CHAPTER 6

Cisco IOS Configurations

This chapter presents the configurations of the four basic TCP/IP architectures for DSL that were described in Chapter 3, "TCP/IP Over ATM." These include three architectures based on RFC 2684 (which made obsolete RFC 1483): Integrated Routing and Bridging (IRB), Routed Bridge Encapsulation (RBE), and Point-to-Point Protocol over ATM (PPPoA). Additionally, both Chapter 3 and this chapter explain Point-to-Point Protocol over Ethernet (PPPoE, RFC 2516). Virtual Private Networks (VPNs) are also described in this chapter, but they are configured beyond the actual DSL network.

In DSL networking, PPPoE is the most popular, although exact market shares of these architectures have not been defined. IRB was the first common architecture in the DSL world and is generally considered obsolescent, although it is still present in legacy networks around the world. RBE is more secure and scalable than IRB while still allowing for the simple, low-cost equipment found in legacy networks. PPPoA is the author's favorite, enabling optimal scalability for provider network expansion and increased end-user sophistication, combined with better security for all parties. All these architectures are encountered in today's DSL networks, and all are covered on the Cisco certification examinations.

This chapter starts with configurations for the Cisco 827 CPE router and then describes the configurations for the Cisco 6000 series of IP/DSL Switches. It finishes with the configurations for the Cisco 6400 Universal Access Concentrator (UAC). You might need to refer to Appendix B, "ATM Overview," to review the details of ATM.

Cisco 827 DSL Configurations

Setting up voice service on the Cisco 827 router is more demanding than just enabling the port(s). Voice activation actually includes two configurations: one for data and one for voice. When you have completed the configuration for the data scenario, you can add voice by configuring the voice ports and dial peers for both analog voice service and Voice over IP (VoIP). The first part of this chapter shows the configuration for data traffic only. Each architecture configuration is explained in its own section. Voice configurations for the 827 are shown in the second half of this chapter.

The data configurations section first lists the configuration commands that are common to all the TCP/IP architectures, such as interface configuration commands. If you understand and can apply those common commands, you can see more quickly the specific commands for each type of architecture. These architecture-specific commands are described in a

different section for each type. Beyond learning the commonalities of all the architectures, you only need to learn the architecture-specific commands for Cisco certification testing, because no DSL service provider would combine all four types simultaneously. Therefore, in the real world, you probably would read about a certain architecture and its configurations, (optimally) try out that configuration or evaluate it for your own network needs, and return to read about alternatives at your convenience.

Interface Commands Common to All DSL Architectures

These configuration commands apply to the ATM and Ethernet interfaces of the Cisco 827, where the ATM interface is also the DSL interface facing the DSL network itself. ATM commands are listed and explained, followed by the Ethernet commands, including enabling basic Network Address Translation (NAT) and Port Address Translation (PAT). After the interface commands, Challenge Handshake Authentication Protocol (CHAP) security commands are listed and explained. Finally in this section of common Cisco 827 commands, basic IP routing commands are listed and explained.

ATM Interface Commands

On the Cisco 827 ADSL router, here are the ATM interface configuration commands you will use most frequently, regardless of the type of TCP/IP architecture:

- **interface ATM0**—Begins interface configuration mode on the main ATM interface, the ADSL port.
- **no shut**—For new routers, the only configuration that is necessary to activate the ADSL port (the ATM interface) is to perform a **no shut** on the ATM interface. If the ADSL trainup is successful, you see the ATM interface line and line protocol become active. If not, use the **debug** command to debug the trainup sequence, as explained throughout other Cisco documentation but not repeated here.
- **no ip address**—Conserves IP addresses by not assigning an address to this main ATM interface.
- **dsl operating-mode auto**—This value might be the default, depending on your version of Cisco IOS Software. Cisco recommends using this command if you are unsure what discrete multitone (DMT) technology the ISP is using. It provides for automatic analysis and synchronization for either of the ITU-defined DMT standards (G.992.1 and G.992.2). If the DSLAM/IP-DSL Switch is using a Cisco 4xDMT ADI-based card with the ANSI-defined DMT2 standard, you should enter the command **dsl operating-mode ansi-dmt**.
- Depending on the version of Cisco IOS Software, you might or might not see the command **no atm ilmi-keepalive** in the configuration, because this is now a default value. When you enable Integrated Local Management Interface (ILMI) keepalives on a dual ATM module, periodic ILMI keepalive messages are sent to the ATM switch on the active. The ATM switch responds to the ILMI keepalives. If the ATM switch fails to respond to

four consecutive keepalives, the dual switches from the active to the backup. The ILMI keepalives feature is useful only if the module is connected to two different ATM switches.

- **bundle-enable**—ATM Permanent Virtual Circuit (PVC) bundles allow you to configure multiple PVCs that have different quality of service (QoS) characteristics between two devices. The purpose is to bind a PVC from the bundle to one or more precedence values. To determine which VC in the bundle will be used to forward specific traffic, the ATM VC bundle management software matches precedence levels between packets and VCs.

  The ATM virtual bundle acts as a single routing link to the destination router. The Operation and Maintenance (OAM) polling mechanism monitors each circuit's integrity individually. For more information, see Appendix B.

- **hold-queue**—Each network interface, including this ATM interface, has a hold-queue limit. This limit is the number of data packets that the interface can store in its hold queue before rejecting new packets. When the interface empties the hold queue by one or more packets, the interface can accept new packets again.

  The command **no hold-queue** with the appropriate keyword restores an interface's default values. The keyword **in** specifies the input queue, and the keyword **out** specifies the output queue. The default input hold queue is 75 packets. The default output hold-queue limit is 100 packets. A queue size has no fixed upper limit. The input hold queue prevents a single interface from flooding the network server with too many input packets. Further input packets are discarded if the interface has too many input packets outstanding in the system. This limit prevents a malfunctioning interface from consuming an excessive amount of memory. For slow links, use a small output hold-queue limit. This approach prevents storing packets at a rate that exceeds the link's transmission capability. For fast links, use a large output hold-queue limit. A fast link might be busy for a short time (and thus require the hold queue), but it can empty the output hold queue quickly when capacity returns.

- **interface ATM0.1 point-to-point**—Configures a virtual ATM subinterface and defines it as a point-to-point type; this command opens subinterface configuration mode. Subinterfaces are handy for differentiating virtual connections by DSL QoS and/or ATM class of service. For instance, priority business DSL subscribers' virtual connections might be configured on one subinterface, and standard residential DSL subscribers' virtual connections would be configured on another subinterface. There might be hundreds, or even thousands, of subinterfaces, so there might be two, three, or more digits after the **ATM0.** prefix. In practice, good DSL network design limits the number of subinterfaces, just as different service-level agreements are limited to a reasonable number that can be easily marketed and configured. Last but not least regarding subinterfaces, an enormous number of subinterfaces would eventually hamper CPU performance.

- **pvc 1/32**—Begins to configure this PVC, as assigned by the DSL network provider, as virtual circuit 32 on virtual path 1 (or any other valid combination) on this interface or subinterface; this command opens PVC configuration mode.

- If NAT is used, the command **ip nat outside** enables external network address translation and establishes this subinterface (the ADSL interface on the 827) as the outside interface.

Ethernet Interface Commands

- On the Cisco 827 ADSL router, the Ethernet interface is the one facing the user network. Here are the Ethernet interface configuration commands you will see and use most frequently. They apply to most types of TCP/IP architectures.
- If NAT is used, the command **ip nat inside** establishes the Ethernet interface as the inside interface for translation direction.
- The command **ip address negotiated** indicates that Internet Protocol Control Protocol (IPCP, RFC 1332) is being used rather than a static IP address. IPCP is an efficient way to obtain the baseline IP address for the Ethernet interface. It is frequently used to enable distribution of IP addresses to the user's LAN devices through Dynamic Host Configuration Protocol (DHCP), where IPCP provides a valid starting address to acquire the range of DHCP addresses.
- Alternatively, a command such as **ip address 192.168.1.1 255.255.255.0** assigns an IP address and subnet to the Ethernet interface.
- The command **mtu** *size* applies to the Maximum Transmission Unit (MTU). Each interface has a default maximum packet size or MTU size. This number generally defaults to the largest size possible for that type of interface. The default MTU size is 1500 on the Ethernet interface. As you will learn later in this chapter, the PPPoE configuration requires changing this Ethernet default. Other architectures and interfaces might require specific MTU configuration as well.

Challenge Handshake Authentication Protocol (CHAP) Commands

The command **ppp authentication chap callin** is one of three commands in typical configurations that together specify CHAP parameters for multiple scenarios. Normally, when two devices use CHAP, each side sends a challenge to which the other side responds, and it is authenticated by the challenger. Each side authenticates the other independently. If you want to operate with non-Cisco routers that do not support authentication by the calling router or device, you *must* use the command **ppp authentication chap callin**.

When you use the **ppp authentication** command with the **callin** keyword, the access server authenticates the remote device only if the remote device initiated the call—that is, if it is the remote device that is calling in. In this case, authentication is specified on incoming (received) calls only.

The second and third CHAP commands configure the central site with a single username and shared secret. These values can be used to authenticate multiple dial-in clients.

For example, consider a situation in which multiple remote devices dial into a central site. Using normal CHAP authentication, the username (which would be the host name) of each remote device and a shared secret must be configured on the central router. In this scenario, the configuration of the central router can get lengthy and cumbersome to manage; however, if the remote devices use a username that is different from their host name, this can be avoided. The username configuration command is **ppp chap hostname** *username1*.

The third CHAP command is **ppp chap password** *password*. It lets the 827 call one or more routers that do not support CHAP as it is defined in up-to-date Cisco IOS Software versions on the 827 router. This configures a common CHAP secret password to use in response to challenges from an unknown peer.

For example, your Cisco 827 might call a rotary of routers (either from another vendor, or running an older version of the Cisco IOS Software) to which a new (that is, unknown) router has been added. The **ppp chap password** command allows you to replace several username and password configuration commands with a single copy of this command on any dialer interface or asynchronous group interface. This command is used for remote CHAP authentication only (when routers authenticate to the peer). It does not affect local CHAP authentication.

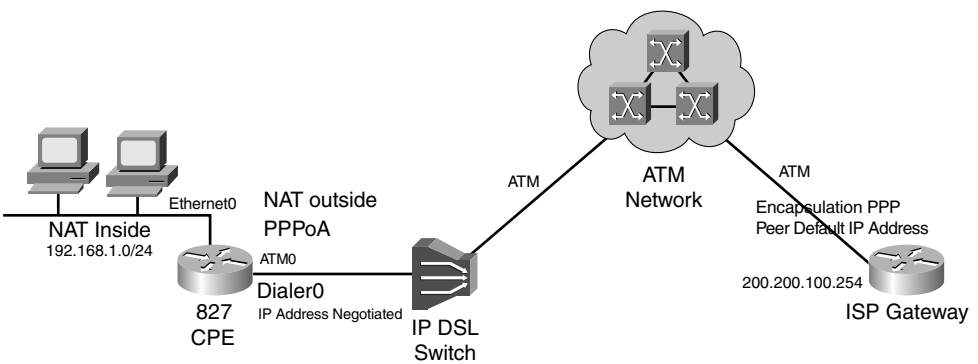
Additional Common Commands

The command **ip route 0.0.0.0 0.0.0.0 192.168.2.254** configures the default gateway, pointing the way through the interface address 192.168.2.254 to the rest of the networked world. The commands discussed previously, starting with the interface configuration commands and including NAT, CHAP, and other common commands, are the basis of the following sections concerning architecture-specific commands.

PPPoA Configuration

Figure 6-1 shows the role of a Cisco 827 router configured with PPPoA.

Figure 6-1 *Cisco 827 Using PPPoA with Dialer Interface, IPCP Negotiation, and NAT Overload*



The following configuration output enables the functionality shown in Figure 6-1. In this configuration, first the Ethernet interface is configured with its IP address and subnet mask. Then the ATM interface (ATM0) is configured, as explained earlier in the "ATM Interface Commands" section, and an ATM subinterface is defined on which PVC 1/32 is configured. The

PPPoA-specific commands begin after the PVC definition. Those commands are explained following this configuration listing:

```
version 12.2
!
interface Ethernet0
    ip address 192.168.1.1 255.255.255.0!
interface ATM0
!(lines omitted)
    interface ATM0.1 point-to-point
      pvc 1/32
         encapsulation aal5mux ppp dialer
         dialer pool-member 1
    interface Dialer1
       ip address negotiated
       ip nat outside
       encapsulation ppp
       dialer pool 1
!
 ip route 0.0.0.0 0.0.0.0 Dialer1
!
hold-queue 224 in

    ppp authentication chap callin
    ppp chap hostname <username1>
    ppp chap password <password1>
!
```

Here are descriptions of the commands:

- **encapsulation aal5mux ppp dialer**—Specifies the encapsulation type for the PVC as aal5mux with Point-to-Point Protocol (PPP) characteristics. It also points back to the dialer interface, which is the ADSL interface on the Cisco 827.
- **dialer pool-member 1**—Specifies that this PVC is a member of dialer pool 1. There can be more than one dialer pool on the physical dialer interface, which is the ADSL interface. Each dialer pool connects to a specific destination subnetwork. That subnetwork usually represents the Cisco IP/DSL Switch.
- **interface Dialer1**—A dialer interface assigns PPP features (such as authentication and IP address assignment method) to a PVC. Dialer interfaces are used when configuring PPP over ATM. This value can be any number in the range 0 through 255. No dialer rotary groups are predefined. This command also opens interface configuration mode for the ADSL interface, which is designated the dialer interface. A dialer interface is not a physical interface, but a logical grouping of physical interfaces with the same configuration. Dialer interfaces allow you to apply a single interface configuration, such as an access control list, to one or more physical interfaces. This standardizes the configuration on those interfaces and reduces configuration labor.
- **ip address negotiated**—Rather than assigning an IP address to this dialer interface, IPCP is used to obtain an IP address as needed upon startup.
- **ip nat outside**—This dialer interface (the ADSL interface, also the ATM interface) is the interface that translates external IP addresses through NAT.
- **encapsulation ppp**—Associates the PPP encapsulation with this dialer (ADSL) interface.

- **dialer pool 1**—Specifies on the dialer (ADSL) interface which dialer pool number to use to connect to a specific destination subnetwork. The number can be any number from 1 to 255. There is no default. This number must match the number used in the **dialer pool-member 1** under the physical interface.
- **ip route 0.0.0.0 0.0.0.0 Dialer1**—Configures the default route available through this dialer (ADSL) interface.

Now suppose that the provider has assigned a single, external IP address. In addition to the configuration shown previously, you can proceed to add the commands for NAT overload, also called PAT, as shown in the following:

```
interface Ethernet0
  ip address 192.168.1.1 255.255.255.0
  ip nat inside
interface Dialer1
    ip address negotiated
    encapsulation ppp
    dialer pool 1
!   (lines omitted)
    ip nat outside
!

ip route 0.0.0.0 0.0.0.0 Dialer0

access-list 1 permit 192.168.1.0 0.0.0.255
!
ip nat inside source list 1 interface Dialer1 overload
```

- **ip nat outside** establishes the interface Dialer1, the ADSL interface, as the NAT outside interface.
- **access-list 1 permit 192.168.1.0 0.0.0.255** defines a standard access list, matching the number 1 of the **list 1** in the next command, permitting addresses that need translation.
- **ip nat inside source list 1 interface Dialer1 overload** enables PAT using the Dialer1 IP address as the inside global address for source addresses that match **access-list 1** in the previous command.

The final set of modifications to the original, simple PPPoA configuration lets the 827 deliver IP addresses to the user workstations and other client devices using DHCP:

```
interface Ethernet0
  ip address 192.168.1.1 255.255.255.0
  ip nat inside
interface Dialer0
     ip address negotiated
     encapsulation ppp
     dialer pool 1
!    (lines omitted)
     ip nat outside

!(lines omitted)

ip dhcp pool POOL-DHCP
       network 192.168.1.0 255.255.255.0
       domain-name cisco.com
       dns-server 192.168.3.1
       default-router 192.168.1.1
```

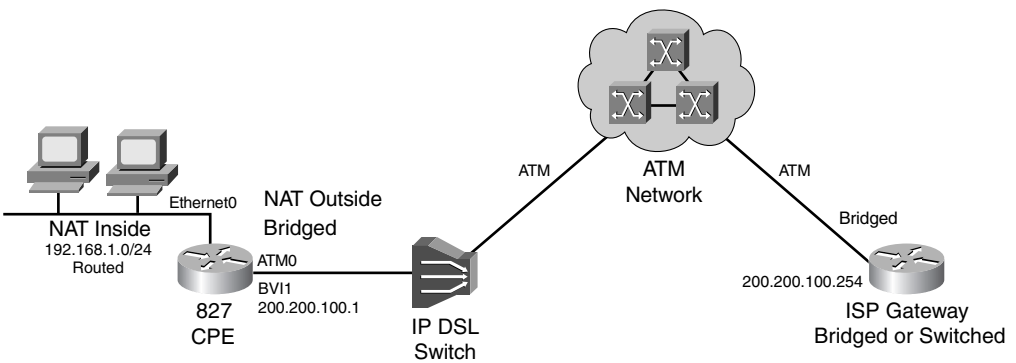
In this configuration, the command **ip dhcp pool POOL-DHCP** defines the range of IP addresses that can be assigned to the DHCP clients. Note that rather than specify a beginning address and an ending address in the pool, subnetting is used for the entire subnet 192.168.1.0.

The command **domain-name cisco.com** simply configures the domain name—in this case, with the variable **cisco.com**. The command **dns-server 192.168.3.1** defines the DNS server with that IP address. The command **default-router 192.168.1.1** designates the 827 router as the default router and specifies an IP address.

RFC 2684 Bridging (Formerly RFC 1483 Bridging)

Figure 6-2 shows the role of the 827 device in a bridged network.

