.

The maximum number of LCNs that a port partition can access on a FIFO basis from the unallocated pool z for new connections can only bring its total allocation up to either z + n(x) or m(x), whichever value is smaller. Also, because z is a shared pool, the value of z will vary as the common pool is accessed by other port partitions in the group.

The values shown in Table 6-8 are obtained as follows:

For ports 1–2:
$\boldsymbol{z} = \boldsymbol{g} - \text{sum}(\boldsymbol{a}(\boldsymbol{x})) - \text{sum}(\boldsymbol{n}(\boldsymbol{x})) - \boldsymbol{t} \times 270$
$\boldsymbol{g} = 7588$
$\text{sum}(\boldsymbol{a}(\boldsymbol{x})) = 256$
$\text{sum}(\boldsymbol{n}(\boldsymbol{x})) = 4608$
$\boldsymbol{t} = 2$
Therefore, $\boldsymbol{z} = 7588 - 256 - 4608 - (2 \times 270)$, which gives you the value of $\boldsymbol{z} = 2184$ unallocated LCNs.

The values shown in Table 6-8 for the port group containing ports 1–2 may be summarized as follows:

Port 1 is guaranteed to be able to support 256 AutoRoute connections (PVCs) and 4096 label VCs (LVCs). It will not support more than 256 PVCs. It might be able to support up to 6280 LVCs, subject to the availability of unallocated LCNs z on a FIFO basis. Also, because z + n(1) of 6280 is less than m(1) of 7588, the maximum number of LVCs that can be supported is 6280.

Port 2 is guaranteed to be able to support 0 AutoRoute connections (PVCs) and 512 label VCs (LVCs). It might be able to support up to 1024 LVCs, subject to the availability of unallocated LCNs z on a FIFO basis. Also, because m(2) of 1024 is less than z + n(2) of 2696, the maximum number of LVCs that can be supported is 1024.

For ports 3–4:
$\boldsymbol{z} = \boldsymbol{g} - \text{sum}(\boldsymbol{a}(\boldsymbol{x})) - \text{sum}(\boldsymbol{n}(\boldsymbol{x})) - \boldsymbol{t} \times 270$
$\boldsymbol{g} = 7588$
$\text{sum}(\boldsymbol{a}(\boldsymbol{x})) = 0$
$\text{sum}(\boldsymbol{n}(\boldsymbol{x})) = 6400$
$\boldsymbol{t} = 2$
Therefore, $\boldsymbol{z} = 7588 - 0 - 6400 - (2 \times 270)$, which gives you the value of $\boldsymbol{z} = 648$ unallocated LCNs.

The values shown in Table 6-8 for the port group containing ports 3–4 may be summarized as follows:

Port 3 is guaranteed to be able to support 0 AutoRoute connections (PVCs) and 3200 Label VCs (LVCs). It might be able to support up to 3848 LVCs, subject to the availability of unallocated LCNs z on a FIFO basis. Because z + n(3) of 3848 is less than m(3) of 7588, the maximum number of LVCs that can be supported is 3848.

Port 4 is guaranteed to be able to support 0 AutoRoute connections (PVCs) and 3200 Label VCs (LVCs). It might be able to support up to 3848 LVCs, subject to the availability of unallocated LCNs z on a FIFO basis. Because z + n(4) of 3848 is less than m(4) of 7588, the maximum number of LVCs that can be supported is 3848.

The following is an example of configuring resources on the VSI partition:

Configure the VSI partition for trunk 5.1 by entering the following command:

```
:cnfrsrc 5.1 256 26000 1 e 4096 7588 2 15 26000 100500
```

NOTE In this example, AutoRoute PVCs a(x) = 256, VSI minimum LCNs n(x) = 4096, and VSI maximum LCNs m(x) = 7588. The m(x) value is derived from the output of the **dspcds** command.

The output of the **dspcds** command provides a value for m(x), the maxvsilcns value. This value needs to be used with the **cnfrsrc** command.

The information in the preceding command line can also be entered individually:

```
cnfrsrc 5.1

PVC LCNs: [256]  [accept default value]
max PVC bandwidth: 26000
partition: 1
y   [to edit VSI parameters]
enabled: e
VSI min LCNs: 4096
VSI max LCNs: 7588
VSI start VPI: 2
VSI end VPI: 15
VSI min b/w: 26000
VSI max b/w: 100500
```

This elicits the following system response:

```
bpx2       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:33 EST

Port/Trunk : 5.1

Maximum PVC LCNS:           256      Maximum PVC Bandwidth:26000

Min Lcn(1) : 0 Min Lcn(2) : 0
Partition 1

Partition State :           Enabled
Minimum VSI LCNS:           4096
```

continues

```
Maximum VSI LCNS:            7588
Start VSI VPI:               2
End VSI VPI :                15
Minimum VSI Bandwidth :      26000    Maximum VSI Bandwidth :  100500



Last Command: cnfrsrc 5.1 256 26000 1 e 4096 7588 2 15 26000 100500


Next Command:
```

This example reserves space on the trunk for 256 AutoRoute PVCs. One VSI partition is supported, and it must be numbered 1. VSI min LCNs = 4096, and VSI max LCNs = 7588. This guarantees that MPLS can set up 4096 LVCs on this link but is allowed to use up to 7588, subject to the availability of LCNs.