Figure 6-2 *Cisco 827 Using RFC 2684 Bridging (IRB) with NAT*



RFC 2684 bridging is possible on the Cisco 827, as shown in this section's configuration and explanations. The 827 is capable of more-sophisticated configurations than bridging. Therefore, its use in bridging would probably be as an interim solution for a DSL provider who is migrating to a more-sophisticated model, such as PPPoE or PPPoA. When the 827 router is in bridge mode, the Ethernet and ATM interfaces can have the same IP address. That would be the MAC address of interface ethernet0, displayed with the Cisco IOS Software command **show interface ethernet 0**. Following is the configuration for RFC 2684 bridging:

```
bridge irb
!
interface Ethernet0
 ip address 192.168.1.1 255.255.255.0
!
interface ATM0
 (lines omitted)
 bundle-enable
!
interface ATM0.1 point-to-point
 pvc 1/32
```

```
!
  encapsulation aal5snap
!
 bridge-group 1
!
interface BVI1
!
 ip address 192.168.2.1 255.255.255.0
 no ip directed-broadcast
!
ip route 0.0.0.0 0.0.0.0 192.168.2.254
!
bridge 1 protocol ieee
 bridge 1 route ip
```

Here are the explanations for the pertinent commands in this configuration:

- **bridge irb** enables IRB as opposed to the Cisco 827 default of routing.
- **encapsulation aal5snap** specifies the encapsulation type for this particular PVC. **aal5snap** means ATM Adaptation Layer 5 Subnetwork Access Protocol, the standard for RFC 2684-based bridging.
- **bridge-group 1** specifies the bridge-group number to which the point-to-point ATM0.1 interface belongs. This corresponds to the **bridge protocol** command.
- **interface BVI1** defines a bridge group virtual interface (BVI) on the existing routed interface from the WAN bridge group to the nonbridged LAN interface—that is, from the DSL provider to the internal LAN. Each bridge group can have only one corresponding BVI. When you configure the BVI and enable routing on it, as shown in the preceding command, packets that come in on a routed interface destined for a host on a segment that is in a bridge group are transferred from Layer 3 routing to Layer 2 bridging across this interface.
- **ip address 192.168.2.1 255.255.255.0** assigns an IP address and subnet to the BVI within interface configuration mode on BVI 1.
- **bridge 1 protocol ieee** enables the Cisco 827 router's bridging engine, identifies the bridging process with a bridge-group number, and specifies the particular spanning tree algorithm used to avoid bridging loops. All routers on the network that expect to bridge between each other need to share the same bridge-group number. The selected spanning-tree protocol must also be consistent on each router. In this example, the IEEE spanning-tree algorithm is specified (as it invariably is) rather than the DEC option.
- **bridge 1 route ip** enables IP routing to and from bridge group 1.

RFC 2684 Bridging with PAT

Beyond the simplest configuration for RFC 2684 bridging, you can also configure the Cisco 827 to include NAT and PAT. As described previously for the basic IRB configuration, the Cisco 827 might be used as a bridging device in a legacy DSL network situation. In this scenario, only one

registered IP address might be assigned by the service provider, indicating the benefit of PAT, which translates one-to-many IP addresses. This is shown in the following configuration:

```
bridge irb
! (lines omitted)
interface Ethernet0
   ip address 192.168.1.1 255.255.255.0
   ip nat inside
! (lines omitted)
interface BVI1
 ip address 192.168.2.1 255.255.255.0
   ip nat outside
!
ip nat inside source list 1 interface BVI1 overload
!
! (lines omitted)
access-list 1 permit 192.168.1.0 0.0.0.255
```

The explanations for the new commands, enabling PAT, are as follows:

- Together, the following three lines in sequence open interface configuration mode for the primary Ethernet interface, assign an IP address to it, and establish the Ethernet interface as the inside (customer-facing) interface:

```
interface Ethernet0
  ip address 192.168.1.1 255.255.255.0
  ip nat inside
```

- Together, the following three lines in sequence open interface configuration mode for the BVI, assign an IP address to it, and establish the BVI interface as the outside (DSL network-facing) interface:

```
interface BVI1
  ip address 192.168.2.1 255.255.255.0
    ip nat outside
```

- **ip nat inside source list 1 interface BVI1 overload** enables dynamic translation of addresses permitted by the access list to the address specified in the BVI.
- **access-list 1 permit 192.168.1.0 0.0.0.255** defines a standard access list permitting addresses that need translation with PAT.

RFC 2684 Bridging with NAT

Continuing the scenario of the Cisco 827 in bridging mode, suppose that the DSL provider has assigned a block of IP addresses instead of a single address as defined previously. For multiple IP addresses, NAT can be used to translate many-to-many IP addresses. Like PAT, NAT provides privacy for the internal, client-side IP addresses on the internal LAN. These additions are shown in the following configuration snippet. The pertinent commands are followed by the explanations for the commands that amplify the original RFC 2684 bridging configuration.

```
bridge irb
! (lines omitted)
interface Ethernet0
   ip address 192.168.1.1 255.255.255.0
   ip nat inside
```

```
!
interface ATM0
! (lines omitted)

!
interface ATM0.1 point-to-point
   pvc 1/35
!
ip address 192.168.2.1 255.255.255.0
   ip nat outside
!
ip nat pool POOL-A 192.168.2.2 192.168.2.10 netmask 255.255.255.0
!
ip nat inside source list 1 pool POOL-A overload
!
access-list 1 permit 192.168.1.0 0.0.0.255
```

Following are the explanations that add to the original RFC 2684 bridging configuration to permit NAT:

- Together, the following three BVI-related commands open configuration mode for this BVI, assign an IP address and subnet to it, and then establish the BVI as the outside (DSL network-facing) interface:

```
interface BVI1
   ip address 192.168.2.1 255.255.255.0
   ip nat outside
```

- **ip nat pool POOL-A 192.168.2.2 192.168.2.10 netmask 255.255.255.0** creates a pool named POOL-A of global (registered) IP addresses for NAT, from addresses 192.168.2.2 through 192.168.2.10, with appropriate subnet masks.
- **ip nat inside source list 1 pool POOL-A overload** enables dynamic translation of addresses permitted by the access list numbered 1 to one of the addresses specified in the pool.
- **access-list 1 permit 192.168.1.0 0.0.0.255** defines a standard access list permitting addresses that need translation.

RBE Configuration

RBE is set up on the Cisco 827 router exactly like its more-basic cousin, IRB. Remember from the descriptions of the TCP/IP architectures in Chapter 3 that RBE is a more-efficient implementation of RFC 2684 bridging than simple IRB. For RBE, the difference consists of only a few different commands configured on the IP DSL Switch, Cisco 6400, or a comparable Layer 3 device. In the DSL environment, RBE actually routes IP over bridged RFC 2684 Ethernet traffic from a stub-bridged LAN. Bridged IP packets received on an ATM interface configured in route-bridged mode are routed through the IP header. These interfaces on the IP/DSL Switch or a similarly-capable device offer increased performance and flexibility over IRB. In addition, RBE reduces the security risk associated with IRB by reducing the size of the unsecured network. By using a single virtual circuit (VC) allocated to a subnet (which could be as small as a single IP address), ATM RBE limits the "trust environment" to a single customer premises using IP addresses in the subnet.

PPPoE Configuration

The PPPoE feature allows a PPP session to be initiated on a simple bridging Ethernet-connected client. The session is transported over the ATM link via encapsulated Ethernet-bridged frames. With PPPoE, multiple PCs can be installed behind the Cisco 827. As with PPPoA, traffic from these clients can be filtered, NAT can be run, and so on.

In Cisco IOS Software Release 12.1(3)XG, a PPPoE client feature was introduced for the Cisco 827, allowing the PPPoE client functionality to be moved to the router.

When the Cisco 827 itself is the PPPoE client, the 827 PPPoE configuration uses Virtual Private Dialup Networking (VPDN). The following configurations show this scenario. This requires Cisco IOS Software version 12.1(3)XG or later. Multiple PCs can be installed behind the Cisco 827. The Cisco 827 performs routing in this case, and it can be set up as the DHCP server and can perform NAT/PAT, as in the PPPoA configuration example.

The following is the configuration of the PPPoE client on the 827. It is followed by detailed explanations of the important commands.

```
vpdn enable
!
vpdn-group pppoe
       request-dialin
       protocol pppoe
!
interface Ethernet0
       ip address 10.92.1.182 255.255.255.0
       ip nat inside

!

interface ATM0
!(lines omitted)
!

interface ATM0.1 point-to-point
        pvc 1/32
              pppoe-client dial-pool-number 1

interface Dialer1
       ip address negotiated
       mtu 1492
       ip nat outside
       encapsulation ppp

        dialer pool 1
       ppp authentication chap callin
       ppp chap hostname client1
       ppp chap password 7 020508520E081B70


ip nat inside source list 1 interface Dialer1 overload
ip classless
ip route 0.0.0.0 0.0.0.0 dialer1
no ip http server
!
access-list 1 permit any
```

The command **vpdn enable** turns on VPDN on the router. PPPoE runs on top of AAL5SNAP, but the **encap aal5snap** command is not used on the interface, as it is with simple RFC 2684 bridging.

The command **vpdn-group pppoe** associates a VPDN group with a VPDN profile. Then the command **request-dialin** means that this endpoint, the 827, represents the PPPoE client requesting to establish a PPPoE session with the aggregation unit (6400 UAC).

The third command in this group, **protocol pppoe**, characterizes this VPDN group as a PPPoE protocol type.

After the ATM subinterface command and the PVC definition, the command **pppoe-client dial-pool-number 1** indicates that the PPPoE client is linked to a dialer interface upon which a virtual-access interface is cloned.

The command **mtu 1492** applies to the MTU, as introduced at the start of this chapter in the Ethernet interface commands. The default MTU size is 1500 on the dialer interface. Because the PPPoE header is 8 bytes long, you must lower the maximum MTU size to 1492 bytes (1500 minus 8) to allow the 8 bytes of PPPoE header overhead.

If an 82X router terminates the PPPoE traffic, a computer connected to the Ethernet interface might have problems accessing web sites, because the default MTU configured on the PC(s) might be too high. The default MTU on the computer is 1460. The solution is to let the router automatically reduce the value of the Maximum Segment Size (MSS) inside the Transmission Control Protocol (TCP) Synchronization (SYN) packets transmitted by the PCs by entering the following command on the router's Ethernet interface:

```
ip adjust-mss mss
```

where *mss* must be 1452 or less to fix the computer-82X PPPoE MTU problem. This command works only if NAT is configured.

NOTE It is vital to note that all devices on a physical medium must have the same protocol MTU to operate.

The command **encapsulation ppp** is self-explanatory. The PPP encapsulation provides for multiplexing of different network-layer protocols simultaneously over the same link. The PPP encapsulation has been carefully designed to retain compatibility with the most commonly used supporting hardware.

To support high-speed implementations, the default encapsulation uses only simple fields, only one of which needs to be examined for demultiplexing. The default header and information fields fall on 32-bit boundaries, and the trailer may be padded to an arbitrary boundary. The packets to be encapsulated consist of a protocol field, followed by the payload and optionally followed by padding.

The command **ip nat inside source list 1 interface Dialer1 overload** enables dynamic translation of addresses permitted by the access list to the address specified in the Dialer interface.

The command **ip route 0.0.0.0 0.0.0.0 Dialer1** configures the default route. For NAT you can overload on the Dialer1 interface and add a default route out to the Dialer1 interface, because the dialer's IP address can change.

The command **access-list 1 permit any** permits any source address to be translated.

When a packet leaves via this Ethernet interface for a multicast routing table entry, the packet is process level-switched for this interface but may be fast-switched for other interfaces in the outgoing interface list. When fast switching is enabled (such as in unicast routing), debug messages are not logged. If you want to log debug messages, you must disable fast switching, as was done in this configuration.

VPN Configuration

A VPN carries private data over a public network, extending remote access to users over a shared infrastructure. VPNs maintain the same security and management policies as a private network. They are the most cost-effective method of establishing a point-to-point connection between remote users and a central network.

A benefit of access VPNs is the way they delegate responsibilities for the network. The customer outsources the responsibility for the IT infrastructure to an Internet service provider (ISP). The ISP maintains the modem pools into which the remote users dial, the access servers, and the internetworking expertise. The customer is then responsible only for authenticating users and maintaining its network. VPNs also allow separate and autonomous protocol domains to share common access infrastructure, including modems and access servers, enabling service provider resource sharing among clients.

Instead of connecting directly to the network by using the expensive Public Switched Telephone Network (PSTN), access VPN users only need to use the PSTN to connect to the ISP local point of presence (POP), constituting a VPDN. The ISP then uses the Internet to forward users from the POP to the customer network.

Forwarding a user call over the Internet provides dramatic cost savings for the customer. Access VPNs use Layer 2 tunneling technologies such as Layer 2 Tunneling Protocol (L2TP) to create virtual point-to-point connections between users and the customer network. These tunneling technologies provide the same direct connectivity as the expensive PSTN by using the Internet. This means that users anywhere in the world have the same connectivity as they would have at customer headquarters.

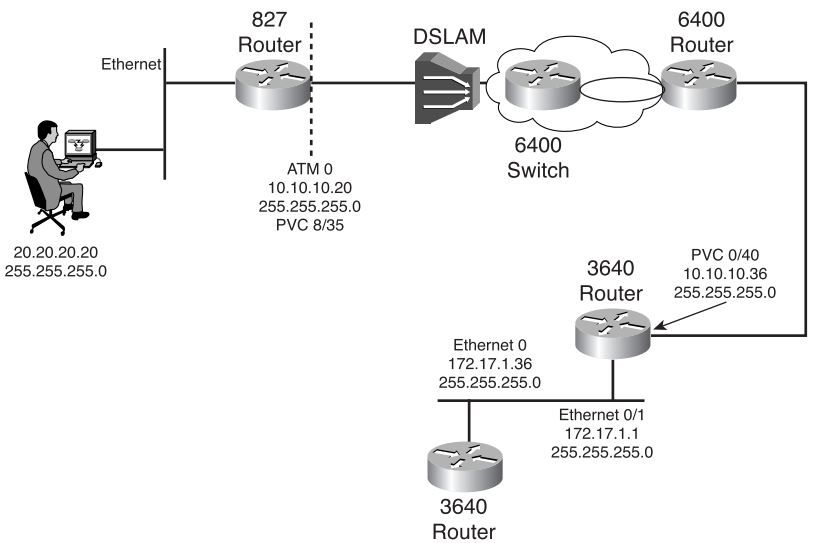
Using L2TP tunneling, an ISP or other access service can create a virtual tunnel to link a customer with remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP POP exchanges PPP messages with the remote users and communicates through L2TP requests and responses with the customer tunnel server to set up tunnels.

L2TP passes routed protocol packets through the virtual tunnel between endpoints of a point-to-point connection. Frames from the remote users are accepted by the ISP POP, stripped of any linked framing or transparency bytes, encapsulated in L2TP, and forwarded over the appropriate tunnel. The customer tunnel server accepts these L2TP frames, strips the Layer 2 encapsulation, and processes the incoming frames for the appropriate interface.

Cisco 827 Configurations for Voice Service

Figure 6-3 shows a theoretical network with VoIP configured on the 827-4V. The voice capability of the 827-4V starts with the same configurations as for the 827 without voice ports.

Figure 6-3 *Cisco 827-4V with Basic VoIP*



As discussed earlier, setting up voice on the router actually includes two configurations—one for data and one for voice. This section leads you through the data configurations first and then the voice configurations. These consist of the following steps:

Step 1 Configure the data network:

- Configure the class map, route map (optional), and policy map
- Configure the Ethernet interface
- Configure the ATM interface
- Configure Enhanced IGRP

Step 2 Configure the voice network:

- Configure the POTS (plain old telephone service, or basic telephone service) dial peers
- Configure VoIP dial peers for H.323 signaling

The following sections discuss the details of each of these steps.

Configuring the Data Network

The data network depends on a specific traffic classification policy that allocates bandwidth and interface access by priority according to traffic type. Traffic types include digitized voice service, standard IP-type packets, and various traffic types whose priorities fall between high-priority voice traffic and standard-priority routine data traffic. Defining the traffic policy determines how many types of packets (number of classes) are to be differentiated from one another. Packets are matched to each other, forming classes based on protocols, access control lists, and input interfaces. These three are the usual match criteria. Before starting the configurations themselves, you must understand the class options.

To characterize a class, you can specify the queue limit for that class, which is the maximum number of packets allowed to accumulate in the class's queue. Packets belonging to a class are subject to the bandwidth and queue limits that characterize the class.

Queuing Considerations

After a queue has reached its configured queue limit, enqueuing additional packets to the traffic class causes either *tail drop* or weighted random early detection (WRED) drop to take effect, depending on how the service policy is configured.

NOTE Tail drop is a means of avoiding congestion that treats all traffic equally and does not differentiate between classes of service. Queues fill during periods of congestion. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full.

WRED drops packets selectively based on IP precedence. Packets with a higher IP precedence are less likely to be dropped than packets with a lower precedence. Thus, higher-priority traffic is delivered with a higher probability than lower-priority traffic in the default scenario. However, packets with a lower IP precedence are less likely to be dropped than packets with a higher IP precedence in certain WRED configurations.

Flow classification is standard weighted fair queuing (WFQ) treatment. That is, packets with the same source IP address, destination IP address, source TCP or User Datagram Protocol (UDP) port, or destination TCP or UDP port are classified as belonging to the same flow. WFQ allocates an equal share of bandwidth to each flow. Flow-based WFQ is also called fair queuing because all flows are equally weighted. WFQ can speed up handling for high-precedence traffic at congestion points.

There are two levels of queuing: ATM queues and IOS queues. Class-based weighted fair queuing (CBWFQ) is applied to IOS queues. It extends the standard WFQ functionality in support of user-defined traffic classes. For CBWFQ, you define traffic classes based on match criteria including protocols, access control lists (ACLs), and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that class.

Each class has a weight derived from the bandwidth you assigned to the class when you configured it. The weight specified for the class becomes the weight of each packet that meets the class's match criteria. Packets that arrive at the output interface are classified according to the match criteria filters you define, and then each one is assigned the appropriate weight.

After a packet's weight is assigned, the packet is enqueued in the appropriate class queue. CBWFQ uses the weights assigned to the queued packets to ensure that the class queue is serviced fairly.

Tail drop is used for CBWFQ traffic classes unless you explicitly configure a service policy to use WRED to drop packets as a means of avoiding congestion. Note that if you use WRED packet drop instead of tail drop for one or more traffic classes making up a service policy, you must ensure that WRED is not configured for the interface to which you attach that service policy.

If a default class is configured, all unclassified traffic is treated as belonging to the default class. If no default class is configured, by default the traffic that does not match any of the configured classes is flow-classified and given best-effort treatment. As soon as a packet is classified, all the standard mechanisms that can be used to differentiate service among the classes apply.

A first-in, first-out (FIFO) IOS queue is automatically created when a PVC is created. If you use CBWFQ to create classes and attach them to a PVC, a queue is created for each class.

CBWFQ ensures that queues have sufficient bandwidth and that traffic gets predictable service. Low-volume traffic streams are preferred; high-volume traffic streams share the remaining capacity, obtaining equal or proportional bandwidth. Bandwidth for the policy map may not exceed 75 percent of the total PVC bandwidth.

Resource Reservation Protocol (RSVP) can be used in conjunction with CBWFQ. When both RSVP and CBWFQ are configured for an interface, RSVP and CBWFQ act independently, exhibiting the same behavior that they would if each were running alone. RSVP continues to work as it does when CBWFQ is not present, even in regard to bandwidth availability assessment and allocation.

RSVP works well on PPP, HDLC, and similar serial-line interfaces. It does not work well on multiaccess LANs. RSVP can be equated to a dynamic access list for packet flows. You should configure RSVP to ensure QoS if the following conditions describe your network:

- Small-scale voice network implementation
- Links slower than 2 Mbps
- Links with high utilization
- You need the best possible voice quality

Configuring the Traffic Policy: Traffic Precedence, Class Maps, Policy Maps

After considering the queuing and prioritization techniques explained previously, you can begin designing the traffic policy configuration for the Cisco 827. Starting with the classification of traffic types, there are two principal aspects to configuring the traffic policy:

- Class maps, which define the traffic classes
- Policy maps, which associate the policies (traffic classes) with interfaces

Traffic Precedence

The first step in building the class map is to configure the access list, including setting an IP precedence, to associate with the class map. As per RFC 791, there are eight classes of service, although later RFCs provide more independence in proprietary precedence definitions. Cisco Systems endeavors to conform to RFC 791, meaning that you can partition traffic in up to six classes of service using IP precedence; two others are reserved for internal network use. The network queuing technologies then use this IP precedence definition to expedite traffic handling. The original, RFC 791-defined classes are as follows, in order from lowest priority to highest priority:

- **Traffic class (TC) = 0: Routine (uncharacterized traffic)**—If otherwise undefined, these packets are assigned the lowest-priority value and are delivered based on the available bandwidth. Non-TCP/IP traffic is assigned to this traffic class. There is a very high possibility of packet drop in the event of congestion.
- **TC = 1: Priority**—There is a high possibility of packet drop if congestion is encountered.
- **TC = 2: Immediate**—There is a medium possibility of packet drop in the event of congestion.
- **TC = 3: Flash**—There is a low possibility of packet drop in the event of congestion.
- **TC = 4: Flash-override**—There is a very low possibility of packet drop compared to the lower-priority classes.
- **TC = 5: Critical**—Cisco recommends this class for voice traffic.
- **TCs 6 and 7**—For Internet and network traffic, respectively. Examples include signaling protocols.

IP precedence is not a queuing method, but it gives other queuing methods (WFQ, WRED) the capability to prioritize based on the packet's IP precedence. The network gives priority (or some type of expedited handling) to the marked traffic through the application of WFQ or WRED at points downstream in the network.

The mapping from keywords such as **routine** and **priority** to a precedence value is useful in only some instances. In other words, the use of the precedence bit is evolving. Bear in mind that IP precedences can be used to establish classes of service that do not necessarily correspond numerically to better or worse handling in the network.

The **ip precedence** command is used by the Cisco 827 router to differentiate voice traffic from data traffic and to assign voice packets a higher priority. Here is an example of this command applied on a Cisco 827 DSL router for voice service:

```
Router (config)#access-list 101 permit ip any any precedence 5
```

This command builds an extended access list numbered 101, which permits IP traffic from any source to any destination and then assigns this permitted traffic the IP precedence of 5 for voice packets. You can also use the plain-text priority designations themselves rather than the numbers:

```
Router (config)#access-list 101 permit ip any any precedence critical
```

Features such as policy-based routing and committed access rate (CAR) can be used to set precedence based on extended access lists.

Class Maps

The next step in building the voice configuration on the 827 is to configure the class map called *voice*. The command **class-map voice** defines a traffic class and the match criteria that are used to identify traffic as belonging to that class. **match** statements can include criteria such as an ACL, an IP precedence value, or a Differentiated Services Code Point (DSCP) value.

The DSCP is a designation by the Internet Engineering Task Force (IETF) of the 6 most significant bits of the 1-byte IP Type of Service (ToS) field. The match criteria are defined with one **match** statement entered in class-map configuration mode.

Here is an example of what might be in the class-map VOICE definition:

```
Router(config)#class-map VOICE
Router(config-cmap)#match ip rtp 16384 16383
Router(config-cmap)#match access-group 101
```

In the first command, IP Real-Time Protocol (RTP) ports 16384 and 16383 (possible values through 32767) are configured as the match criteria. In the second command, access list 101 is matched with the class map. That is, the class map is now associated with IP packets whose IP precedence is 5, the recommended voice packet precedence you defined earlier with the command **access-list 101 permit ip any any precedence 5**.

Policy Maps

Policy maps group one or more class maps, up to 64 different classes of service, for later association with a particular interface. The policy map thereby confers all its referenced class map values onto the interface. In the commands discussed in the preceding section, you first defined an access list and assigned permitted traffic the priority of 5. Then you referenced that access list, 101, in defining the class map called VOICE. The class map is also related to traffic only on ports 16383 and 16384. Because that is a relatively narrow definition, the policy map should also contain a class for other types of traffic, although this is more of a security consideration than a voice configuration consideration.

The commands in the following listing define the policy map named MYPOLICY, which associates the class maps VOICE and class-default (the default for unreferenced traffic). As an example of one option, the command **Priority 176** guarantees 176 kbps of bandwidth for the priority traffic. Beyond the guaranteed bandwidth, the priority traffic is dropped in the event of congestion to ensure that the nonpriority traffic is not starved. Another option is to define the guaranteed bandwidth as a percentage of the overall interface bandwidth. A third option is to specify a maximum burst size in bytes to be tolerated before dropping traffic.

```
Policy-map MYPOLICY
     Class VOICE
          Priority 176
     Class class-default
```

You have finished configuring the class map and policy map, the first steps in configuring the data network, leading to final voice service configuration. Now you will adjust the interface configurations. You learned earlier about configuring the Ethernet interface for the TCP/IP architectures common to DSL. The ATM interface has some details beyond those earlier, basic ATM interface commands for the Cisco 827. These new ATM configuration commands provide for voice service and draw on the concepts already explained in this chapter, as well as some more specific details.

ATM Interface Configuration

The next step is to configure the ATM interface. Here are the commands to do so:

```
interface ATM0
      mtu 300
!
     ip address 192.168.2.1 255.255.255.0
     no atm ilmi-keepalive
     pvc 1/32
          service-policy out MYPOLICY
          vbr-rt 640 640 10
          encapsulation aal5snap
```

The first step in configuring the ATM interface is to adjust the size of the MTU. If you are configuring PPP, either PPPoA or PPPoE, you should decrease the ATM interface's MTU size so that large data packets are fragmented. It is recommended that you use 300 for the MTU size because it is larger than the size of the voice packets generated by the different codecs.

With multiclass multilink PPP interleaving, large packets can be multilink-encapsulated and fragmented into smaller packets to satisfy the delay requirements of real-time voice traffic. Small real-time packets, which are not multilink-encapsulated, are transmitted between fragments of the large packets. The interleaving feature also provides a special transmit queue for the smaller, delay-sensitive packets, enabling them to be transmitted earlier than other flows. Interleaving provides the delay bounds for delay-sensitive voice packets on a slow link that is used for other best-effort traffic.

Next, the policy map named MYPOLICY that you created earlier is associated with the PVC in the outbound direction.

You can then specify the PVC's service class. In this case, the command **vbr-rt 640 640 10** defines Variable Bit Rate Real Time with a peak cell rate (PCR) of 640 kbps, a sustained cell rate (SCR) of 640 kbps, and a Maximum Burst Rate (MBR) of ten cells in a single burst. You should configure the SCR to be at least four times the particular codec's bandwidth when the four voice ports are used. For example, if you have a 640 kbps upstream PVC running codec G.729, you could configure the PVC with an SCR of 176.

Finally in configuring the ATM interface, this PVC is assigned the encapsulation of aal5snap.

Enhanced IGRP Configuration

Continuing with configuring the data aspects of the Cisco 827, you should enter router configuration mode and enable Enhanced IGRP. The autonomous-system number identifies the route to other Enhanced IGRP routers and is used to tag the Enhanced IGRP information.

Specify the network number for each directly connected network.

The following configuration shows the Enhanced IGRP routing protocol enabled in IP networks 10.0.0.0 and 172.17.0.0. The Enhanced IGRP autonomous system number is assigned as 100:

```
Config#router eigrp 100
Config-router#network 10.0.0.0
Config-router#network 172.17.0.0
```

You can now proceed to the voice-specific configurations.

Voice Network Configuration

Following is the voice-specific configuration:

```
!(lines omitted)
voice-port 1
      timing hookflash-in 0
voice-port 2
      timing hookflash-in 0
voice-port 3
      timing hookflash-in 0
voice-port 4
      timing hookflash-in 0
!(lines omitted)
scheduler max-task-time 5000
```

```
dial-peer voice 1 pots
    destination-pattern 1001
    port 1
!
dial-peer voice 10 voip
    destination-pattern 2...
    session target ipv4:192.168.2.8
!
    codec g711ulaw (optional)
```

The commands **voice-port** *X* and **timing hookflash-in 0** turn off any hookflash indications that the gateway could generate on an FXO interface. Currently the Cisco 827-4V does not support hookflash indications, although that support is probably pending, because it is already available on other Cisco platforms with H.323 Version 2 Phase 2. On an analog phone, hookflash means pressing the switchhook for a moment (about one-half second) to produce a special stutter dial tone. This engages supplemental services, such as call waiting.

The command **scheduler max-task-time 5000** is not specific to the 827. It is how long, in milliseconds, a specific process is handled by the CPU before it reports debugging information—in this case, 5 seconds.

Dial Peer Configuration

Dial peers enable outgoing calls from a particular telephony device. All the voice technologies use dial peers to define the characteristics associated with a call leg. A call leg is a discrete segment of a call connection that lies between two points in the connection.

Bear in mind that these terms are defined from the router perspective. An inbound call leg means that an incoming call comes to the router. An outbound call leg means that an outgoing call is placed from the router.

Two kinds of dial peers can be configured for each voice port: POTS and VoIP:

- POTS associates a physical voice port with a local telephone device. The **destination-pattern** command defines the telephone number associated with the POTS dial peer. The **port** command associates the POTS dial peer with a specific logical dial interface, normally the voice port connecting the 827-4V to the POTS network. You can expand an extension number into a particular destination pattern with the command **num-exp**. You can use the **show num-exp** command to verify that you have mapped the telephone numbers correctly.
- The VoIP dial peer also associates a telephone number with an IP address. The key configuration commands are the same **destination-pattern** and **session target** commands that are used with the POTS dial peer. The former command is the same as with the POTS dial peer, defining a telephone number. The **session target** command specifies a destination IP address for the VoIP dial peer. This command must be used in conjunction with the **destination-pattern** command. Going further than the POTS dial peer, you can use VoIP dial peers to define characteristics such as IP precedence, QoS parameters, and codecs. For instance, you can optionally specify a different codec than the default codec of g.729.

For both POTS and VoIP, after you have configured dial peers and assigned destination patterns to them, you can use the **show dialplan** *number* command to see how a telephone number maps to a dial peer.

When a router receives a voice call, it selects an outbound dial peer by comparing the called number (the full E.164 telephone number) in the call information with the number configured as the destination pattern for the POTS peer. The router then strips the left-justified numbers corresponding to the destination pattern matching the called number. On POTS dial peers, the only digits that are sent to the other end are the ones specified with the wildcard character (.) with the command **destination-pattern** *string*. The POTS dial peer command **prefix** *string* can be used to include a dial-out prefix that the system enters automatically instead of having people dial it. If you have configured a prefix, it is put in front of the remaining numbers, creating a dial string, which the router then dials. If all the numbers in the destination pattern are stripped, the user receives (depending on the attached equipment) a dial tone.

For example, suppose there is a voice call whose E.164 called number is 1 (310) 555-2222. If you configure a destination pattern of 1310555 and a prefix of 9, the router strips 1310555 from the E.164 telephone number, leaving the extension number of 2222. It then appends the prefix 9 to the front of the remaining numbers so that the actual numbers dialed are 9, 2222. The comma in this example means that the router pauses for 1 second between dialing the 9 and dialing the first 2 to allow for a secondary dial tone.

Earlier, you defined a class called VOICE with the **class-map** command. You matched the access control list 101 with this class of service using the **match access-group** command. You also defined a policy map with the **policy-map** command. Those commands are shown here, along with some new options:

```
class-map VOICE
     match access-group 101
!
policy-map POLICY
      class VOICE
           priority 480

pvc 1/32
          service-policy out POLICY
          vbr-rt 640 640 10
          encapsulation aal5snap
!
bundle-enable
!
dial-peer voice 1 pots
     destination-pattern 1001
     port 1

dial-peer voice 10 voip
     destination-pattern 2...
!
     session target ipv4:192.168.2.8
!
     ip precedence 5
!
access-list 101 permit ip any any precedence critical
```

The command **priority 480** defines the priority of the VOICE class in terms of guaranteed bandwidth. In this case, if there is congestion on the network, even this priority traffic is dropped when it exceeds 480 kbps. This ensures that the nonpriority traffic is not starved.

The command **service-policy out POLICY** attaches the policy map to this particular PVC, 1/32. The policy map could also be attached to an interface, either inbound or outbound.

The command **vbr-rt 640 640 10** defines Variable Bit Rate Real Time (suitable for this voice traffic) with a PCR of 640 kbps, an SCR of 640 kbps, and an MBR of ten cells in a single burst.

This PVC's encapsulation type is aal5snap, suitable for either RFC 2684 bridging (IRB or RBE) or PPPoE.

The command **bundle-enable** creates ATM PVC bundles, about which you learned earlier. The command **dial-peer voice 1 voip** simply uses one of two options; this was explained earlier as well.

Two values at a minimum are required to configure a VoIP peer: the associated destination telephone number and a destination IP address. The command **destination-pattern** defines the destination telephone number. This specification is then associated with port 1. In this configuration example, the last digits in the VoIP dial peer's destination pattern are replaced with wildcards.

The **ip precedence** command defines precedence 5, preferred for voice.

Last, the **access-list** command clears the way through access list 101 for any IP traffic and sets that traffic's precedence as critical.

Returning to check the steps in configuring the 827-4V for data and voice, you are now ready for the last step, which completes the process by configuring the VoIP dial peers for H.323 signaling.

VoIP Dial Peers for H.323 Signaling

The H.323 signaling protocol was explained in Chapter 4, "Cisco DSL Products." Following is the configuration:

```
interface ATM0
     h323-gateway voip interface

h323-gateway voip id GATEKEEPER ipaddr 192.168.1.2 1719
!

h323-gateway voip h323-id GATEWAY
!
!(lines omitted:  define telephone number, specify port number)
!
dial-peer voice 10 voip
  destination-pattern +.T
session target ras

gateway
```

The first H.323-related command in this configuration, **h323-gateway voip interface**, identifies the interface ATM0 as the gateway interface.

The command **h323-gateway voip id GATEKEEPER ipaddr 192.168.1.2 1719** defines the name and location (IP address) of the gatekeeper for this gateway.

The next command, **h323-gateway voip h323-id GATEWAY**, defines the gateway's H.323 name, identifying this gateway to its associated gatekeeper.

The command **destination-pattern +.T** introduces a new value. The plus sign (+) indicates an E.164 standard number, and the T indicates the default route.

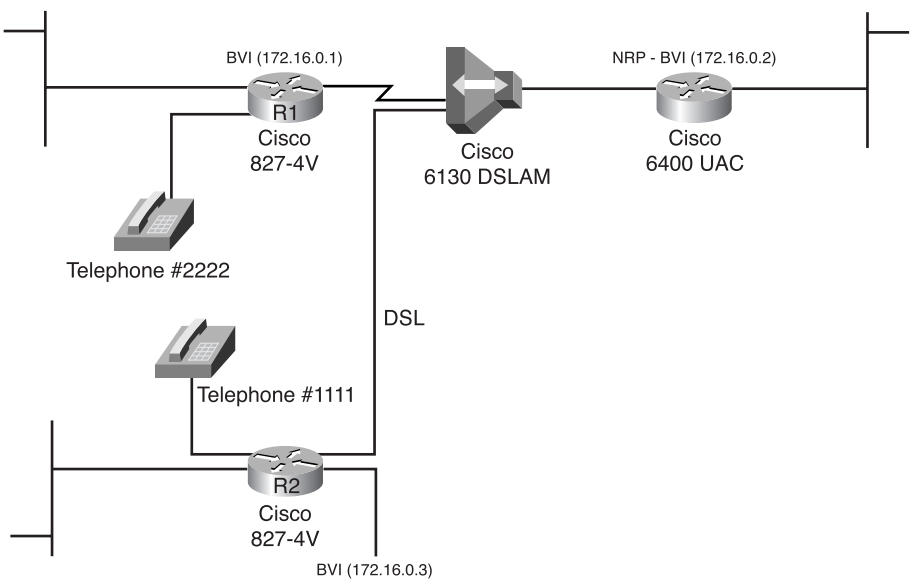
The command **session target ras** specifies the destination as having Registration, Admission, and Status (RAS) functionality, providing gateway-to-gatekeeper functionality.

Finally, the one-word command **gateway** defines this 827-4V as the H.323 gateway device.

Completing the 827-4V Configuration

You can now complete the configuration of the Cisco 827-4V. Figure 6-4 shows the use of the Cisco 827-4V configured for RFC 2684 bridging (IRB) and VoIP.

Figure 6-4 *Cisco 827-4V Using IRB for VoIP*



When the Cisco 827 replaces existing bridged DSL modems, the IRB configuration is a typical starting point. Although the Cisco 827-4V supports voice service in all the other previously discussed architectures in which the network scheme would be different, IRB is shown here simply as an example. The new commands are explained after the configuration listing.

Here is the configuration required for the 827-4V in this legacy replacement scenario:

```
version 12.1
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname R1
!
bridge irb
!
interface Ethernet0
no ip mroute-cache
!
interface ATM0
no ip address
no atm ilmi-keepalive
     pvc 1/150
         encapsulation aal5snap
bundle-enable
  bridge-group 1
  hold-queue 224 in
!
interface BVI1
      ip address 172.16.0.1 255.255.0.0
!
ip classless
ip route 0.0.0.0 0.0.0.0 BVI1
no ip http server
!
bridge 1 protocol ieee
      bridge 1 route ip
!

voice-port 1
     timing hookflash-in 0
!
voice-port 2
     timing hookflash-in 0
!
voice-port 3
     timing hookflash-in 0
!
voice-port 4
     timing hookflash-in 0
!
dial-peer voice 1 pots
     destination-pattern 2222
     port 1
!
dial-peer voice 2 voip
     destination-pattern 1111
     session target ipv4:172.16.0.3
!
```

The command **bridge irb** enables RFC 2684 bridging (IRB) for the whole Cisco 827-4V router.

The command **ip mroute-cache** configures IP multicast fast switching. In this Cisco 827-4V, it is disabled on the Ethernet interface. When packets arrive on this Ethernet interface for a multicast routing table entry with mroute caching disabled, those packets are sent at process level for all interfaces in the outgoing interface list.

When packets leave via this Ethernet interface for a multicast routing table entry, the packet is process level-switched for this interface, but it may be fast-switched for other interfaces in the outgoing interface list.

The command **bridge-group 1** specifies the bridge group to which the interface belongs.

The command **BVI1** creates a BVI and assigns a corresponding bridge group number to that BVI, as discussed earlier in this chapter.

The command **bridge 1 protocol ieee** is an IOS standard specifying Spanning Tree Protocol for bridge group 1.