VSI starting VPI = 2 and VSI ending VPI = 15. This reserves VPIs in the range of 2–15 for MPLS. Only one VP is really required, but a few more can be reserved to save for future use. AutoRoute uses a VPI range starting at 0, so MPLS should use higher values. It is best to always avoid using VPIs 0 and 1 for MPLS on the BPX 8650. VPIs are locally significant.

A different VPI value could be used for each of the three ports 5.1, 5.2, and 5.3. However, at each end of a trunk, the same VPI must be assigned. VSI min bandwidth = 26000, and VSI maximum = 100500. This guarantees that MPLS can use 26000 cells/sec (about 10 Mbps) on this link but allows it to use up to 100,500 cells/sec (about 40 Mbps) if bandwidth is available. More can be allocated if required.

The maximum PVC bandwidth has been configured with a value of 26000. This guarantees that PVCs can always use up to 26,000 cells per second (about 10 Mbps) on this link.

NOTE Resource partitioning using the **cnfrsrc** command must be performed on all BXM trunks that will carry LVCs across them.

Step 5 Display the configuration of the specified Qbin on a BXM:

`dspqbin` *`slot.port qbin number`*

The **dspqbin** command displays the Qbin resources or CoS buffer resources on a selected trunk, port, or virtual trunk. It displays the Qbin parameters currently configured for an interface and shows whether the Qbin resources have been configured by the user or automatically by a template. It also displays whether the Qbin has EPD enabled or disabled.

The following is an example of displaying the Qbins on the BPX switch.

Display the MPLS queues on the BXM card on port 5.1 for Qbin 10:

```
dsqbin 5.1 10
```

NOTE MPLS CoS uses Qbins 10–14. BPX software release 9.1 only uses Qbin 10.

This elicits the following system response:

```
bpx2       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:41 EST

Qbin Database 5.1 on BXM qbin 10

Qbin State:                 Enabled
Minimum Bandwidth:          0
Qbin discard threshold:     65536
Low CLP/EPD threshold:      95%
High CLP/EPD threshold:     100%
EFCI threshold:             40%




Last Command: dspqbin 5.1 10


Next Command:
```

If preconfigured correctly, the display should show parameters similar to the preceding display.

Step 6 Use the **cnfqbin** command to configure the Qbin CoS buffers parameters on a selected BXM port or trunk as follows:

```
:cnfqbin <slot.port> <Qbin_number> <e/d> <y/n> <Qbin_discard_threshold>
  <Low_EPD_threshold> <High_CLP_Threshold> <EFCI_threshold>
```

Table 6-9 describes the parameters used in the **cnfqbin** command.

Label-switched VC connections are grouped into large buffers called Qbins. Qbins 10–14 are used for label-switched connections. MPLS for VSIs on a BXM card needs the default Qbin value to be adjusted. Qbin 10 is assigned to MPLS. The **cnfqbin** command is used to adjust the threshold for the traffic arriving in Qbin 10 of a given VSI interface as a way of fine-tuning traffic delay. If you use the **cnfqbin** command to set an existing Qbin to disabled, the egress of the connection traffic to the network is disabled. Reenabling the Qbin restores the egress traffic.

When a VSI interface is activated, the default template gets assigned to an interface. The corresponding Qbin template gets copied into the card Qbin data structure for that interface. Assigning new values using the **cnfqbin** command can change this. The Qbin is now user-configured as opposed to template-configured. This information is displayed on the **dspqbin** screen. It indicates whether the values in the Qbin are from the template assigned to the interface or whether the values have been changed to user-defined values.

Table 6-9 **cnfqbin** *Command Parameters*

Parameter	Description
slot.port	Specifies the slot and port number for the BXM.
Qbin_number	Specifies the ID number of the Qbin available for use by the LSC (MPLS controller) for VSI. The range is 0 to 255. 0 is the default. Always use 10 in BPX software releases 9.1 and 9.2.
e/**d**	Enables or disables the Qbin.
y/**n**	You enter **n** not to accept default values so that you can configure the following parameters.
Qbin_discard_threshold	Specifies the threshold in percentage for Qbin discard. The range is 0 to 100.
Low_EPD_threshold	Specifies the threshold in percentage for CLP low. The range is 0 to 100; 80% is the default.
High_CLP_threshold	Specifies a percentage of the Qbin depth. When the threshold is exceeded, the node discards cells with CLP = 1 in the connection until the Qbin level falls below the depth specified by CLP Lo. The range is 0 to 100; 80% is the default.
EFCI_threshold	Explicit Forward Congestion Indication. The percentage of Qbin depth that causes EFCI to be set. The range is 0 to 100; 30% is the default.

The following is an example of how to configure the Qbin values for the BPX:

```
:cnfqbin 5.1 10 e 0 65536 95 100 40
```

This elicits the following system response:

```
bpx2       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:45 EST


Qbin Database 5.1 on BXM qbin 10

Qbin State:                  Enabled
Minimum Bandwidth:           0
Qbin Discard threshold:      65536
Low CLP/EPD threshold:       95%
High CLP/EPD threshold:      100%
```

```
EFCI threshold:              40%



Last Command: cnfqbin 5.1 10 e 0 65536 95 100 40

Next Command:
```

Step 7 Add an ATM link between a BXM card on a BPX node and an MPLS controller such as a series 6400, 7200, or 7500 router as follows:

```
addshelf <slot.port> <device-type> <control_id> <partition_id> <control_vpi>
    <control_vci_start>
```

NOTE The link between the BXM card of the BPX node and the LSC must be free of major alarms before you can add it with the **addshelf** command.