The command **bridge 1 route ip** lets the BVI accept and route routable packets received from its corresponding bridge group. You must enter this command for each routed protocol (such as IPX) that you want the BVI to route from its corresponding bridge group to other routed interfaces.

You are now done configuring the Cisco 827-4V for IRB and VoIP. Look again at Figure 6-3 and consider the more-complex explanation of the use of your new, complete configuration. This figure shows a voice scenario configuration using the Cisco 827-4V router in an H.323 signaling environment. Traffic is routed through the 827 router and then is switched onto the ATM interface. The 827 router is connected through the ATM interface through one PVC, and it is associated with a QoS policy called mypolicy. Data traffic coming from the Ethernet must have an IP precedence below 5 (critical) to distinguish it from voice traffic.

NAT (represented by the dashed line at the edge of the 827 routers) signifies two addressing domains and the inside source address. The source list defines how the packet travels through the network.

Now that you have configured the 827-4V as a voice-carrying router, you need to configure the PVC endpoint. An interesting option is to use multiple PVCs. Multiple PVCs, separating voice and data, create an easily expandable, easily traced configuration, although this is not required for minimal functionality. Here is that configuration:

```
!(lines omitted)

interface ATM0.1 point-to-point
     ip address 192.168.2.1 255.255.255.0

     pvc 1/35
          protocol ip 192.168.2.2 broadcast


          vbr-rt 424 424 5
          encapsulation aal5snap
!
```

```
interface ATM0.2 point-to-point


     pvc 1/36 (data PVC)
               protocol ip 192.168.3.2 broadcast
               encapsulation aal5snap

dial-peer voice 1 pots
destination-pattern 1001
port 1


dial-peer voice 10 voip
destination-pattern 2...


session target ipv4:192.168.2.8
```

In this configuration, the first PVC is for voice service. It is configured on a point-to-point subinterface, ATM0.1. This IP PVC has a point-to-point IP address of 192.168.2.1, with a subnet mask of 255.255.255.0.

Then the service class of Variable Bit Rate (VBR) is set, with parameters of PCR of 424 kbps, SCR of 424 kbps, and MBR of five cells in a single burst. This voice PVC's encapsulation is aal5snap.encapsulation aal5snap.

Troubleshooting the Cisco 827

The first thing you should do when troubleshooting the Cisco 827 is check the front panel CD LED. If the light is not on, no ADSL carrier is detected. Usually this is a physical problem, probably due to a bad cable or a problem with an ADSL line or WAN service. You can try replacing the cable, but you will probably have to contact the DSL provider.

Another simple solution to 827 problems might lie with the ATM interface. To verify its status, you can enter the command **show interface ATM 0**. If the status is **up/down**, the Cisco 827 sees the ADSL carrier but cannot train up with the central office (CO)/exchange IP-DSL Switch properly. In this case, check the cable itself. The Cisco 827 uses pins 3 and 4 of the ADSL cable. The ADSL cable must be 10BASE-T Category 5 unshielded twisted-pair (UTP) cable. Using regular telephone cable can introduce line errors. Contact your ADSL line or service provider to determine if there is a problem.

If the Cisco 827 does not establish a satisfactory DSL circuit to the CO/exchange ADSL port, you can observe the process of DSL synchronization as the 827 trains up to help isolate the problem. Following are the normal stages of the synchronization so that you can verify which steps are occurring correctly to aid your troubleshooting. To observe the training process, you can enter the command **debug atm events** and observe the outputs, shown in the following:

Normal activation state changes are

STOP	In shutdown state
INIT	Initialization

DLOAD_1	Initialized and downloading first image
DLOAD_2	Downloading second image
DO_OPEN	Requesting activation with CO

In DO_OPEN state, look for the modem state for the progress information:

Modem state = 0x0	Modem down
Modem state = 0x8	Modem waiting to hear from CO
Modem state = 0x10	Modem heard from CO and now is training
Modem state = 0x20	Activation completed and link is up
SHOWTIME	Activation succeeded

Central Office/Exchange Equipment

This section discusses the configurations for the two pieces of CO/exchange gear in the DSL network—the Cisco 6000 series IP DSL Switch and the Cisco 6400 UAC.

IP DSL Switch/NI-2

The NI-2 Cisco IOS Software is based on the Catalyst 8500 (also known as the LightStream 1010) code, with added extensions for DSL. The IOS code supports ATM services such as ATM QoS and traffic management, PVCs and soft-PVCs, and ILMI and OAM cell support.

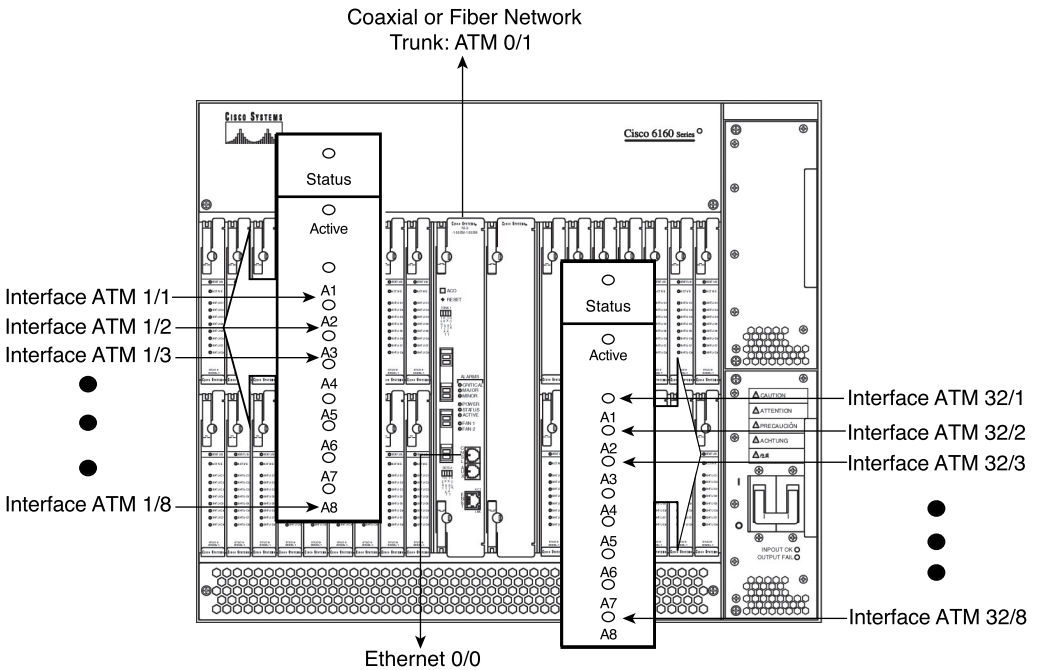
After the initial configuration, you can either continue to manage the DSLAM/IP DSL Switch with the Cisco IOS Software command-line interface (CLI) or use the Cisco DSL Manager (CDM) graphical user interface (GUI) software. Even if you choose to manage the device with CDM (this is covered in Chapter 7, "Cisco DSL Manager (CDM)"), you must still use a few Cisco IOS Software commands to prepare the device for CDM.

Basic Setup Commands

To start setting up the NI-2, you can access the CLI by connecting a terminal directly to the console port on the NI-2 card or by Telnetting to the management port if an IP address has been configured on the NI-2 Ethernet interface. Because the NI-2 runs a specific version of Cisco IOS Software, many commands are unique to the DSL service environment. The software image itself is designated with the letters DA. The D indicates that this is a Cisco IOS Software release for the DSL environment, and the A indicates that it is specifically for the NI-2. (As you will learn in the section "Aggregator/Concentrator: Cisco 6400," other specific software versions also start with D for DSL but have a different second letter.)

The System Configuration dialog and the CLI use the interface numbering scheme shown in Figure 6-5.

Figure 6-5 *Cisco 6160 Interfaces*



Interfaces whose names begin with ATM0 (ATM0/0, ATM0/1, and so on) are NI-2 card WAN interfaces. ATM0/0 is the ATM switch's interface with the processor. There is no need to configure ATM0/0 unless you plan to use in-band management. ATM0/1 is the trunk port, also called the upstream or network trunk. ATM0/2 and ATM0/3 (if present) are subtending interfaces.

Interfaces whose names begin with ATM1, including all the higher numbers after the ATM-designator, are xTU-C (line card) interfaces. The range of line cards is ATM1 through ATM6 on the compact 6015, ATM1 through ATM30 on the international 6260, or ATM1 through ATM32 on the North American 6160. ATM10 and ATM11, which would indicate the NI-2 slots, are omitted in the software configuration on the two larger IP DSL Switches, although these are labeled as slots 10 and 11 on the chassis itself.

Ethernet0/0 is the interface for the LAN that connects the Cisco IP DSL Switch to its management system.

For individual line card ports, the number before the slash indicates the slot number. The number after the slash indicates the interface or port number. For example, ATM6/4 is port 4 in slot 6.

The following steps show you how to configure the NI-2 for basic operations:

Step 1 Specify software codes. You should specify the source and filename of the configuration that will be used to boot the NI-2. Use the global configuration mode command **boot system flash:** *path filename* to specify the boot file. You can use the file system in Flash memory to copy files and troubleshoot configuration problems. Use the privileged EXEC command **dir flash:** to display the contents of Flash memory. This process might take a few minutes while Flash memory is being initialized.

```
DSLAM#dir flash:
Directory of flash:/
2  -rw- 4883532  Jan 01 2000  00:02:46  NI-2-dsl-mz.120-5.DA1
3  -rw- 5396464  Jan 02 2000  02:04:08  NI-2-dsl-mx.flexi.aluia
4  -rw-  345324  Jan 02 2000  02:06:13  flexd.bin.aluia
15990784 bytes total (413568 bytes free)
```

Step 2 Set the IP address on the primary Ethernet interface. You should set the IP address, and the subnet mask if you won't use the default, on the interface Eth 0/0. Use the interface configuration mode command **ip address** *XXX.XXX.XXX.XXX yyy.yyy.yyy.yyy,* where *X* represents the IP address and *Y* represents the subnet mask.

Step 3 Set the passwords. At a minimum, you will set three passwords:

A. Console password—Set the console password using the standard Cisco IOS Software configuration command **password**.

B. Telnet password—At the privileged EXEC mode prompt, enter the following commands:

```
NI-2#config terminal
NI-2 (config)#line vty 0 4
NI-2 (config- line)#password <TELNET PASSWORD>
NI-2 (config- line)#login
```

Exit, and then test by Telnetting into the device.

C. Privileged EXEC password, also erroneously but popularly called the enable password—You set the privileged EXEC password using the standard Cisco IOS Software configuration command **enable secret** or **enable password**.

Step 4 Set the time, date, and host name. Although it isn't absolutely required, you can set several system parameters as part of the initial system configuration.

Step 5 Set the clocking options. Each port has a transmit clock and derives its receive clock from the receive data. You can configure transmit clocking for each port in one of the following ways:

— **Network derived**—Transmit clocking is derived from the highest-priority configured source, either from the internal clock (the default) or the public network.

— **Loop-timed**—Transmit clocking is derived from the receive clock source. The IP DSL Switch receives derived clocking, along with data, from a specified interface. Because the port providing the network clock source could fail, Cisco IOS Software lets you configure additional interfaces as clock sources with priorities 1 to 4.

If the network clock source interface stops responding, the software switches to the next-highest-configured priority network clock source.

NOTE By default, the network clock is configured as nonrevertive. This is because the industry standard is to prefer a stable, if less-accurate, clock source over an unstable, even if more-accurate, clock source. Virtually every service provider's policy is to verify original clock source stability for a set period of several hours before reverting manually to that original source, while depending on the secondary, stable clock source in the interim.

The algorithm to switch to the highest-priority best clock runs only if you configure the **network-clock-select** command as revertive.

To configure the network clocking priorities and sources, use the following command in global configuration mode:

```
network-clock-select {priority} {bits} {atm} {system | card/port} [revertive]
```

The following example configures interface 0/0 as the highest-priority clock source to receive the network clocking, interface 0/2 as the second-highest priority, and interface 0/1 as the third-highest priority:

```
NI-2#config term
NI-2(config)#network-clock-select 1 atm 0/0
NI-2(config)#network-clock-select 2 atm 0/2
NI-2(config)#network-clock-select 3 atm 0/1
```

The following example shows how to configure the network clock to revert to the highest-priority clock source after a failure:

```
NI-2(config)#network-clock-select revertive
```

To configure the location from which an interface receives its transmit clocking, perform these tasks, beginning in global configuration mode:

Select the interface to be configured:

```
interface atm card/port
```

Configure the interface network clock source:

```
clock source {free-running | loop-timed | network-derived}
```

Network-derived means the highest-priority clock that is both configured and functional.

The following example configures ATM interface 0/1 to receive its transmit clocking from a network-derived source:

```
NI-2(config)#interface atm 0/1
NI-2(config-if)#clock source network-derived
```

Any module in a DSLAM chassis that can receive and distribute a network timing signal can propagate that signal to any similarly capable module in the chassis. The following entities can receive and distribute a Primary Reference Source (PRS) for synchronization:

- A Building Integrated Timing Supply (BITS) clock through the I/O card
- An OC-3/STM1 in an IP DSL Switch chassis
- A DS3/E3 module in an IP DSL Switch chassis that derives the clock from the trunk interface

The two trunk ports can propagate a clocking signal in either direction.

If you issue the **network-clock-select** command with the appropriate parameters, you can define any particular port in an IP DSL Switch chassis (subject to the previously discussed limitations) to serve as the clock source for the entire chassis or for other devices in the networking environment.

You can also use the **network-clock-select** command to designate a particular port in an IP DSL Switch chassis to serve as a master clock source for distributing a single clocking signal throughout the chassis or to other network devices. You can distribute this reference signal wherever the network needs to globally synchronize the flow of constant bit rate (CBR) data.

Step 6 Set the subtending numbers. If your DSL network will use subtending (subtending is discussed in Chapter 4), how can the network administrator guarantee adequate bandwidth and fairness of access for all subscribers in a subtended implementation?

The answer lies in leveraging the scheduler process on each IP DSL Switch that subtends another IP DSL Switch. (Incidentally, in a regular subtended tree configuration, there is automatic recovery from a failure of any node. In other words, a single failed node in the subtended family does not disrupt user traffic in the other, working IP DSL Switches.)

To guarantee that all subtended subscribers have equal access to the trunk port, the IP DSL Switch uses 13 numbered queues that are accessed in round-robin fashion. This approach keys off the **subtend-id,** which must be set in each subtended IP DSL Switch. This process is shown in Figure 6-6 and is

described in more detail in the following list:

A. Each IP DSL Switch has 13 queues, starting with 0, which services the local (onboard) DSL subscribers for each IP DSL Switch.

B. When traffic comes from a subtended trunk, the General Flow Control (GFC) number determines the queue in which the traffic will be put.

C. The GFC number for subtended nodes is set equal to the subtend-id plus 3.

D. In Figure 6-6, the last node on the subtend tree has a subtend-id of 2. Local DSL subscriber traffic populates queue 0.

E. When that traffic is transmitted on the subtended trunk (0/1), it is assigned a GFC of 5 (subtend-id + 3).

F. In subtend node 1, that traffic populates queue 5. Local DSL subscriber traffic populates queue 0. The node services these two queues in a round-robin fashion.

G. When subtend node 1 traffic from queue 0 is sent up the subtended trunk, it is assigned GFC = 4 (subtend-id + 3) and populates queue 4. Traffic from queue 5 retains GFC = 5 and populates queue 5.

H. In the top-level node, local DSL subscriber traffic populates queue 0.

I. The top subtend node services queues 0, 4, and 5 in a round-robin fashion, thereby giving all DSL subscribers equal access to the network trunk.

You can set the subtend node identifier using the following global command:

```
subtend-id node#
```

where *node#* is the node for which the command sets the subtend node identifier. The range is 0 to 12.

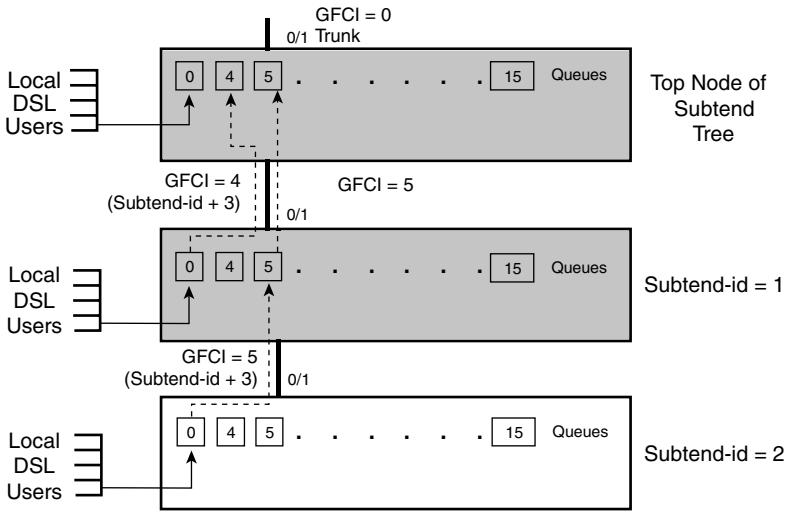
In this example, the command sets the DSL subtend node identifier to node 12:

```
NI-2#conf t
NI-2 (config)subtend-id 12
```

Step 7 Configure Simple Network Management Protocol (SNMP) for CDM use or to direct SNMP traps to another system. Enter the following commands while in privileged EXEC mode:

```
NI-2#config terminal
NI-2 (config)#snmp-server community <PUBLIC> ro
NI-2 (config)#snmp-server community <PRIVATE> rw
NI-2 (config)#snmp-server host <CEMF Server IP Address> traps version 2c public
NI-2 (config)#snmp-server enable traps
```

Figure 6-6 *Subtending Queues*



Step 8 After the basic NI-2 setup, you should verify that the NI-2 has autorecognized the line cards. Here is how that autodiscovery works:

The NI-2 autodiscovers all xTU-Cs (line cards) when they are installed. First the NI-2 verifies that the card is valid for that type of chassis. Second, for all but the Flexi card, the NI-2 compares the line card's installed software image to the particular software image that the NI-2 itself contains in its configuration. If the line card's installed image does not match the NI-2's image for that card type, the NI-2 automatically updates the line card with the software image that the NI-2 has available.

In the case of the Flexi card, the NI-2 can complete only the first step. As you saw in Chapter 4, the Flexi card offers connectivity for either the legacy Carrierless Amplitude Phase (CAP) modulation or the standardized DMT modulations (both full-rate and half-rate DMT). You must manually set the card to either CAP or DMT after initial recognition by the NI-2. (The Flexi card itself autorecognizes the difference between DMT2 and G.Lite coming from the CPE after the Flexi has been configured for DMT.) The NI-2 then completes its analysis of the Flexi line card's onboard image version and updates it as necessary. Here is the manual command format, starting in global configuration mode:

```
slot slot# cardtype
```

slot# is the slot number. For the 6160 IP DSL Switch, the range is 1 to 32. For the 6260 device, the range is 1 to 30. For the 6015, the range is 1 to 6.

cardtype is the modulation type for which you want to configure the slot.

Here's an example:

```
6160-94(config)#slot X atuc-4flexi{cap | dmt}
```

You cannot simply physically remove a line card from the chassis without changing the NI-2's configuration. To remove a line card from the NI-2's configuration, use the standard Cisco **no** option of the command for all card types, like this:

```
6160-94(config)#no slot X atuc-1-4dmt
```

At this point, after completing the previous eight steps (or the standard Cisco IOS Software startup configuration menu), with or without the optional subtending configurations, you could begin using the GUI CDM program. CDM lets you manage the IP DSL Switch, including provisioning the individual connections.

Continuing with Cisco IOS Software configuration of the NI-2, the next section describes configuring the NI-2 for redundant operations.

Redundancy Commands

Starting with Cisco IOS Software version 12.1(6)DA, NI-2 cards can be configured for redundancy in the IP DSL Switch. Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) NI-2s (OC-3 or STM1) have redundant physical links, and Automatic Protection Switching (APS) is in place. APS in the modern telecommunication equipment world is virtually hitless, because most modern fiber-equipped devices switch traffic in much less time than the APS-defined maximum 50 milliseconds. For instance, the NI-2 switches traffic in about 8 to 10 milliseconds. Because the human ear rarely detects gaps of 9 milliseconds or less in speech, even voice traffic is considered hitless. For data traffic, such a small interval is easily overcome by buffering and automatic retransmission if necessary.

For coaxial NI-2 cards, such as the DS3 and E3, if a physical link fails (the cable is cut), there is no switchover. Currently no redundant DS3/E3 physical links exist, so the NI-2 cannot overcome coaxial link failures. If a coaxial NI-2 card itself fails, there is protection, because the standby card takes activity automatically and takes over the coaxial ports on the IP DSL Switch I/O card on the backplane.

In cases of fiber- and coaxially-connected NI-2 failure, the secondary NI-2 assumes the configuration of the primary NI-2. All the ATM information stays the same, with the exception of the dynamically mapping switched virtual circuits (SVCs). These have to be reconfigured in the event of NI-2 failure.

The Ethernet IP address stays the same, but the MAC address might have to be reacquired through Reverse Address Resolution Protocol (RARP), depending on the degree of prior synchronization between the two NI-2 cards. Because the MAC address is stored on the I/O card, it can be shared between the NI-2s.

You must load the same redundancy-capable IOS image in both NI-2 cards. Do not install an NI-2 with IOS earlier than 12.1(2)DA in an IP DSL Switch with another NI-2 already in service. (A service interruption will result.)

APS-Related CLI Commands

The APS CLI commands include **show APS** and **show controllers**.

The **show APS** command displays APS state information.

The **show controllers** command is a classic IOS command that provides information on both the active and inactive OC-3/STM1 interfaces. You can use this command to determine which OC-3/STM1 NI-2 is online and which OC-3/STM1 interface is active. Following are seven examples of this command's output:

```
6160-1#show controller atm 0/1
IF Name: ATM0/1    Chip Base Address: B3809000
Port type: OC3  Port rate: 155000 kbps Port medium: SM Fiber

local             peer
                                (working)        (protection)

ACTIVE           INACTIVE

                                --------------  --------------
Port status                        Good Signal       Good Signal
Loopback                                  None              None
Flags                                   0x8300            0x8308
TX clock source                     loop-timed        loop-timed
Framing mode                            sts-3c            sts-3c
Cell payload scrambling                    On                On
Sts-stream scrambling                      On                On
TX Led:                                   Off               Off
RX Led:                                   Off               Off
TST Led:                                  Off               Off
```

The output of the command **show controller atm0/1** reports that the secondary NI-2 (slot 11, protection) is online. The primary NI-2 (slot 10) is installed, functional, and in standby mode. The OC-3 datastream received by the secondary NI-2 port 0/1 is the datastream being processed by the system (active).

Controller terminology includes the following:

- **Local**—This is always the online NI-2, either slot 10 or slot 11.
- **Peer**—This is always the standby card, either slot 10 or slot 11.
- **Working**—This is always the card in slot 10. *Working* is an APS term that is the same as the term *primary* in redundant NI-2 systems.

- **Protection**—This is the same as *secondary* in NI-2 systems. It is always slot 11.
- **Active**—The OC-3/STM1 receiving datastream on this port is the one used by the system.
- **Inactive**—The OC-3/STM1 receiving datastream on this port is not being used as the active traffic source.

Following is different output for the **show controller atm0/1** command:

```
6160-1#show controller atm0/1
IF Name: ATM0/1    Chip Base Address: B3809000
Port type: OC3  Port rate: 155000 kbps  Port medium: SM Fiber

                                 local              peer
                              (protection)        (working)
                                ACTIVE             INACTIVE
                             --------------   -------------
Port status                   Good Signal       Good Signal
Loopback                             None              None
Flags                              0x8300            0x8308
TX clock source                loop-timed        loop-timed
Framing mode                       sts-3c            sts-3c
Cell payload scrambling               On                On
Sts-stream scrambling                 On                On
TX Led:                              Off               Off
RX Led:                              Off               Off
TST Led:                             Off               Off
```

In this case, the output from the command **show controller atm0/1** reports that the secondary NI-2 (slot 11, protection) is online. The primary NI-2 (slot 10) is installed, functional, and in standby mode. The OC-3 datastream received by the secondary NI-2 port 0/1 is the datastream being processed by the system (active).

Here is the third example of the command **show controller atm0/1**:

```
6160-1#show controller atm 0/1
IF Name: ATM0/1    Chip Base Address: B3809000
Port type: OC3  Port rate: 155000 kbps  Port medium: SM Fiber
Alarms:  Source: ATM0/1 protect
Severity: CRITICAL Description: 12  Loss of Signal

                                 local              peer
                              (working)        (protection)
                                ACTIVE           INACTIVE
                             --------------   --------------
Port status                   Good Signal       SECTION LOS
Loopback                             None              None
Flags                              0x8300            0x8308
TX clock source                loop-timed        loop-timed
Framing mode                       sts-3c            sts-3c
Cell payload scrambling               On                On
Sts-stream scrambling                 On                On
TX Led:                              Off               Off
RX Led:                              Off                On
TST Led:                             Off               Off
```

In this case, the output indicates a loss of signal (LOS) on the protect (secondary) NI-2 ATM0/1 port. The primary NI-2 (slot 10) is online, and the active OC-3 datastream is on slot 10.

Here is the fourth example of the command **show controller atm0/1**:

```
6160-1#show controller atm0/1
IF Name: ATM0/1    Chip Base Address: B3809000
Port type: OC3  Port rate: 155000 kbps  Port medium: SM Fiber

Alarms:
Source: ATM0/1 protect
Severity: CRITICAL
Description: 12 Loss of Signal

                                  local             peer
                               (protection)       (working)
                                INACTIVE           ACTIVE
                              ---------------   ---------------
Port status                       SECTION LOS       Good Signal
Loopback                             None              None
Flags                              0x8300            0x8308
TX clock source                  loop-timed        loop-timed
Framing mode                         sts-3c            sts-3c
Cell payload scrambling                 On                On
Sts-stream scrambling                   On                On
TX Led:                                Off               Off
RX Led:                                 On               Off
TST Led:                               Off               Off
```

In this case, the output from **show controller atm0/1** reports a LOS on the protect (secondary) NI-2 0/1 port (slot 11), but the secondary NI-2 is online. The OC-3 datastream received by the NI-2 in slot 10 is being processed by the system (NI-2 in slot 11).

Here is the fifth of the seven examples of the command **show controller atm0/1**:

```
6160-1#show controller atm0/1
IF Name: ATM0/1    Chip Base Address: B3809000
Port type: OC3  Port rate: 155000 kbps  Port medium: SM Fiber

                                  local             peer
                               (working)        (protection)
                                 ACTIVE           INACTIVE
                              ---------------   ---------------
Port status                       Good Signal     Not available
Loopback                             None         Not available
Flags                              0x8300         Not available
TX clock source                  loop-timed       Not available
Framing mode                         sts-3c       Not available
Cell payload scrambling                 On        Not available
Sts-stream scrambling                   On        Not available
TX Led:                                Off        Not available
RX Led:                                Off        Not available
TST Led:                               Off        Not available
```

The *Not available* status for peer (protection) in the output indicates that the standby NI-2 in slot 11 has not booted to a standby state. This is a normal status after switching from active to standby. *Not available* is reported for approximately 60 seconds after switchover, while the previously online card restarts.

Following is the sixth example of **show controller atm0/1**. In this case, it is for a coaxial (DS3) NI-2 variant:

```
6160-1#show controller atm0/1
IF Name: ATM0/1, Chip Base Address: B3809000
Port type: DS3 Port rate: 45000 kbps    Port medium: Coax
Loopback:None  Flags:8000 Port status: LOS
Source: ATM0/1
Severity: CRITICAL
Description: 6  LOS Detected
TX Led: Off    RX Led: On    TST Led: Off
TX clock source:  network-derived
DS3 Framing Mode:  m23 plcp
FERF on AIS is on
FERF on LCD is on (n/a in PLCP mode)
FERF on RED is on
FERF on OOF is on
FERF on LOS is on
LBO: <= 225'
```

The output shows a LOS on the DS3 trunk port 0/1. The status is reported by the online NI-2.

Finally, the following is the last output example for **show controller atm0/1**, following the command **redundancy switch-activity**. The online NI-2 reboots but does not complete the restart sequence, as evidenced by the following output:

```
Cisco Internetwork Operating System Software
IOS (tm) NI2 Software (NI2-DSL-M), Experimental Version 12.1(20000906:224310)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Thu 21-Sep-00 14:54 by satrao
Image text-base: 0x800082B8, data-base: 0x80BD6000
*** This is the STANDBY unit. Initialization is being held. ***
```

This standby NI-2 completes its restart sequence and becomes the online unit. After the statement *Initialization is being held* is sent, the console port is inactive.

Secondary Unit Sync Commands

This section describes different levels of synchronization between the primary and secondary NI-2 cards and the commands to set up the synchronization. The NI-2 cards have different levels of memory and therefore different levels of synchronization of the memory contents. As with almost all Cisco IOS Software-based devices, there is bootflash memory, Flash memory, NVRAM memory, and the running configuration itself, which might not be saved in a particular moment in time.

For optimal redundancy preparedness, you should ensure that both NI-2s have the same running configuration and startup configuration. These two files are automatically synchronized between the two NI-2s when both NI cards are loaded with a redundancy-capable version of IOS, 12.1(6) or later (you should have identical IOS versions loaded on both cards). However, auto-sync can be enabled only when Flash itself is in sync. You must manually sync Flash before enabling auto-sync, because **auto-sync flash** is *not* enabled by default.

Here is the command to synchronize Flash, along with its related command to synchronize bootflash:

```
NI-2(config)#auto-sync flash
NI-2(config)#auto-sync bootflash
```

After you enter at least the first command, if not both, the configurations are automatically synchronized on an ongoing basis by default.

For verification, you can also directly enter the following commands to enable automatic synchronization:

```
NI-2(config)#auto-sync running-config
NI-2(config)#auto-sync config
```

You can also directly create synchronization for each type of memory on the NI-2 cards with the following commands:

- The **NI-2#secondary sync flash** command mirrors Flash on the secondary NI-2 by copying the contents of the primary NI-2 Flash to the secondary NI-2 Flash. This function is disabled by default.
- The **NI-2#secondary sync bootflash** command provides the same function as **sync flash**, but for bootflash. This function is disabled by default.

Verifying Memory Content

The following commands involve checking memory contents (don't overlook the ending colons!) from the privileged EXEC (enable) prompt.

The following two commands allow a quick comparison of which files are in Flash memory on the primary and secondary NI-2s:

```
NI-2#dir flash:
NI-2#dir secondary-flash:
```

The next two commands allow a quick comparison of the files in bootflash on the primary and secondary NI-2:

```
NI-2#dir bootflash:
NI-2#dir secondary-bootflash:
```

The next two commands allow a quick comparison of primary and secondary NVRAM, which is the simplest way to verify synchronization of the two NI-2 cards' configurations:

```
NI-2#dir nvram:
NI-2#dir secondary-nvram:
```

You can gain access to each NI-2 independently, which results in two different NI-2s unless auto-sync mode is in effect. You might use the following command if one of the NI-2s is being prepared for an upgrade or for use in another Cisco 6000 series IP DSL Switch or is a regional spare that happens to be housed onboard this device.

Disable split mode with the **NI-2#split-mode [enable | disable]** command to access each NI-2 independently.

Reload Redundancy Commands

The two redundancy commands that reload NI-2 cards are

- **NI-2#redundancy reload-peer**—Reloads the standby NI-2 card from Flash memory.
- **NI-2#redundancy reload-shelf**—Reloads both NI-2 cards (typically from Flash).

Changing Roles of NI-2 Cards

IOS commands can be used to manually switch the active receive port from primary to secondary and from secondary to primary. The first command is primarily used when neither NI-2 card is carrying traffic. The other commands are used to force a switch that might result in a loss of data. All the following commands are entered at the privileged EXEC (enable) prompt.

To manually switch from the active NI-2 to the standby NI-2, use the following command:

```
NI-2#redundancy switch-activity
6160-1#aps force atm 0/1 from working
6160-1#aps force atm 0/1 from protection
```

IOS commands and responses can also show an attempt to manually switch from a working interface to a faulty interface, as shown here:

```
NI-2#aps manual atm 0/1 from working
NI-2#ATM0/1 Protection link is not available
NI-2#
```

The **aps** command instructs the system to use the datastream from the named fiber interface. The **manual** option performs an audit before switching to ensure that the named datastream is available. The **force** option simply makes the switchover, even if no datastream is present on the named interface. The option **from protection** instructs the system to use the datastream received at the working interface instead of the protection interface. The option **from working** instructs the system to use the datastream received at the protection (standby) interface instead of the working interface. This configuration control allows maintenance on optical fiber systems without affecting traffic. The next step after switching the NI-2 cards is to shut down the NI-2 that is now inactive, as shown in the next command.

The **NI-2#shutdown interface configuration** command applies to both the active and protection OC-3/STM1 interfaces at the same time. The following command sequence shuts down all ATM traffic on the trunk port of both the active and standby NI-2 cards:

```
NI-2#config t
NI-2#[config-if]interface atm 0/1
NI-2#[config-if]shutdown
```

The next command disables automatic and manual APS. After entering this command, you can proceed to replace or otherwise work on the disabled unit:

```
NI-2#aps lockout
```

Verifying Redundancy States

The **NI-2#show redundancy states** command shows important information about NI-2 card and port status, used to show which card is online. Following are two sample outputs from this command:

```
NI-2#show redundancy states
       my state = 11 -ACTIVE
     peer state = 8  -STANDBY READY
           Mode = Duplex (The standby NI-2 is operational, otherwise this shows
  simplex)

           Unit = Primary (The on-line NI-2 is slot 10, the primary)


   Config  Sync = Enabled
  File Sys Sync = Enabled
      Bulk Sync = Enabled
   Dynamic Sync = Enabled

     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up
```

This output shows that the NI-2 unit in slot 10 (primary) is online. The mode is duplex, indicating that a redundant NI-2 (slot 11) is installed and functional. (Note that state = 11 does not imply slot 11.) If the standby unit is not installed or not functional, the mode is simplex, as shown here:

```
barf1#show redundancy states
       my state = 11 -ACTIVE
     peer state = 8  -STANDBY READY
           Mode = Duplex
           Unit = Secondary

   Config  Sync = Enabled
  File Sys Sync = Enabled
      Bulk Sync = Enabled
   Dynamic Sync = Enabled

     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up
```

This output shows that the NI-2 unit in slot 11 (secondary) is online. The mode is duplex, indicating that a redundant NI-2 (slot 10) is installed and functional.

With these commands, you can set up, manage, and display reports of redundancy on the Cisco IOS Software-based NI-2 cards on the 6000 series of IP DSL Switches. After you have configured the NI-2 cards for basic operations, you can proceed to configure them to perform DSL-to-ATM switching, which is explained in the next section.

Configurations for DSL-to-ATM Switching

DSL-to-ATM switching involves defining the DSL circuit parameters on the Cisco 6000 series IP DSL Switch and continuing the ATM circuit (PVC) through the NI-2 card to the ATM network. You can define the DSL circuit parameters and map the ATM circuits in either the Cisco IOS Software CLI or the CDM GUI software. This section discusses the Cisco IOS Software commands, and the next chapter discusses the CDM activities.

After you configure the Flexi line card to specify the precise ADSL type of traffic it will carry, and after the NI-2 cards automatically recognize the other line cards, you can proceed with the creation of subscriber profiles. Subscriber profiles provide various service levels for individual DSL subscribers. Profiles are discussed in the following sections.

Defining Profiles

A *profile* is a named list of specified values. To configure a subscriber, you attach a profile to that subscriber's port. You can change the configured items for a subscriber simply by changing that subscriber's profile.

There is a provided profile named *default*. You may configure the default profile with specific parameters, but you may not delete it. When you create new profiles, each new profile automatically takes on the values of the default profile unless you specifically override those values. This is useful when you want to modify one or two default parameters and apply the changes to every port in the system. This lets you avoid creating a new profile with minor changes and associating the new profile with every port in the system. Work smarter, not harder.

Except for a few dynamic operational modes, such as rate adaptation due to local impairments, port configuration takes place only through a configuration profile rather than by direct configuration. If you modify an existing profile, the change takes effect on every ADSL port linked to that profile, but only after all related connections are resynchronized (retrained).

To create or delete a DSL profile, or to select an existing profile for modification, use the following global commands:

- **dsl-profile profile-name**—Creates a new profile.
- **no dsl-profile profile-name**—Deletes an existing profile. A profile can be deleted only when it is no longer associated with any port.

profile-name is the name of the profile you want to create or an existing profile you want to delete or modify. Profile names are case-sensitive.

Remember that when you create a profile, it inherits all the configuration settings of the special profile named default. If you subsequently modify the special profile named default, the changes do not propagate to the previously created profiles.

The following command creates a profile called ALPHA_USERS:

```
c6260 (config)#dsl-profile ALPHA_USERS
```

Within this profile are many options for service parameters. In reality, you should not need the majority of these. In theory, you could define a unique profile for every subscriber, applying a different profile to each port. A typical service provider marketing plan would have no more than about a dozen profiles at most. For instance, the marketing department might offer both business and residential service profiles, each category of which might contain three different bit rates and other basic parameters, for a total of six levels of service. If you added a specialized service, such as for streaming video only, that might require one or two more profiles. You can see that unless the service provider's marketing department goes frantic after too much coffee in too many meetings, any DSL network should have no more than 10 or 12 profiles at most.