Table 6-10 describes the parameters used in the **addshelf** command.

Table 6-10 **addshelf** *Command Parameters*

Parameter	Description
slot.port	Specifies the BXM slot and port number of the trunk. You can configure the port for either trunk (network) or port (service) mode.
device-type	Specifies a virtual interface to an MPLS controller (LSC) such as a Cisco 6400, 7200, or 7500 series router. The value **vsi** is used for the MPLS Virtual Switch Interface.
control_id	Control IDs must be in the range of 1–32. These must be identical on the LSC and in the **addshelf** command. A control ID of 1 is the default used by the MPLS controller (LSC).
partition_id	Indicates the ID assigned to the VSI partition. A partition ID of 1 is the default used by the MPLS controller (LSC).

continues

Table 6-10 **addshelf** *Command Parameters (Continued)*

Parameter	Description
control_vpi	Starting VPI of the VSI control channels used for communication between the VSI master residing on the SES and VSI slaves residing on the BXM cards. There can be a total of 12 such channels—one for each slave residing on each BXM card. **For a trunk interface with NNI header type:** Valid values for this parameter are 0–4095 **For a trunk interface with UNI header type:** Valid values for this parameter are 0–255. The default value is 0.
control_vci_start	Starting VCI of the VSI control channels. This VCI value is assigned to the first VSI control channel (between the VSI master and the VSI slave residing on the BXM card in slot 1). The last VSI control channel corresponding to communication with the VSI slave on slot 14 uses the VCI value of (<start_vci>+14-1). The valid values are 33–65521. The default value is 40.

The following is an example of adding the MPLS controller to the BPX.

Add an MPLS controller link to port 5.1 of the BXM card on the BPX node by entering the **addshelf** command as follows:

```
:addshelf 5.1 vsi 1 1
```

The first **1** after **vsi** is the VSI controller ID, which must be set the same on both the BPX 8650 and the LSC. The default controller ID on the LSC is 1. The second **1** after **vsi** is the partition ID, which indicates that this is a controller for partition 1.

This elicits the following system response:

```
bpx2       TN   SuperUser    BPX 8620    9.2     Mar. 20 2001 16:50 EST

                        BPX Interface Shelf Information

Trunk    Name      Type      Alarm
5.1      VSI       VSI       OK
```

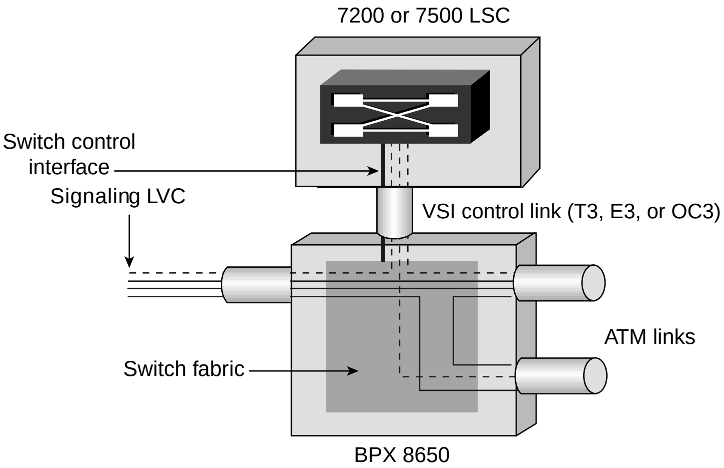
```
This Command: addshelf 5.1 v 1 1

Next Command:
```

Label Switch Controller Configuration

The Label Switch Controller (LSC) is a Label-Switched Router (LSR) that controls the operation of a separate ATM switch. Together, the router and ATM switch function as a single ATM MPLS router (ATM-LSR). As shown in Figure 6-24, a Cisco 7200 or 7500 series router acts as the LSC, and a Cisco BPX 8600 service node acts as the VSI-controlled ATM switch. The LSC controls the ATM switch using the Virtual Switch Interface (VSI), which runs over an ATM link connecting the two.

Figure 6-24 *Label Switch Controller (LSC) and BPX*



Before configuring the LSC for the label switch (MPLS) controlling function, you must perform the initial router configuration if this has not been done. As part of this configuration, you must enable and configure the ATM adapter interface and the extended ATM interface for label switching. On the LSC, the LC-ATM ports on the controlled switch are represented as an IOS interface type called extended label ATM (XmplsATM). You associate XmplsATM interfaces with particular physical interfaces on the controlled switch through the **extended-port** interface configuration command.

LSC as a Label Edge Device

The LSC can function simultaneously as a controller for an ATM switch and as a label edge device. Traffic can be forwarded between a router interface and an LC-ATM interface on the controlled switch as well as between two LC-ATM interfaces on the controlled switch. The LSC can perform the imposition and removal of labels and can serve as the head or tail of a Label-Switched Path (LSP) tunnel. However, when acting as a label edge device, the LSC is limited by the capabilities of its control link with the switch as follows:

- Total throughput between all other router interfaces and switch interfaces is limited by the bandwidth of the control link (that is, OC-3, 155 Mb per second).
- The number of VCs supported on the control link limits label space for LSC-terminated VCs.