The following are examples of parameter options for profiles. These commands are entered in dsl profile configuration mode:

- **Setting the bit rate**—To set the maximum and minimum allowed bit rates for the fast-path and interleaved-path DMT profile parameters, use the following command:

  ```
  dmt bitrate max interleaved downstream dmt-bitrate upstream dmt-bitrate
  ```

  dmt-bitrate is a multiple of 32 kbps. If you enter a nonmultiple of 32 kbps, the Cisco IOS Software parser code rejects and aborts the command.

  In the following example, the command sets the maximum interleaved-path bit rate of the default profile to 8032 kbps downstream and 832 kbps upstream:

  ```
  NI-2#conf t
  NI-2(config)#dsl-profile default
  NI-2(config-dsl-profile)#dmt bitrate maximum interleaved downstream 8032
    upstream 832
  ```

- **Setting the margin**—To set upstream and downstream signal-to-noise ratio (SNR) DMT margins, use the following command:

  ```
  dmt margin downstream dmt-margin upstream dmt-margin
  ```

  dmt-margin equals the upstream and downstream SNR margins in decibels. Values must be nonnegative integers. The range is from 0 to 15 dB. The default is 6 dB in each direction; this default is recommended by the DSL Forum and was adopted by Cisco.

CAUTION The margin command causes the port to retrain when you change the parameter. Setting a parameter to its previous value does not cause a retrain. If a port is training when you change this parameter, the port untrains and retrains to the new parameter.

In this example, the command sets the default profile's SNR DMT margins to 6 dB upstream and 3 dB downstream:

```
NI-2#conf t
NI-2(config)#dsl-profile default
NI-2(config-dsl-profile)#dmt margin downstream 3 upstream 6
```

- **Setting check bytes**—As discussed in Chapter 1, "DSL Primer," check bytes are the redundant, unaltered bytes that are used to verify DSL interleaving and deinterleaving. This interleaving process minimizes the impact of the inevitable bit errors, spreading out missed bits rather than losing a sequential string of bits. The higher the number of check bytes, the more accurately the algorithm can code and decode the straight bits, but a higher number of check bytes also means more overhead, reducing data throughput.

  Here is how to define the check bytes:

```
NI-2#conf t
NI-2 (config)#dsl-profile default
NI-2 (cfg-dsl-profile)#dmt check-bytes interleaved downstream number check
  upstream number
```

  number is the number of redundancy check bytes per DMT frame. Values are from 0 to 16 bytes in increments of 2 (0, 2, 4, ... , 14, 16). You can set different values for both upstream and downstream, but you must reference (specify) both directions even if the number of check bytes is equal in both directions. The default is 16 (the maximum) in both directions. You can also set the number of check bytes by turning off check bytes. This means that the system determines and uses the optimal number of check bytes for that line.

  In general, you should probably use the default settings, unless and until the local transmission environment or the individual service-level agreement dictates otherwise.

- **Set the interleaving delay**—Presuming that this is not a fast-path service (see Chapter 1's discussion of fast and interleaved paths in DSL service), the interleaving delay helps protect against impulse noise and clipping. However, it adds delay, which might not be tolerable for some applications. To set the interleaving delay parameter, use the following command:

```
dmt interleaving-delay downstream delay-in-usecs upstream delay-in-usecs
```

  delay-in-usecs is the interleaving delay in microseconds. The default interleaving delay is 16000 microseconds (16 milliseconds) for both upstream and downstream directions. Allowable values are 0, 500, 1000, 2000, 4000, 8000, and 16000 microseconds.

CAUTION Like the **margin** command, this command causes the port to retrain when you change the parameter. Setting this parameter to its current value does not cause a retrain. If a port is training when you change the value, the port untrains and retrains to the new value.

In the following example, the command sets the default profile's interleaving delay to 2000 microseconds downstream and 4000 microseconds upstream:

```
NI-2#conf t
NI-2(config)#dsl-profile default
NI-2(config-dsl-profile)#dmt interleaving-delay downstream 2000 upstream 4000
```

The four parameters are not all the options available, but they are probably the most common settings made for typical DSL networks. You can even safely use the default settings, such as 6 dB for the margin, and just define the bit rate, for a very simple and quick beginning profile. The very simplest starting procedure is to keep using the default profile, possibly making a few changes to its parameters, which is already attached to every port until you specify a new profile.

In this case, after defining the new profile with the bit rates, margin, check bytes, and interleave delay, you can proceed to assign the profile to the port(s) desired. At this time you cannot attach a profile to a group or range of ports, so you must address each port individually.

Attaching and Detaching a Profile

To attach a profile to or detach a profile from a specific port, use the following interface commands:

- **dsl profile** *profile-name* (to attach)
- **no dsl profile** *profile-name* (to detach from the port)

profile-name is the profile you want to attach to or detach from the selected port.

In the following example, the command attaches the profile ALPHA_USERS to slot 20, port 1:

```
NI-2#conf t
NI-2(config)#int atm 20/1 (Enters interface configuration mode for port 20/1)
NI-2(config-if)#dsl profile ALPHA_USERS (Attaches ALPHA_USERS profile to this
  port)
```

In the following example, the command detaches the profile ALPHA_USERS from slot 20, port 1:

```
NI-2#conf t
NI-2(config)#int atm 20/1 (Enters interface configuration mode for port 20/1)
NI-2(config-if)#no dsl profile ALPHA_USERS (Detaches ALPHA_USERS profile from this
  port)
NI-2(config-if)#exit
```

The preceding commands detach this particular instance (application) of the profile from this particular port. However, the original definition of the profile, the class, still exists in the overall configuration. After you have detached the profile from the port, then and only then can you delete the original definition of the profile from the overall configuration, if you're sure you will never use this profile again. To do this, use the following command:

```
NI-2#conf t
NI-2(config)#no dsl-profile ALPHA_USERS
```

PVCs

Having defined and attached your profile, you can move on to creating an ATM connection from the line card port to the network trunk. The simplest type of connection is a PVC, which is the starting basis for the more-sophisticated soft-PVC and Permanent Virtual Path (PVP).

PVCs must be configured in both the IP DSL Switch and the ATM switch cloud. PVCs remain active until the circuit is removed from either configuration.

To create a PVC on an ATM interface, use the **atm pvc** interface configuration command. The **no** form of this command removes the specified PVC. The labor-saving news is that PVCs are bidirectional and need to be removed from only one direction or the other.

Here's the syntax:

```
atm pvc  vpi vci int atm [slot/port] vpi  vci
```

vpi is the virtual path identifier for this PVC. The *vpi* value is unique on only a single link, not throughout the ATM network, because it has local significance only. The *vpi* value must match that of the switch.

vci is the ATM network's virtual channel identifier. It is in the range of 0 to 1 less than the maximum value set for this interface by the ATM per-VC per-VP command. The *vci* value is unique on only a single link.

As an example, suppose that a DSL circuit is connected to DSL port 1 on line card 19. On that DSL side, it is configured as an ATM PVC using the virtual path of 0 and the virtual channel of 33. Here are the commands to switch that incoming connection to VPI 0, VCI 100 outbound on the network trunk, which is the ATM interface 0/1:

```
6260(config)#int atm 19/1
6260(config-if)#atm pvc 0 33 int atm 0/1 0 100
```

In a subtended configuration, you must configure the passthrough connections from the subtended trunk to the network trunk. In this case, you can build on the previous line card-to-trunk definitions to make the same PVC follow through to the host device's ATM trunk, 0/1 again. Bear in mind that the PVC definition turned the PVC 0/100 loose on this IP DSL Switch's incoming trunk 0/2. Here are the commands to switch the passthrough (subtended) PVC through this IP DSL Switch onto the network trunk, which is ATM 0/1:

```
6260(config)#int atm 0/2
6260(config-if)#atm pvc 0 100 int atm 0/1 0 200
```

Even if you use the same numbers for both ends of the connection, both the line card/port and the trunk or the trunk-trunk connection, you must specify the whole line, both incoming and outgoing identifiers.

You should be starting to realize that configuring each DSL subscriber's port on the IP DSL Switch with PVCs is quite labor-intensive. Remember also that you can configure multiple PVCs on the Cisco 827 DSL router—that is, one PVC for voice service and another for data traffic. You can also create two or more PVCs on each line card port on the IP DSL Switch.

Obviously, using nothing but PVCs would be inefficiently burdensome. In the interest of working smarter, not harder, consider the next two labor-saving ATM connections.

Soft-PVCs (SPVCs)

The SPVC is a combination of permanent circuits, manually configured at each end of the connection, and an SVC through the middle of the connection. The switched circuit depends on the ATM format's internal communications and routing protocols—specifically, the autoconfigured point-to-point signaling connection using VCI 5. (For more information, refer to Appendix B, "ATM Overview.")

Soft-PVCs have two main advantages. The first advantage is its flexible sustainability. If a particular link is unavailable, as with a cable cut, the SPVC takes advantage of the ATM network's other paths to reach its destination. This automatic PVC rerouting does not require human intervention, unlike the PVC in the case of connection unavailability. The second SPVC advantage is its efficiency in configuration labor. You can trust the ATM signaling protocols to automatically allocate available identifiers, both VPIs and VCIs, at each interface, eliminating the need to manually define each new connection.

However, these two advantages also suggest the SPVC's disadvantages. First, the SPVC requires more overhead to accommodate the signaling and updating required to maintain the network knowledge. Second, the SPVC is harder to trace, unlike the PVC, which is mapped to a known set of identifiers, because the SPVC takes on dynamic VPIs and VCIs at each interface.

Overall, many providers prefer to save human labor and tax the system itself, so you should know how to create a soft-PVC. You can use these steps:

Step 1 Find the target device's ATM address, such as the host IP DSL Switch in a subtending configuration, or the Layer 3 termination device, such as the 6400 UAC. You can do this with the IOS command **show atm addresses** on the target device. Just as obvious as the command is the listing of ATM addresses, with their own heading of *Soft-PVC Addresses* in the case of the 6400. The ATM address probably looks like this:

```
47.0091.8100.0000.0030.7b2d.0001.4000.0c80.0010.00
```

NOTE For efficiency and to prevent human hand-copying errors, copy that ATM address to your computer's software clipboard.

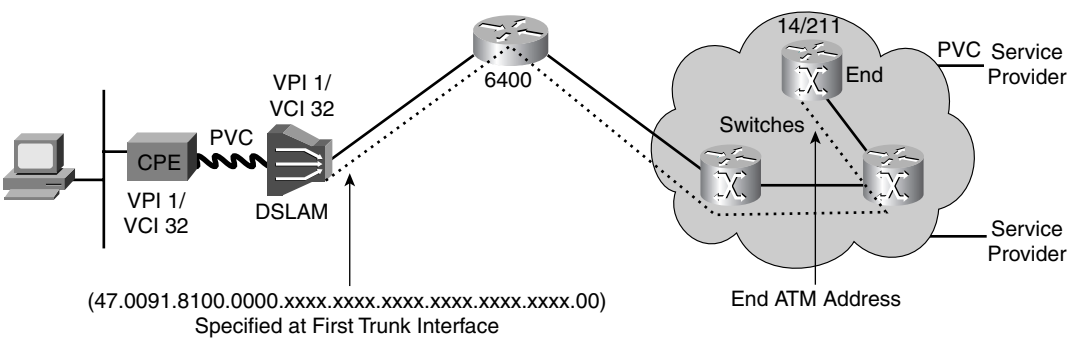
Step 2 Telnet back (or otherwise connect) to the IP DSL Switch's NI-2. You cannot duplicate the existing PVC identifiers, so you must either remove the existing PVC or use a different set of identifiers. (Remember that you will have the same numbers of VCIs on differently-numbered VPIs as long as those combinations are still unique.)

Step 3 Map the new soft-PVC from the line card port to the target ATM address rather than out the DSLAM's network trunk interface as you would for a standard PVC. You specify that destination address rather than the outgoing network trunk. That is, the only interface you specify is the line card/port, *not* the network trunk interface. Following are the commands:

```
NI-2(config)#int atm 21/4 (for the 21st line card's fourth port)
NI-2(config-if)#atm soft-pvc 0 101 dest-address
  47.0091.8100.0000.0030.7b2d.0001.4000.0c80.0010.00 0 200
```

The command in interface configuration mode is actually all on one line. You can see why it's easier to paste the ATM address you obtained in the first step! Notice also that you must specify the exiting VPI and VCI—in this case, VPI 0 and VCI 200. That's how the circuit is identified at the other end, the target ATM device, although it takes on seemingly random identifiers in the ATM network between the two end devices. This is shown in Figure 6-7 (which is also repeated in Appendix B).

Figure 6-7 *Soft-PVC Mapping*



PVPs

Another type of labor-saving ATM connection is a PVP. As mentioned in Appendix B's discussion of basic ATM, a PVP is a manually configured connection that carries a bundle of virtual circuits. This is commonly implemented between ATM switches within an ATM network (node to node). The advantage of this type of connection is that a single VPI configured through several ATM switches carries thousands of VCs without the need to individually configure them. The common VPI means that any single channel with that VPI is automatically assimilated into the larger path without having to configure each circuit like adding threads.

Many service providers use a PVP to group the individual PVCs from a subtended IP DSL Switch so that the VPI highlights the common source as the subtended connections traverse the hosting devices. Another use of PVPs is to group connections with a common service-level agreement, according to an obvious mapping scheme that matches a VPI with a type of DSL service.

The first disadvantage of the PVP is that you cannot extract or work with a single PVC within the larger pipe without rebuilding the entire PVP. Among other considerations, this means that you must be careful when identifying the starting PVCs, because the system assimilates all the common VPIs regardless of whether you want to bundle them into that PVP. Second, as with the PVC, you must define the PVP at each interface through which it traverses (although defining a single element is still much easier than defining the thousands of separate PVCs). A third disadvantage is like the PVC's fragility: If a link is broken, as with a cable cut, the PVP must be manually reconfigured.

Following are the commands to configure a PVP:

- **NI-2(config)#int atm 0/2**—This command begins to configure the first subtending trunk interface, ATM 0/2.
- **NI-2(config-if)#atm pvp 0 int atm 0/1 0**—In the interface configuration command, notice that you specify only the VPI—0 in this case—on both ends, the subtending trunk of 0/2 and the network trunk of 0/1. This makes VPI 0 unavailable for any incoming connections on this host IP DSL Switch, and you must have previously identified all the desired incoming connections on the subtended IP DSL Switch with the common VPI of 0.

  You can repeat the PVP definition at the next ATM device, whether it is another IP DSL Switch in a subtended daisy chain or tree, or any other device. You can even change the PVP number as you did the PVC earlier, like this:

  ```
  Nextdevice(config)#int atm 0/2
  Nextdevice(config-if)#atm pvp 0 int atm 0/1 14
  ```

  In the second command, all the individual PVCs, which started their lives with the VPI of 0, are now grouped in the PVP numbered 14.

Configuring Inverse Multiplexing Over ATM (IMA)

As discussed at length in Chapter 4, IMA is most common on the Cisco 6015 IP DSL Switch, but it is available on the other switches in the 6000 series as well. As a very brief reminder here, the T1/E1 IMA feature aggregates multiple low-speed links (T1/E1) into one or more IMA groups. These multiple ATM links act as a single ATM physical layer element.

To enable IMA, you can configure any WAN interface (the DS3, any T1 link, any E1 link, or any IMA group) as the trunk. When you configure a T1 link or an IMA group as the trunk, the DS3 port is disabled. When you select the DS3 port as the trunk, the T1 links and IMA groups are all treated as subtended ports.

Configuring IMA involves three major processes:

- Configuring a trunk interface
- Configuring T1/E1 interfaces
- Configuring IMA interfaces

First, configure the trunk interface with the following command, which designates the interface to use as the trunk—in this case, the 0/1 (network) trunk:

```
6015(config)#atm ni2-switch trunk atm 0/1
```

Next, select the link's transmit clock source, which you learned about earlier:

```
6015(config-if)#clock source network-derived
```

Select the link's framing type, which must match on both sides of that link:

```
6015(config-if)#framing m23adm
```

Enable DS3 cell payload scrambling on the link; scrambling is required if you use ami line coding:

```
6015(config-if)#scrambling cell-payload
```

Specify the cable length line build-out (**short** or **long**), followed by mandatory values: the length following **short**, or the gain and margin value following **long**. You can view the acceptable lengths, such as 0 to 133 feet for **short**, by including the **?** option after the **long** or **short** commands. The default setting is long haul with gain36 and 0db (**lbo long gain36 0db**):

```
6015(config-if)#lbo short
```

Next, configure each T1/E1 interface that will go into the IMA group(s), starting in interface configuration mode for each interface in turn:

```
6015(config)#interface atm 0/2
```

If ami line coding is selected, as it is in the next command for this T1 link, you must have enabled cell scrambling on the link. T1 and/or E1 links have different, specific options, although ami applies to both T1 and E1 links. The defaults are b8zs for the T1s and hdb3 for the E1s.

```
6015(config-if)#linecode ami
```

Next, select the frame type for the T1 or E1 data links. The framing type must match on both sides of the link. The defaults are as follows:

E1—**pcm30**

T1—**esf**

DS3—**cbitadm**

```
6015(config-if)#framing esf (or framing pcm30 for the E1)
```

Specify the line build-out (LBO) length as either **short** or **long**, followed by the appropriate parameters, as you did earlier for the trunk interface. Here is an example of the **lbo** command in this case:

```
6015(config-if)#lbo short 133
```

Now you are ready for the third major step, which is to configure the IMA interfaces themselves. You repeat the following command sets for each IMA interface you want to configure:

```
6015(config)#interface atm 0/2 (first T1/E1 to be assigned to an IMA group)
```

Assign the ATM interface to an IMA group (numbered from 0 to 3, for a total of four possible IMA groups). After the interface is assigned to an IMA group, individual ATM functionality is no longer available on the link:

```
6015(config-if)#ima-group 2
```

Enable the individual link by canceling the shutdown state:

```
6015(config-if)#no shutdown
```

Now that you have created an IMA interface from the individual links, you can begin configuring the IMA interface as a whole:

```
6015 (config-if)#interface atm0/ima2
```

Select the transmit clock mode for the selected IMA group:

```
6015(config-if)#ima clock-mode independent
```

Enter the maximum differential delay in milliseconds for the selected IMA group. Although the ranges are different for T1s and E1s, the default for both is 25 milliseconds, which is the minimum delay for both these standards:

```
6015(config-if)#ima differential-delay-maximum 68
```

Enter the minimum number of links that need to be operational for the selected IMA group:

```
6015(config-if)#ima active-links-minimum 2
```

Enable the IMA group by canceling the shutdown state:

```
6015(config-if)#no shutdown
```

This completes the three major processes for configuring the NI-2 for IMA. The next section describes the commands to verify the configuration.