The following steps outline the configuration process for the LSC:

Step 1 Enable IP routing on the LSC:

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

Step 2 Enable the Cisco Express Forwarding (CEF) protocol:

```
Router(config)#ip cef switch
```

Step 3 Enable a physical interface link to BPX:

```
Router(config)#interface atm slot/adapter/port
```

Step 4 Remove any IP address assignment from the main ATM interface:

```
Router(config-if)#no ip address
```

Step 5 Enable the router ATM port as an LSC. The controller ID default is 1. Optional values are up to 32 for BPX.

```
Router(config-if)#tag-control-protocol vsi [controller ID]
```

Step 6 Create a logical XmplsATM virtual interface, and bind it to BPX port *SP*:

```
Router(config-if)#interface XtagATM SP
```

Step 7 Bind the extended port XtagATM *SP* to the BPX slave port *Slot.Port:*

```
Router(config-if)#extended-port <ATM slot/adapter/port> <BPX Slot.Port>
```

Step 8 Assign an IP address to XtagATM *SP*:

```
Router(config-if)#ip address ip_address mask
```

Step 9 Enable MPLS for Xmpls interface XtagATM *SP*:

```
Router(config-if)#tag-switching ip
```

NOTE Extended label ATM interfaces differ from ordinary ATM interfaces in that MPLS is configured on the primary interface of an extended label ATM interface, whereas it is configured on an MPLS subinterface of an ordinary ATM interface.

Step 10 Set up the IGP routing process (OSPF or IS-IS) and other routing parameters.

Case Study of an ATM-Based MPLS VPN

Consider the service provider shown in Figure 6-25, with PoPs in Chicago, Seattle, San Diego, Miami, and Washington. The service provider can offer Layer 3 IP VPN services across its MPLS backbone. The service provider offers MPLS VPN services to three customers—A, B, and C—as shown in Table 6-11. MPLS-aware backbone ATM LSRs have replaced the core P routers. The customers are operating a single VPN each. Virtual circuits (PVCs, PVPs, SVCs, or soft PVCs) are not configured on the ATM switches.

The service provider has provisioned the VPNs as shown in Table 6-11.

Table 6-11 *Case Study: VPN Deployment (Point of Presence)*

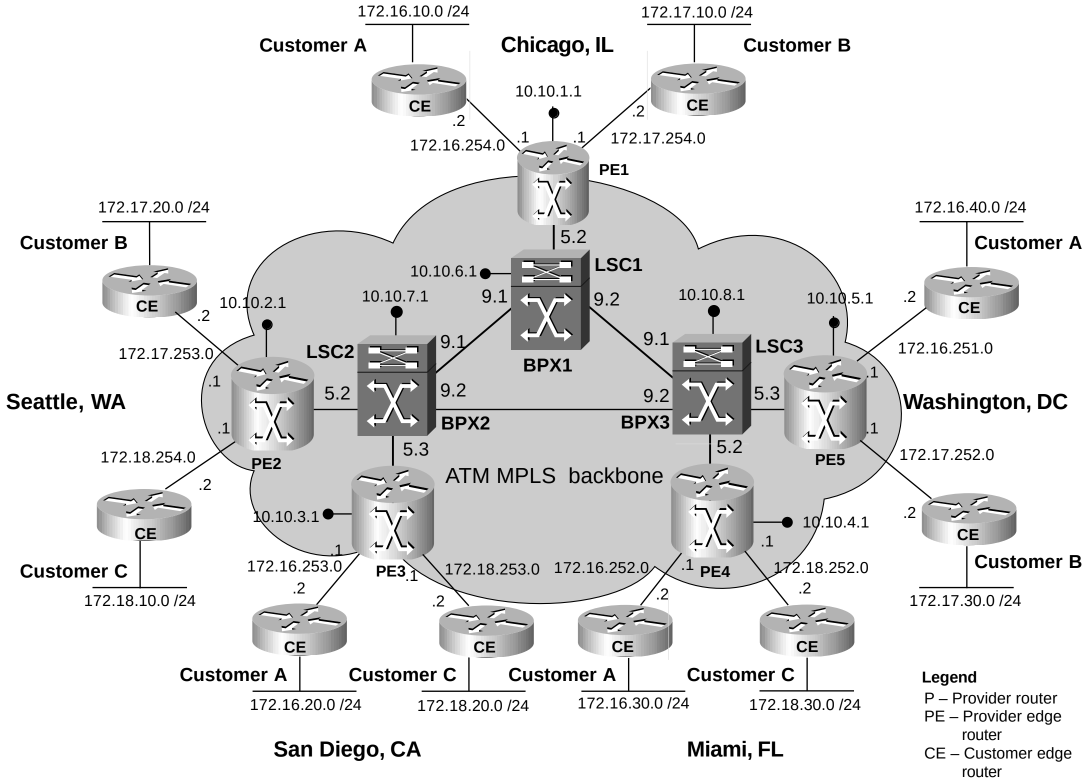
	Customer A	Customer B	Customer C
Chicago, IL	VPN A	VPN B	
Seattle, WA		VPN B	VPN C
San Diego, CA	VPN A		VPN C
Miami, FL	VPN A		VPN C
Washington, DC	VPN A	VPN B	

The customer edge (CE) routers connect to the service provider's edge (PE) routers as shown in Table 6-12.

Table 6-12 *Case Study: PE Deployment*

	Customer A	Customer B	Customer C
Chicago, IL	PE1	PE1	
Seattle, WA		PE2	PE2
San Diego, CA	PE3		PE3
Miami, FL	PE4		PE4
Washington, DC	PE5	PE5	

Figure 6-25 *Case Study: ATM-Based MPLS VPN Configuration*



The core ATM MPLS LSRs are fully meshed in this example. However, this is not necessary. Layer 3 Interior Gateway Routing Protocol (IGRP) running across the backbone normally provides any-to-any connectivity for IBGP peers to communicate with each other, even though the IBGP peers (PE routers) are not directly connected to each other.

The IP addressing scheme used in this case study is explained in Table 6-13.

Table 6-13 *Case Study VPN IP Address Architecture*

	PE Router	CE WAN Subnet	CE LAN Subnet
Customer A			
Chicago	10.10.1.1/32	172.16.254.0/24	172.16.10.0/24
Seattle	No presence	No presence	No presence
San Diego	10.10.3.1/32	172.16.253.0/24	172.16.20.0/24
Miami	10.10.4.1/32	172.16.252.0/24	172.16.30.0/24

Table 6-13 *Case Study VPN IP Address Architecture (Continued)*

	PE Router	CE WAN Subnet	CE LAN Subnet
Washington	10.10.5.1/32	172.16.251.0/24	172/16.40.0/24
Customer B			
Chicago	10.10.1.1/32	172.17.254.0/24	172.17.10.0/24
Seattle	10.10.2.1/32	172.17.253.0/24	172.17.20.0/24
San Diego	No presence	No presence	No presence
Miami	No presence	No presence	No presence
Washington	10.10.5.1/32	172.17.252.0/24	172/17.30.0/24
Customer C			
Chicago	No presence	No presence	No presence
Seattle	10.10.2.1/32	172.18.254.0/24	172.18.10.0/24
San Diego	10.10.3.1/32	172.18.253.0/24	172.18.20.0/24
Miami	10.10.4.1/32	172.18.252.0/24	172.18.30.0/24
Washington	No presence	No presence	No presence

The customer MPLS VPNs could have overlapping IP address architectures. In the MPLS VPN model, each PE router maintains multiple VRF routing tables and a single global routing table. Each VRF routing table corresponds to the VPN routes for each customer.

Provider Edge Configuration

The PE router configurations for the ATM interfaces are shown in the following sections. The ATM LVCs provide MPLS connectivity between the various PE routers and ATM LSRs. The PE routers also run an IGP and multiprotocol IBGP within the cloud. The following configuration examples show the ATM configuration. The RD, VRF, and multiprotocol IBGP configurations are similar to the case study configurations detailed in Chapter 5.

Chicago Configuration

Refer to Figure 6-25 for the Chicago area connectivity and addressing information. Chicago uses the VPI range 2 to 10 to establish LVCs with the ATM LSR (LSC controlled BPX) and remote PE routers. The labels are carried in the VCI field of the ATM header.

Chicago PE Configuration

The following is the Chicago PE configuration for the ATM-based MPLS VPN:

```
!
interface Loopback 0
 ip address 10.10.1.1  255.255.255.255
 no ip directed-broadcast
!
interface atm 1/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no atm ilmi-keepalive
!
interface atm 1/0/0.1 tag-switching
 description ATM link to BPX1
 ip unnumbered Loopback0
 no ip directed-broadcast
 tag-switching atm vpi 2-10
 tag-switching ip
!
```

Seattle Configuration

Refer to Figure 6-25 for the Seattle area connectivity and addressing information. Seattle uses the VPI range 2 to 10 to establish LVCs with the ATM LSR (LSC controlled BPX) and remote PE routers. The labels are carried in the VCI field of the ATM header.

Seattle PE Configuration

The following is the Seattle PE configuration for the ATM-based MPLS VPN:

```
!
interface Loopback 0
 ip address 10.10.2.1  255.255.255.255
 no ip directed-broadcast
!
interface atm 1/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no atm ilmi-keepalive
!
interface atm 1/0/0.1 tag-switching
 description ATM link to BPX2
 ip unnumbered Loopback0
 no ip directed-broadcast
 tag-switching atm vpi 2-10
 tag-switching ip
!
```

San Diego Configuration

Refer to Figure 6-25 for the San Diego area connectivity and addressing information. San Diego uses the VPI range 2 to 10 to establish LVCs with the ATM LSR (LSC controlled BPX) and remote PE routers. The labels are carried in the VCI field of the ATM header.

San Diego PE Configuration

The following is the San Diego PE configuration for the ATM-based MPLS VPN:

```
!
interface Loopback 0
 ip address 10.10.3.1  255.255.255.255
 no ip directed-broadcast
!
interface atm 1/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no atm ilmi-keepalive
!
interface atm 1/0/0.1 tag-switching
 description ATM link to BPX2
 ip unnumbered Loopback0
 no ip directed-broadcast
 tag-switching atm vpi 2-10
 tag-switching ip
!
```

Miami Configuration

Refer to Figure 6-25 for the Miami area connectivity and addressing information. Miami uses the VPI range 2 to 10 to establish LVCs with the ATM LSR (LSC controlled BPX) and remote PE routers. The labels are carried in the VCI field of the ATM header.