Verifying IMA Status

After you have configured IMA, you should verify the operational status of the IMA interfaces using these **show** commands:

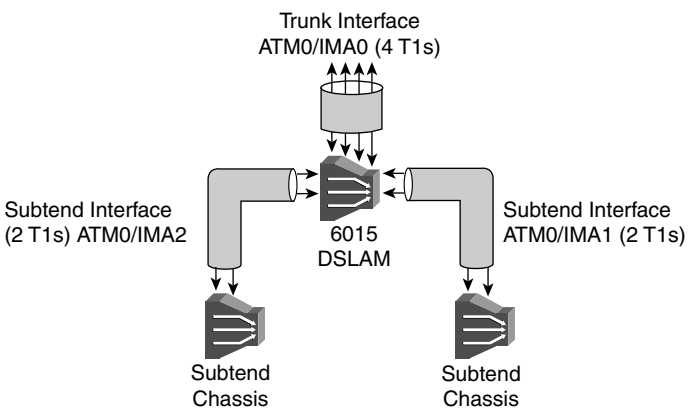
- **6015#show ima interface**—Displays information about all IMA groups and the links in those groups.
- **6015#show interface atm0/ima2**—Displays interface configuration, status, and statistics for the IMA interface.
- **6015#show controllers**—Displays information about current settings and performance at the physical level. You saw detailed examples of this command earlier, in the section "APS-Related CLI Commands."

- **6015#show controller atm0/ima2**—Displays diagnostic information for the specified IMA group.
- **6015#show ima interface atm0/ima2**—Displays configuration information and operational status for the specified IMA group.
- **6015#show ima interface atm0/2**—Displays information for a single link in an IMA group.
- **6015#show ima counters**—Displays IMA statistics in 15-minute intervals, with 24-hour totals.

Now you can verify your work with sample configurations. The first sample configuration describes how to configure the topology shown in Figure 6-8, which consists of the following:

- An IMA group containing four links as a trunk interface
- Two IMA groups, each containing two links, connecting subtended Cisco 6000 series IP DSL Switch chassis

Figure 6-8 *IMA Trunk with IMA Subtended Chassis*



Here is the configuration, shown in the running config output:

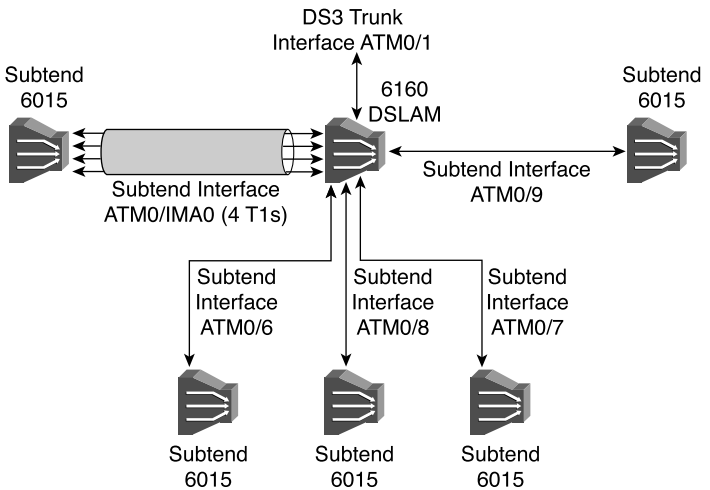
```
atm ni2-switch trunk ATM0/IMA0 !Configures interface ATM0/IMA0 as the trunk!
!
interface ATM0/0
 no ip address
 no ip route-cache
 atm maxvp-number 0
 atm maxvc-number 4096
 atm maxvci-bits 12
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
 no ip route-cache
```

```
!
interface ATM0/1
 no ip address
 no ip route-cache
 shutdown
 no atm ilmi-keepalive
!
interface ATM0/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
 ima-group 0 !Adds this interface to IMA group 0!
!
interface ATM0/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
 ima-group 0 !Adds this interface to IMA group 0!
!
interface ATM0/4
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
 ima-group 0 !Adds this interface to IMA group 0!
!
interface ATM0/5
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
 ima-group 0 !Adds this interface to IMA group 0!
!
interface ATM0/6
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
 ima-group 1 !Adds this interface to IMA group 1!
!
interface ATM0/7
 no ip address
```

```
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
 ima-group 1 !Adds this interface to IMA group 1!
!
interface ATM0/8
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
 ima-group 2 !Adds this interface to IMA group 2!
!
interface ATM0/9
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
 ima-group 2 !Adds this interface to IMA group 2!
!
interface ATM0/IMA0 !IMA group 0 configuration!
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 ima active-links-minimum 2
 ima clock-mode independent
 ima differential-delay-maximum 68
!
interface ATM0/IMA1 !IMA group 1 configuration!

 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 ima active-links-minimum 2
 ima clock-mode independent
 ima differential-delay-maximum 68
!
interface ATM0/IMA2 !IMA group 2 configuration!
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 ima active-links-minimum 2
 ima clock-mode independent
 ima differential-delay-maximum 68
!
interface ATM0/IMA3
 no ip address
 no ip route-cache
 shutdown
 no atm ilmi-keepalive
```

This output shows the configuration for an IMA trunk with IMA-subtended chassis for the Cisco 6000 series IP DSL Switch. The next configuration is for the topology shown in Figure 6-9.

Figure 6-9 *DS3 Trunk with IMA and T1 Subtended Chassis*



This configuration contains a combination of IMA, T1, and DS3 interfaces:

```
atm ni2-switch trunk ATM0/1 !DS3 is the default trunk!
!
interface ATM0/0
 no ip address
 no ip route-cache
 atm maxvp-number 0
 atm maxvc-number 4096
 atm maxvci-bits 12
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
 no ip route-cache
!
interface ATM0/1
 no ip address
 no ip route-cache
 no atm ilmi-keepalive
!
interface ATM0/2
 no ip address
 no ip route-cache
 no atm ilmi-keepalive
 ima-group 0 !Adds this interface to IMA group 0!
!
interface ATM0/3
 no ip address
 no ip route-cache
 no atm ilmi-keepalive
 ima-group 0 !Adds this interface to IMA group 0!
```

```
!
interface ATM0/4
 no ip address
 no ip route-cache
 no atm ilmi-keepalive
 ima-group 0 !Adds this interface to IMA group 0!
!
interface ATM0/5
 no ip address
 no ip route-cache
 no atm ilmi-keepalive
 ima-group 0 !Adds this interface to IMA group 0!
!
interface ATM0/6 !T1 configuration!
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
!
interface ATM0/7 !T1 configuration!
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
!
interface ATM0/8 !T1 configuration!
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
!
interface ATM0/9 !T1 configuration!
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 clock source loop-timed
 scrambling cell-payload
 linecode ami
 lbo short 133
!
interface ATM0/IMA0 !IMA group 0 configuration!
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 ima active-links-minimum 4
 ima clock-mode independent
 ima differential-delay-maximum 68
!
interface ATM0/IMA1
 no ip address
```

```
 no ip route-cache
 shutdown
 no atm ilmi-keepalive
!
interface ATM0/IMA2
 no ip address
 no ip route-cache
 shutdown
 no atm ilmi-keepalive
!
interface ATM0/IMA3
 no ip address
 no ip route-cache
 shutdown
 no atm ilmi-keepalive
```

Verifying NI-2 Processes with **show** Commands

Several general and DSL-specific **show** commands can help you verify proper operations on the NI-2. The first command is probably the most common Cisco IOS Software command. It produces output that is valuable for DSL and ATM troubleshooting. The other commands are more specific to DSL and/or ATM. These **show** commands are as follows:

- **show running-configuration**
- **show atm vc** (traffic and other options)
- **show hardware** (slot and chassis)
- **show dsl profile**
- **show facility-alarm status**
- **show environment**
- **show network-clocks**
- **show ATM Status**

Each of these is detailed in the following list. It includes the pertinent information you need to monitor DSL activity through the NI-2 card on the Cisco 6000 series of IP DSL Switches:

- **show running-configuration**

  This command tells you which ports are attached to each profile. You can obtain more-specific reporting for selected profiles by using the **show dsl profile** command.

  The following example shows the running configuration:

```
alpha_c6260#show running-config
Building configuration...
Current configuration:
!
! Last configuration change at 12:58:27 EDT Fri June 14 2002
! NVRAM config last updated at 14:13:58 EDT Thu June 13 2002
!
version 12.2 !Notice the more generic version designator!
no service pad
service timestamps debug uptime
```

```
service timestamps log uptime
no service password-encryption
service internal
!
hostname alpha_c6260
!
boot system flash:NI-2-dsl-mz.12-2.DA
slot 1 atuc-1-4dmt !These are the line cards, all four-port DMT cards in
  this chassis; remember that slots 10 and 11 are reserved for the NI-2
  card(s) and are ignored here!
slot 2 atuc-1-4dmt
slot 3 atuc-1-4dmt
slot 4 atuc-1-4dmt
slot 5 atuc-1-4dmt
slot 6 atuc-1-4dmt
slot 7 atuc-1-4dmt
slot 8 atuc-1-4dmt
slot 9 atuc-1-4dmt
slot 12 atuc-1-4dmt

no logging monitor
enable password lab
!
dsl-profile default !As with almost all Cisco show commands, if the default
  values have not been changed, no details are shown for any default
  parameters, such as this default profile. You can display the precise
  default values with the show dsl profile command.!
!
dsl-profile ALPHA_USERS
 dmt interleaving-delay downstream 3 upstream 6
 dmt bitrate maximum interleaved downstream 2048 upstream 256
!
dsl-profile test !This profile has not been defined yet beyond its name, so
  no values are shown.!
!
dsl-profile alpha_677
 dmt interleaving-delay downstream 0 upstream 0
 dmt bitrate maximum interleaved downstream 8032 upstream 864
!
dsl-profile alpha-677
 dmt bitrate maximum interleaved downstream 8032 upstream 864
!
dsl-profile jurgen
 dmt bitrate maximum interleaved downstream 8032 upstream 864
dsl-profile train
network-clock-select revertive !This clocking selection has been changed
  from the default value of non-revertive, which is much more typical in
  service provider environments.!
network-clock-select 1 ATM0/1
network-clock-select 2 system
ip subnet-zero
ip host-routing
ip host zeppelin 1.0.0.253
ip domain-name cisco.com
ip name-server 171.69.204.11
!
atm address 47.0091.8100.0000.0077.d0fe.4301.0077.d0fe.4301.00
atm address 47.0091.8100.0000.00e0.b0ff.b501.00e0.b0ff.b501.00
atm address 47.0091.8100.0000.0050.0fff.cc01.0050.0fff.cc01.00
atm router pnni
 no aesa embedded-number left-justified
 node 1 level 56 lowest
  redistribute atm-static
!
```

```
clock timezone EDT -5
clock summer-time EDT recurring
!
process-max-time 200
!
interface ATM0/0 !This is the backplane trunk, which you can think of as
  the backplane itself; unless you want to configure a PVC for in-band
  management, you need not configure this trunk at all, leaving the default
  values as they are.!
 no ip address

 no ip mroute-cache
 atm cac service-category abr deny
 atm maxvp-number 0
!
interface Ethernet0/0
 ip address 171.69.204.250 255.255.255.0

 no ip proxy-arp
 no ip mroute-cache
 no keepalive
!
interface ATM0/1
 no ip address

 no atm ilmi-keepalive
 atm cac service-category abr deny
 atm manual-well-known-vc
 atm pvc 0 5 pd on rx-cttr 3 tx-cttr 3  interface  ATM0/0 0 any-vci  encap
  qsaal
 atm pvc 0 16 pd on rx-cttr 3 tx-cttr 3  interface  ATM0/0 0 any-vci  encap
  ilmi
 atm pvc 0 18 pd on rx-cttr 3 tx-cttr 3  interface  ATM0/0 0 any-vci  encap
  pnni
 clock source loop-timed
!
interface ATM0/2
 no ip address

 shutdown !Shutdown is the default state of all interfaces, and in this
  case, the subtend trunk ATM0/2 is left shut down because there is no
  subtending on this chassis.!
no atm ilmi-keepalive
 atm cac service-category abr deny
!
interface ATM1/1
 no ip address

 dsl subscriber RayBudge
 dsl profile alpha_676
 no atm ilmi-keepalive
 atm cac service-category abr deny
 atm pvc 0 35  interface  ATM0/1 0 101 !Here is the PVC that is designated
  0/35 as it enters from the DSL subscriber side and is designated 0/101
  as it leaves on the network trunk ATM 0/1.!
```

- **show atm vc**

  The details of this command and its output are explained after the output:

```
6160-93#sho atm vc

Interface  VPI VCI Type  X-Interface X-VPI X-VCI Encap Status
```

```
ATM0/0      0    35 PVC   ATM0/1       0       16   ILMI       UP
ATM0/0      0    36 PVC   ATM0/2       0       16   ILMI     DOWN
ATM0/0      0    37 PVC   ATM0/1       0        5   QSAAL      UP
ATM0/0      0    38 PVC   ATM0/2       0        5   QSAAL    DOWN
ATM0/0      0    39 PVC   ATM0/1       0       18    PNNI      UP
ATM0/1      0     5 PVC   ATM0/0       0       37   QSAAL      UP

ATM0/1      1    34 PVC   ATM1/1       1        1              UP
ATM0/1      1    36 PVC   ATM1/2       1        1              UP
ATM0/1      1    44 PVC   ATM2/2       1        1            DOWN

ATM1/1      1     1 PVC   ATM0/1       1       34              UP
ATM1/2      1     1 PVC   ATM0/1       1       36              UP
ATM2/2      1     1 PVC   ATM0/1       1       44            DOWN
```

This output displays statistics for all PVCs, both manually created and those that were autocreated by the system for ATM signaling and management. All ATM interfaces on the IP DSL Switch are reflected, although you can specify a particular interface's connections to display, such as a particular DSL port. The Status field is either UP or DOWN.

ATM VCs are shown twice, once on each interface. For instance, when you look a bit more than halfway down the second and third columns, VCC 1/34 is first shown on the network trunk interface ATM0/1, mapped to the Cisco 6400 from the line card interface ATM1/1, where it is connected to the CPE modem as 1/1. Then the line card interface itself is shown further down, as you see in the first column, mapping 1/1 to the interface ATM0/1.

To display all ATM virtual circuits (PVCs, soft-PVCs, and SVCs) and traffic information, you can use the **show atm vc** command. You can also use the command **show atm vc | interface** *interface-number,* where **interface** *interface-number* specifies the interface number or subinterface number of the PVC or SVC. This displays all VCs on the specified interface or subinterface.

In this example, ATM 0/2 is shut down, because subtending is not being used. Therefore, the signaling and OAM protocols in the reserved circuits are also down, such as PVC 0/36 in the second line of the list.

The following is an example of a more-specific display, using optional parameters where traffic displays the virtual channel cell traffic:

```
NI-2#show atm vc traffic int atm
Interface   VPI    VCI    Type   rx-cell-cnts    tx-cell-cnts
ATM0/0        0     35       PVC           43              38
ATM0/0        0     36       PVC            0               0
ATM0/0        0     37       PVC           27              29
ATM0/0        0     38       PVC            0               0
ATM0/0        0     39       PVC           64             144
ATM0/1        0      5       PVC           29              27
ATM0/1        0     16       PVC           38              43
ATM0/1        0     18       PVC          144              64
ATM0/1        0    100       PVC            0               0
ATM0/2        0      5       PVC            0               0
ATM0/2        0     16       PVC            0               0
ATM19/1       0     33       PVC            0               0
```

You can specify a particular interface to show the connections on that interface only. You can also use the command **show atm vp** if this is a subtended host with PVPs passing through it from the subtended system.

- **show hardware status**

  You can display information about the chassis type and the physical cards in the chassis and determine whether the power supply and fan modules are present:

```
NI-2#show hardware (Displays all hardware, as shown here:)
Chassis Type: C6260
Slot 1 : ATUC-1-4DMT    Slot 17: ATUC-1-4DMT
Slot 2 : ATUC-1-4DMT    Slot 18: ATUC-1-4DMT
Slot 3 : ATUC-1-4DMT    Slot 19: ATUC-1-4DMT
Slot 4 : ATUC-1-4DMT    Slot 20: ATUC-1-4DMT
Slot 5 : ATUC-1-4DMT    Slot 21: ATUC-1-4DMT
Slot 6 : ATUC-1-4DMT    Slot 22: ATUC-1-4DMT
Slot 7 : ATUC-1-4DMT    Slot 23: ATUC-1-4DMT
Slot 8 : ATUC-1-4DMT    Slot 24: ATUC-1-4DMT
Slot 9 : ATUC-1-4DMT    Slot 25: ATUC-1-4DMT
Slot 10: NI-2-155SM-155SM     Slot 26: ATUC-1-4DMT
Slot 11: EMPTY  Slot 27: ATUC-1-4DMT
Slot 12: ATUC-1-4DMT    Slot 28: ATUC-1-4DMT
Slot 13: ATUC-1-4DMT    Slot 29: ATUC-1-4DMT
Slot 14: ATUC-1-4DMT    Slot 30: ATUC-1-4DMT
Slot 15: ATUC-1-4DMT    Slot 31: ATUC-1-4DMT
Slot 16: ATUC-1-4DMT    Slot 32: ATUC-1-4DMT
Fan Module 1: Present   2: Present
Power Supply Module 1:  Not Present   2:  Present
```

  The **show hardware chassis** command shows the manufacturing data for the NI-2 motherboard and daughter card, I/O controller, power module, and backplane, plus the chassis type, chassis name, manufacturer's name, hardware revision, serial number, asset ID, alias, and CLEI code.

- **show dsl profile**

  This command displays all profiles unless you use the option *profile-name* at the end of the command to specify a particular profile.

  The following example displays the command profile named ALPHA_USERS:

```
NI-2#show dsl profile ALPHA_USERS

dsl profile ALPHA_USERS:
Alarms Enabled: NO
ATM Payload Scrambling: Enabled

DMT profile parameters
    Maximum Bitrates:
Interleave Path: downstream: 8032/kbs, upstream: 864/kbs
    Minimum Bitrates:
Interleave Path: downstream: 0/kbs, upstream: 0/kbs
    Margin: downstream: 3 db, upstream: 3 db
    Interleave Delay: downstream: 0 usecs, upstream: 0 usecs
    Check Bytes (FEC):
        Interleave Path: downstream: 16, upstream: 16
    R-S Codeword Size: downstream: auto, upstream: auto
    Trellis Coding: Disabled
```

```
    Overhead Framing: Mode 1
    Bit-Swap:     Enabled
    Bit-Swap From Margin:     3 dB
    Bit-Swap To Margin:     3 dB     Operating Mode:     Automatic
     Training Mode:        Standard

SDSL profile parameters

CAP profile parameters
```

The last two lines, for SDSL profile parameters and CAP profile parameters, are legacy provisions for these obsolescent modulations. In this case, no profile named ALPHA_USERS has been defined for SDSL or CAP, so these display areas are blank.

- **show facility-alarm status**

  This command shows the current major and minor alarms and the thresholds for all user-configurable alarms on a Cisco IP DSL Switch.