Miami PE Configuration

The following is the Miami PE configuration for the ATM-based MPLS VPN:

```
!
interface Loopback 0
 ip address 10.10.4.1  255.255.255.255
 no ip directed-broadcast
!
interface atm 1/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
```

continues

```
 no atm ilmi-keepalive
!
interface atm 1/0/0.1 tag-switching
 description ATM link to BPX3
 ip unnumbered Loopback0
 no ip directed-broadcast
 tag-switching atm vpi 2-10
 tag-switching ip
!
```

Washington Configuration

Refer to Figure 6-25 for the Washington area connectivity and addressing information. Washington uses the VPI range 2 to 10 to establish LVCs with the ATM LSR (LSC controlled BPX) and remote PE routers. The labels are carried in the VCI field of the ATM header.

Washington PE Configuration

The following is the Washington PE configuration for the ATM-based MPLS VPN:

```
!
interface Loopback 0
 ip address 10.10.5.1  255.255.255.255
 no ip directed-broadcast
!
interface atm 1/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no atm ilmi-keepalive
!
interface atm 1/0/0.1 tag-switching
 description ATM link to BPX3
 ip unnumbered Loopback0
 no ip directed-broadcast
 tag-switching atm vpi 2-10
 tag-switching ip
!
```

Label Switch Controller Configuration

The LSC router configurations for the BPX switches are shown in the following sections. The LSC routers run an IGP such as OSPF or IS-IS within the cloud. In this example, the LSCs are running OSPF with a process ID (PID) of 66. The ATM interface to the BPX slave does not have an IP address and is configured for VSI control. This enables the creation and configuration of logical extended MPLS ATM ports.

LSC1 Configuration

Refer to Figure 6-25 for LSC1 connectivity and addressing information. The following is the LSC1 configuration for the ATM-based MPLS VPN:

```
!
hostname lsc1
!
ip cef
!
interface Loopback0
 ip address 10.10.6.1 255.255.255.255
 no ip directed-broadcast
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 tag-control-protocol vsi
 no atm ilmi-keepalive
!
interface XTagATM51
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 5.1
tag-switching IP
!
interface XTagATM91
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 9.1
 tag-switching IP
!
interface XTagATM92
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 9.2
 tag-switching IP
!
router ospf 66
 network 10.10.6.1 0.0.0.0 area 0
!
```

LSC2 Configuration

Refer to Figure 6-25 for LSC2 connectivity and addressing information. The following is the LSC2 configuration for the ATM-based MPLS VPN:

```
!
hostname lsc2
!
ip cef
!
interface Loopback0
 ip address 10.10.7.1 255.255.255.255
 no ip directed-broadcast
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 tag-control-protocol vsi
 no atm ilmi-keepalive
!
interface XTagATM51
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 5.1
 tag-switching IP
!
interface XTagATM52
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 5.2
 tag-switching IP
!
interface XTagATM91
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 9.1
 tag-switching IP
!
interface XTagATM92
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 9.2
 tag-switching IP
!
router ospf 66
 network 10.10.7.1 0.0.0.0 area 0
!
```

LSC3 Configuration

Refer to Figure 6-25 for LSC3 connectivity and addressing information. The following is the LSC3 configuration for the ATM-based MPLS VPN:

```
!
hostname lsc3
!
ip cef
!
interface Loopback0
 ip address 10.10.8.1 255.255.255.255
 no ip directed-broadcast
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 tag-control-protocol vsi
 no atm ilmi-keepalive
!
interface XTagATM51
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 5.1
 tag-switching IP
!
interface XTagATM52
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 5.2
 tag-switching IP
!
interface XTagATM91
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 9.1
 tag-switching IP
!
interface XTagATM92
 IP unnumbered Loopback0
 no IP directed-broadcast
 extended-port ATM1/0 bpx 9.2
 tag-switching IP
!
router ospf 66
 network 10.10.8.1 0.0.0.0 area 0
!
```

BPX Switch Configuration

The partition's BPX switch resources must be distributed between the associated BXM trunk ports. This is set up with the **cnfrsrc** command. The Qbin CoS buffer parameters on the BXM trunks are set up using the **cnfqbin** command. Finally, the LSC is linked to the BPX with the **addshelf** command to allow the router's LSC function to control the MPLS operation of the BPX node.

BPX1 Configuration

Refer to Figure 6-25 for BPX1 connectivity and addressing information. In this case study, BPX1 has a four-port BXM 155-4 in slot 5 with an SMF LM 155-8 back card. A two-port BXM 622-2 is installed in slot 9 with an SMF LM 622-2 back card. Two BCC-4 broadband controller cards are installed in slots 7 and 8 with LM-BCCs installed as back cards. The Alarm Service Module (ASM) card with its associated back card (LM-AMS) is installed in slot 15. Refer to Table 6-14 for the BPX port assignment.

Table 6-14 *BPX Port Assignment*

	BPX1	BPX2	BPX3
BPX1		9.1	9.1
BPX2	9.1		9.2
BPX3	9.2	9.2	
PE1	5.2		
PE2		5.2	
PE3		5.3	
PE4			5.2
PE5			5.3
LSC	5.1	5.1	5.1

The following steps provide the basic configuration commands required by the BPX to achieve operational readiness.

Step 1

Perform the initial BPX configuration as follows:

1. Configure the node name using the **cnfname** command.
2. Configure the time zone using the **cnftmzn** command.
3. Configure the date using the **cnfdate** command.
4. Configure the LAN interface using the **cnflan** command.

5 Configure the IP Relay address using the **cnfnwip** command.

6 Configure the auxiliary or terminal ports to support any necessary external devices such as a local printer or modem using the **cnfprt**, **cnfterm**, and **cnftermfunc** commands.

NOTE Refer to the *Cisco BPX 8600 Series Installation and Configuration Guide* for detailed information on the initial setup procedures.

Step 2

Display the status of all cards, and verify Active status:

```
:dspcds
```

Step 3

Check the logical connection capabilities of the four-port OC3 BXM-155 card in slot 5 using the **dspcd** command:

```
:dspcd 5
```

The following is the output:

```
bpx1       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:10 EST

Detailed Card Display for BXM-155 in slot 5
Status:              Active
Revision:            CD18
Serial Number:       783314
Fab Number:          28-2158-02
Queue Size:          228300
Support:             FST,4 Pts,OC3,Vc
Chnls:16320, PG[1}:7588, PG[2]:7588
PG [1} : 1,2,
PG [2] : 3,4,
Backcard Installed
  Type:         LM-BXM
  Revision:     BA
  Serial Number: 784533
  Supports: 4 pts, OC3, SMF Md


Last Command: dspcd 5



Next Command:
```

This example shows that ports 1 and 2 together have a total of 7588 connections or channels available for use. Ports 1 and 2 form a port group (PG). Similarly, ports 3 and 4 are a port group with a limit of 7588 connections.

Step 4

Enable the BXM trunk interfaces:

```
:uptrk 5.1
:uptrk 5.2
:uptrk 9.1
:uptrk 9.2
```

The following is the output:

```
bpx1       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:20 EST

TRK       Type        Current Line Alarm Status           Other End
9.1       OC3         Clear  -  OK                        bpx1/9.1
9.2       OC3         Clear  -  OK                        bpx2/9.2
5.1       OC3         Clear  -  OK                        VSI(VSI)









Last Command: uptrk 9.2


Next Command:
```

In this example, trunk 5.1 is the link to the LSC controller, and trunks 9.1 and 9.2 are used as the broadband trunks to other BPX nodes on the network. Trunk 5.2 connects to the Chicago PE router and also needs to be *upped*.

Step 5

Use the **cnfrsrc** command to configure partition resources for AutoRoute PVCs and VSI-MPLS. The four-port OC-3 BXM is configured in trunk mode with all ports configured as trunks. Table 6-15 lists the configured connection space (LCN) allocations for each port with respect to a(x), n(x), and m(x). It also shows the unallocated LCN pool, z for each port group, and the total common pool access, g. The total number of LCNs available to the port group is g = 7588 for the four-port OC-3 BXM card. This value is obtained from the **dspcd** command output. Also, the number of trunks per port group is t = 2, which gives you the value of $t \times 270 = 540$.

Table 6-15 *LCN Allocations for Port Groups*

Port(x)	a(x)	n(x)	m(x)	Unallocated LCNs z = g – sum(a(x)) – sum(n(x)) – t x 270	Total LCNS Available to Port VSI Partition = min(z + n(x), m(x))
Port Group 1					
1	256	4096	7588	2184	6280
2	0	512	1024	2184	1024
Sum for Port 1 to 2	256	4608	N/A	N/A	
Port Group 2					
3	0	3200	7588	648	3848
4	0	3200	7588	648	3848
Sum for Port 3 to 4	0	6400	N/A	N/A	

The values shown in Table 6-15 are obtained as follows:

For ports 1–2:
$\boldsymbol{z} = \boldsymbol{g} - \text{sum}(\boldsymbol{a}(\boldsymbol{x})) - \text{sum}(\boldsymbol{n}(\boldsymbol{x})) - \boldsymbol{t} \times 270$
$\boldsymbol{g} = 7588$
$\text{sum}(\boldsymbol{a}(\boldsymbol{x})) = 256$
$\text{sum}(\boldsymbol{n}(\boldsymbol{x})) = 4608$
$\boldsymbol{t} = 2$
Therefore, $\boldsymbol{z} = 7588 - 256 - 4608 - (2 \times 270)$, which gives you the value of $\boldsymbol{z} = 2184$ unallocated LCNs.

For ports 3–4:
$\boldsymbol{z} = \boldsymbol{g} - \text{sum}(\boldsymbol{a}(\boldsymbol{x})) - \text{sum}(\boldsymbol{n}(\boldsymbol{x})) - \boldsymbol{t} \times 270$
$\boldsymbol{g} = 7588$
$\text{sum}(\boldsymbol{a}(\boldsymbol{x})) = 0$
$\text{sum}(\boldsymbol{n}(\boldsymbol{x})) = 6400$
$\boldsymbol{t} = 2$
Therefore, $\boldsymbol{z} = 7588 - 0 - 6400 - (2 \times 270)$, which gives you the value of $\boldsymbol{z} = 648$ unallocated LCNs.

In this case study, AutoRoute PVCs a(x) = 256, VSI minimum LCNs n(x) = 4096, and VSI maximum LCNs m(x) = 7588. The maximum PVC bandwidth = 26000 cells/sec, minimum VSI bandwidth = 26000 cells/sec, and maximum VSI bandwidth = 100500 cells/sec.

The starting VPI is 2, and the ending VPI is 15:

```
:cnfrsrc 5.1 256 26000 1 e 4096 7588 2 15 26000 100500
```

The information in the preceding command line can be entered individually using the BPX command menu prompt:

```
cnfrsrc 5.1

PVC LCNs: [256]  [accept default value]
max PVC bandwidth: 26000
partition: 1
y   [to edit VSI parameters]
enabled: e
VSI min LCNs: 4096
VSI max LCNs: 7588
VSI start VPI: 2
VSI end VPI: 15
VSI min b/w: 26000
VSI max b/w: 100500
```

```
bpx1       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:33 EST

Port/Trunk : 5.1

Maximum PVC LCNS:           256      Maximum PVC Bandwidth:26000

Min Lcn(1) : 0 Min Lcn(2) : 0
Partition 1

Partition State :           Enabled
Minimum VSI LCNS:           4096
Maximum VSI LCNS:           7588
Start VSI VPI:              2
End VSI VPI :               15
Minimum VSI Bandwidth :     26000    Maximum VSI Bandwidth :  100500



Last Command: cnfrsrc 5.1 256 26000 1 e 4096 7588 2 15 26000 100500


Next Command:
```

Repeat this command for all BXM trunks that will carry MPLS LVCs.

Step 6

Display the configuration of Qbin 10 on BPX port 5.1:

```
:dsqbin 5.1 10
```

```
bpx1       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:41 EST

Qbin Database 5.1 on BXM qbin 10

Qbin State:                  Enabled
Minimum Bandwidth:           0
Qbin discard threshold:      65536
Low CLP/EPD threshold:       95%
High CLP/EPD threshold:      100%
EFCI threshold:              40%




Last Command: dspqbin 5.1 10


Next Command:
```

If preconfigured correctly, the display should show parameters similar to the preceding display. If the configuration needs modification, go to Step 5.

Step 7

Configure the Qbin CoS buffer parameters on BXM trunk 5.1:

:cnfqbin 5.1 10 e 0 65536 95 100 40

```
bpx1       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:45 EST


Qbin Database 5.1 on BXM qbin 10

Qbin State:                    Enabled
Minimum Bandwidth:             0
Qbin Discard threshold:        65536
Low CLP/EPD threshold:         95%
High CLP/EPD threshold:        100%
EFCI threshold:                40%



Last Command: cnfqbin 5.1 10 e 0 65536 95 100 40

Next Command:
```

Step 8

Add an MPLS controller link to port 5.1 of the BXM card on the BPX node:

:addshelf 5.1 vsi 1 1

```
bpx1       TN   SuperUser     BPX 8620    9.2     Mar. 20 2001 16:50 EST

                        BPX Interface Shelf Information

Trunk    Name      Type      Alarm
5.1      VSI       VSI       OK







This Command: addshelf 5.1 v 1 1

Next Command:
```

NOTE Repeat this procedure for BPX2 and BPX3. The configurations for BPX2 and BPX3 are similar to the BPX1 configuration.

Summary

Service providers that currently operate ATM or Frame Relay networks over an ATM backbone can leverage the benefits provided by MPLS. They can utilize their existing infrastructure to provide VPN services using MPLS. This is possible if the ATM switches are MPLS-aware. For non-MPLS ATM switches, MPLS can be configured on MPLS-aware routers. The underlying ATM virtual circuits will be considered ATM links.

The operation of MPLS over ATM PVCs results in an overlay model. MPLS is configured on ATM routers, which perform provider (P) and provider edge (PE) router functionality. This model does not realize the full advantages of the underlying ATM QoS. However, for service providers running core ATM networks with non-MPLS ATM switches, MPLS can still be deployed to create VPNs or leverage the advantages of traffic engineering.

The operation of MPLS over native ATM using ATM LSRs as PE routers and WAN switched ATM LSRs as P routers offers service providers the QoS levels guaranteed by ATM core networks and completely alleviates the scalability problem posed by the ATM overlay model.

In an ATM environment, the label switching forwarding function is carried out similarly to normal switching. The label information needed for label switching is carried in the VCI field within one or a small number of VPs.

The control component of MPLS consists of link-state IP routing protocols running in conjunction with MPLS label allocation and maintenance procedures. The control component is responsible for setting up label forwarding paths along IP routes. ATM LSRs use the downstream-on-demand allocating mechanism. Each ATM-LSR maintains a label forwarding information base (LFIB) that contains a list of all IP routes that the ATM LSR uses.

ATM virtual circuits (VCs) established for MPLS are called Label Virtual Circuits (LVCs). A link between two ATM LSRs, or between an ATM Edge LSR and an ATM LSR, is an ordinary ATM link. Because ATM MPLS uses the VCI fields of a few separate VPIs to carry a label, each label on a link corresponds to a different LVC.

The Label Switch Controller (LSC) manages the control and forwarding component of the ATM LSR. Using information provided by the IGP, LDP establishes labels (such as VCs) on links connected to the ATM LSR. When the LSC has established incoming and outgoing labels for the same route in its LFIB, it instructs the switch fabric to set up a connection with the parameters (incoming interface, incoming label VCI, outgoing interface, outgoing label VCI). The LSC can be implemented in integrated switch software or using an external router platform.

A Virtual Switch Interface (VSI) provides a standard interface so that a controller other than the built-in BPX controller card can control a resource in the BPX switch. External controllers such as the LSC are usually implemented as an external 7200 or 7500 router.

The MPLS VPN integration function resides only on the PE routers of the MPLS network. The MPLS cloud is composed of MPLS-aware ATM switches such as the BPX 8650, MGX 8850, 8540 MSR, and LightStream 1010. The BPX 8650 is the recommended platform for industrial-strength core ATM switched networks. The PE feeder nodes can be a combination of MGX 8850s, 8540 MSRs, LS1010, 10000, 12000 GSRs, 7500, or 7200 ATM routers.