  The following are different examples of the output. The first example shows a single major alarm and a single informational notice:

```
NI-2#show facility-alarm status
System Totals  Critical: 0  Major: 1  Minor: 0
Source: Fan Slot 0   Severity: MAJOR  Description: 1  Not detected or missing
Source: Slot 19      Severity: INFO   Description: 4  Module was detected
```

  The next example of the **show facility-alarm status** command shows one critical alarm, one major alarm, and one informational notice:

```
NI-2#show facility-alarm stat
System Totals  Critical: 1  Major: 1  Minor: 0
Source: NI-2 Module Severity: MAJOR  Description: 1  Loss of active clock sync
Source: Slot 19    Severity: INFO   Description: 4  Module was detected
Source: ATM0/1     Severity: CRITICAL Description: 6 Line RDI
```

- **show environment**

  This command displays temperature, voltage, and chassis status information. The **show environment** command has two valuable options—**all** and **table**.

  all lists temperature readings, fan status, and chassis status. **table** displays the temperature and voltage thresholds and lists the ranges of environmental measurements that are within the specified ranges.

  Here is an example:

```
NI-2#show environment all
                    Slot 1     Slot 2
Power/Fan Presence:
       Power Module:  No          Yes
           Fan Tray:  Yes         Yes

Power Modules:
     48 VDC voltage:  0  volts    48 volts
     48 VDC current:  0  amps     1  amps
     24 VDC thresh.:  0  volts    20 volts
        Power Fault:  No          No

Fans:
```

```
       Fan Number 0:  on          on
       Fan Number 1:  on          on
       Fan Number 2:  on          on
       Fan Number 3:  on          on

Temperature readings:
          NI-2 inlet: 21C/69F
          NI-2 outlet: 27C/80F
   Slot 1 PM internal: 0C/32F
   Slot 1 PM external: 0C/32F
   Slot 2 PM internal: 23C/73F
   Slot 2 PM external: 15C/59F
```

- **show network-clocks**

  You can see which ports are designated as network clock sources with this command. For example:

```
NI-2#show network-clocks
PLL failed: 42; PLL Passed: 2741 !(PLL stands for Phase Locked Loop)!
FAIL: 0; NCO: E391; REF: E390; ERR: 1; ERR_D: -1; MAG: 2;
clock configuration is NON-Revertive
Priority 1 clock source: ATM0/1
Priority 2 clock source: No clock
Priority 3 clock source: No clock
Priority 4 clock source: No clock
Priority 5 clock source: System clock
Current clock source:System clock, priority:5
Nettime Config Register Contents:
SLOCK:0, TLOCK:0, NFAIL:0, E1:1, NSEL:0
BITS Register Contents:
CR1: C8, CR2: 0, CR3: 0, ICR: 0, TSR: C1, PSR: 11, ESR: 77, CR4: 0
BITS Source configured as: E1 Short Haul, ITU G.703 pulse, 120 ohm TP/75
  ohm Coax, 12 db gain
```

- **Confirming the interface status**

  You can use the **show atm status** command to confirm the status of ATM interfaces.

  For example:

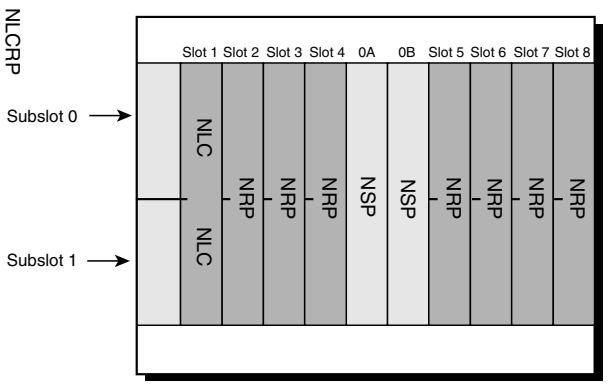
```
NI-2#show atm status
NUMBER OF INSTALLED CONNECTIONS: (P2P=Point to Point, P2MP=Point to
  MultiPoint, MP2P=Multipoint to Point)
Type   PVCs SoftPVCs SVCs TVCs   PVPs SoftPVPs SVPs    Total
P2P      1     0       0     0     0     0     0           1
P2MP     0     0       0     0     0     0     0           0
MP2P     0     0       0     0     0     0     0           0

TOTAL INSTALLED CONNECTIONS =          1
PER-INTERFACE STATUS SUMMARY AT 10:27:54 EDT Thu Jun 10 2002:
Interface IF Admin  Automation :-Cfg ILMI Addr SSCOP Hello
Name  Status Status  Status  Reg State  State State
------------ ------- ------------ -------- ---------
ATM0/0   UP    up      n/a  UpAndNormal  Idle   n/a
ATM0/1   UP    up      n/a    n/a       Idle   down
ATM0/2  DOWN  down  waiting   n/a       Idle    n/a
```

Aggregator/Concentrator: Cisco 6400

The 6400 UAC provides both ATM switching and Layer 3 IP routing for the ATM signals coming from the DSL network. This section explains the 6400 configuration for a variety of scenarios, starting with redundancy commands and ending with sophisticated routing topologies.

You might want to review the 6400 hardware components that were discussed in Chapter 4 to refresh your memory and to help you better understand the Cisco 6400 ATM interfaces shown in Figure 6-10.

Figure 6-10 *Cisco 6400 ATM Interfaces Through the Node Switch Processor*



- Interface ATM0/0/0 Interface connected to backplane (not used for data traffic)
- Interface ATM1/0/1 NLC in slot 1, subslot 0, port 1
- Interface ATM5/0/0 NRP in slot 5
- Interface ATM6/0/0 NRP in slot 6

Bear in mind that there are eight full-height slots into which you can fit any combination of half-height node line cards (NLCs), full-height NLCs (the OC-12 variant), and/or full-height second-generation node route processors (NRP-2s).

The upper, half-height NLC is identified as subslot 0, as is the entire full-height NLC. A half-height NLC in the lower subslot is identified as occupying subslot 1. The first port on the half-height NLCs, and the only port on the full-height NLCs (OC-12/STM4), are designated as port 0.

The 6400 control is through the full-height, optionally redundant node switch processor (NSP), centered on the 6400 in a dedicated slot. Therefore, all interfaces are referenced from the NSP's point of view. The NSP is the only avenue to configure the NRP-2 and NLC. All 6400 cards are connected via the ATM backplane to the NSP. This interface is known as ATM INT 0/0/0 and can be thought of as the interface to the NSP, which itself is the interface to the entire 6400 device.

NRP-2

This section describes basic setup and initial configuration for the second-generation NRP blade on the Cisco 6400 (NRP-2), as well as redundancy options for all modules.

Cisco 6400 Redundancy Configurations

All the modules on the Cisco 6400 are optionally redundant. This section explains how to configure the 6400 for redundancy.

Memory Considerations for Redundancy

When you configure redundancy between two NRPs or two NSPs, the two cards must have identical memory capabilities and hardware specifications. Check each card in a redundant pair, and make sure they share the following parameters:

- DRAM size
- Flash memory size
- PCMCIA disk size (NSP only)
- Hardware version (module part number)

If redundancy is configured between two cards with different amounts of memory or disk capacity, the Cisco 6400 displays a warning message. Depending on which card is identified as the primary card, the Cisco 6400 performs the following actions:

- If the primary card has more memory than the secondary card, the Cisco 6400 shuts down the secondary card.
- If the secondary card has more memory than the primary card, the Cisco 6400 displays a message indicating that the secondary card has more memory than the primary card. This configuration causes redundancy to be disabled if the secondary card is activated.

Redundant NSPs

Both NSP slots are numbered slot 0 for consistent interface identification between primary and secondary devices. Nevertheless, the left NSP slot is labeled slot A and the right slot is labeled slot B to distinguish between the two slots when required. You do not need to explicitly specify redundancy between NSPs using slot identification. If two NSPs are installed in the Cisco 6400, they automatically act as a redundant pair.

You can use Enhanced High System Availability (EHSA) redundancy for simple hardware backup or for software error protection. Hardware backup protects against NSP card failure, because you configure both NSP cards with the same software image and configuration information. Additionally, you configure the system to automatically synchronize configuration information on both cards when changes occur.

Software error protection protects against critical Cisco IOS Software errors in a particular release, because you configure the NSP cards with different software images but use the same configuration. If you are using new or experimental Cisco IOS Software, consider using the software error protection method.

After you have installed the second NSP, you can verify NSP redundancy with the **show redundancy** command (the results might vary slightly with the Cisco IOS Software version):

```
Switch#show redundancy
!
NSP A:Primary
NSP B:Secondary
!
Secondary NSP information:
Secondary is up
Secondary has 131072K bytes of memory.
!
User EHSA configuration (by CLI config):
secondary-console = off
keepalive     = on
config-sync modes:
  standard   = on
  start-up   = on
  boot-var   = on
  config-reg = on
  calendar   = on
!
Debug EHSA Information:
!
Primary   (NSP A) ehsa state:SANTA_EHSA_PRIMARY
Secondary (NSP B) ehsa state:SANTA_EHSA_SECONDARY
!
EHSA pins:
peer present = 1
peer state   = SANTA_EHSA_SECONDARY
crash status:this-nsp=NO_CRASH(0x1) peer-nsp=NO_CRASH(0x1)
!
EHSA related MAC addresses:
this bpe mac-addr = 0000.0c00.0003
peer bpe mac-addr = 0000.0c00.0004
!
Switch#
```

To ensure that the configuration is consistent between redundant NSPs or NRPs, you can configure automatic synchronization between the two devices. You have the option of synchronizing just the startup configuration, the boot variables, the configuration register, or all three configurations. Boot variables are ROM monitor (ROMMON) environment variables used to control the booting process. The configuration register, stored in NVRAM, contains startup time parameters for the system. For more information about the booting process, see the *Cisco IOS Configuration Fundamentals Configuration Guide*. Frankly, though, you might as well use the default standard synchronization unless you have good reason to do otherwise.

After the configuration is complete, you can disable autoconfiguration using the **no** command. The default setting for individual synchronizable options is **no auto-sync**.

The following example shows how to synchronize the configurations on two redundant NSPs:

```
Switch#config term
Switch(config)#redundancy
Switch(config-r)#main-cpu
Switch(config-r-mc)#auto-sync standard
Switch(config-r-mc)#end
Switch#
```

NRP-1 Redundancy

Redundant first-generation NRPs use EHSA signals. These signals let the two NRPs negotiate which is the master and which is the secondary. After the NRPs determine which is the primary, they communicate that information to the NSP. The NSP then communicates with that specific NRP.

Configuring an NRP pair for redundancy starts with configuring the NSP, which is the controlling module for the Cisco 6400 UAC. Therefore, there are two sets of commands for NRP-1 redundancy:

- Configure NRP-1 redundancy on the NSP:

```
Switch(config)#redundancy
Switch(config-r)#main-cpu
Switch(config-r-mc)#associate slot 1 2 (You need specify only the first slot of
  the redundant pair, because redundant pairs must be adjacent. The second slot
  is assumed to be the adjacent slot. Specifying both slots is not harmful,
  though.)
Switch(config-r-a-sl)#prefer 1
```

- Configure auto-sync on the primary NRP-1:

```
Router(config)# redundancy
Router(config-r)# main-cpu
Router(config-r-mc)# auto-sync standard
```

NSP PCMCIA Disk Protection for NRP-2

NRP-2s do not support redundancy. However, an important aspect of NSP functionality affects the NRP-2 in the case of NSP failover. Bear in mind that the NRP-2 is controlled and configured entirely through the NSP, without direct access to the NRP-2 itself, and the NRP-2 depends on the NSP for image and file storage. The NRP-2 operation is ensured during switching from a failed NSP to a working NSP through disk mirroring of the PCMCIA disks on the redundant NSPs. Introduced in Cisco IOS Software Release 12.1(5)DB, PCMCIA disk mirroring enables automatic data synchronization between the PCMCIA disks of the two redundant NSPs. Disk synchronization is the act of copying data from one disk to another.

Without disk mirroring, there is no guarantee of NRP-2 support after an NSP failover. This means that you might have to manually restore the NRP-2 state to its status before the failover. With disk mirroring enabled, NRP-2 has continued support from the NSP, except during the relatively short NSP failover period.

When PCMCIA disk mirroring is enabled, as it is by default, disk synchronization is initiated in any of the following situations:

- The primary or secondary NSP boots or reloads
- The secondary NSP is inserted into the Cisco 6400 chassis
- A PCMCIA disk is inserted into disk slot 0 of the primary or secondary NSP
- The PCMCIA disk in disk slot 0 of either NSP is formatted
- A command is entered to
  - Re-enable disk mirroring (**mirror**)
  - Explicitly initiate disk synchronization (**redundancy sync**)
  - Modify or reorganize the files on the disks (**copy**, **rename**, **delete**, **mkdir**, **format**)

Cisco recommends that you use PCMCIA disks that have the same memory capacity.

PCMCIA disk mirroring is not supported in Cisco IOS Software Release 12.1(4)DB and earlier releases. Use the **dir**, **mkdir**, and **copy** EXEC commands to manually copy files from the primary NSP's PCMCIA disks to the secondary NSP's PCMCIA disks.

PCMCIA disk mirroring also introduced new labels for pairs of mirrored disks:

- **mir-disk0**—PCMCIA disks in disk slot 0 of both NSPs
- **mir-disk1**—PCMCIA disks in disk slot 1 of both NSPs

The mir-disk0 and mir-disk1 labels let you perform any integrated file system (IFS) operation (such as copy, rename, or delete) on the same file on both the primary and secondary disks.

Disk mirroring (automatic data synchronization between a pair of disks) is not supported between the following:

- Two disks on a single NSP
- Two disks with mismatched slot numbers (disk0: and disk1:)

You can initiate disk synchronization between disk0: and disk1: on the active NSP, even in a single-NSP system.

There are other uses for the PCMCIA disks aside from providing NRP-2 configuration redundancy. For instance, you can make full image and configuration backups if the disks are of sufficient size. Visit the appropriate online Cisco documentation for your version of Cisco IOS Software for other tips about these versatile disks.

NLC Redundancy and SONET APS

SONET APS provides a mechanism to support redundant transmission interfaces (circuits) between SONET devices. Automatic switchover from the working (primary) circuit to the protection (secondary) circuit happens when the working circuit fails or degrades.

The Cisco 6400 supports the following SONET APS operations:

- **1+1**—There is one working interface and one protection interface, and the payload from the transmitting end is sent to both the receiving ends. The receiving end decides which interface to use. The line overhead (LOH) bytes (K1 and K2) in the SONET frame indicate both status and action.
- **Linear**—A back-to-back connection (as opposed to a ring topology), as defined in the "Telcordia GR-253-CORE" document.
- **Unidirectional**—Transmit and receive channels are switched independently.
- **Nonreverting**—Nonreverting channels continue to operate after a failure has been corrected, thus preventing data from flowing back to the working channel.
- **Enabling and disabling SONET APS**—In the Cisco 6400, a pair of redundant ports is represented as a single interface. APS commands are accepted only for an interface that represents a pair of redundant ports.

For APS operation, the APS mode must be specified for each interface associated with a redundant pair of ports. To enable SONET APS, use these commands, beginning in global configuration mode:

```
Switch(config)#interface atm slot/subslot/port
```

You can use either NLC port number, 0 or 1. When an NLC is configured for redundancy, all ports on that card are automatically configured to operate in redundant mode using SONET APS.

```
Switch(config-if)#aps mode linear 1+1 nonreverting unidirectional
```

This command enables SONET APS on the interface. This command must be entered before any other **aps** commands. SONET APS is enabled by default when you install an NLC in a slot already configured for redundancy. Here is an example of a configuration with redundancy:

```
redundancy
 associate slot 1 2
!
interface ATM1/0/0
 no ip address
 no ip redirects
 no ip proxy-arp
 no atm auto-configuration
 no atm ilmi-keepalive
 atm uni version 4.0
 aps mode linear 1+1 nonreverting unidirectional
 aps signal-fail BER threshold 3
!
```

If you disable the redundant NLC configuration by using the **no associate slot** or **no associate subslot** redundancy configuration command, two interface configuration sections are created, one for each port, but all the APS configuration commands are removed. Here is an example of adjacent NLCs that are operating independently, not redundantly:

```
interface ATM1/0/0
 no ip address
 no ip redirects
 no ip proxy-arp
 no atm auto-configuration
 no atm ilmi-keepalive
 atm uni version 4.0
!
interface ATM2/0/0
 no ip address
 no ip redirects
 no ip proxy-arp
 no atm auto-configuration
 no atm ilmi-keepalive
 atm uni version 4.0
!
```

For two full-height (OC-12/STM4) NLCs to act as a redundant pair, they must be installed in adjacent slots, such as slots 1 and 2. By default, the NLC in the lower-numbered slot is the working device, and the NLC in the higher-numbered slot is the protection device.

To configure redundant full-height NLCs, use the **redundancy** and **associate slot** commands, as in the following example, where the OC-12s in slots 5 and 6 are configured for redundancy:

```
!
redundancy
 associate slot 5 6
!
```

For two half-height NLCs to act as a redundant pair, they must be installed in adjacent slot/subslot pairs. Here are some examples:

1/0 and 2/0, or 1/1 and 2/1

3/0 and 4/0, or 3/1 and 4/1

5/0 and 6/0, or 5/1 and 6/1

7/0 and 8/0, or 7/1 and 8/1

To configure redundant half-height NLCs, use the **redundancy** command as you have been doing, and use a variant of the **associate slot** command, **associate subslot**. In the following example, the OC-3s in subslots 3/0 and 4/0 are configured as a redundant pair:

```
!
redundancy
 associate subslot 3/0 4/0
!
```

Verifying NLC Redundancy

To verify NLC redundancy, use the **show aps** EXEC command on the NSP. The **show aps** command displays the status for all NLCs configured for port redundancy:

```
Switch#show aps

ATM7/0/0: APS Lin NR Uni, Failure channel: Protection
          Active Channel: CHANNEL7/0/0, Channel stat: Good
          Port stat (w,p): (Good, Good)
ATM7/0/1: APS Lin NR Uni, Failure channel: Protection
          Active Channel: CHANNEL7/0/1, Channel stat: Good
          Port stat (w,p): (Good, Good)
```

Verifying SONET APS

To verify that SONET APS is enabled, or to determine if a switchover has occurred, use the **show aps** EXEC command or the **show controller atm** *slot/subslot/port* command.

In the following example, slot 7 contains the working (primary) card, and slot 8 contains the protection (secondary) card:

```
Switch#show aps

ATM7/0/0: APS Lin NR Uni, Failure channel: Protection
          Active Channel: CHANNEL7/0/0, Channel stat: Good
          Port stat (w,p): (Good, Good)
ATM7/0/1: APS Lin NR Uni, Failure channel: Protection
          Active Channel: CHANNEL7/0/1, Channel stat: Good
          Port stat (w,p): (Good, Good)
```

In the following example, the OC-3 interface ATM 5/0/0 is not configured for redundancy:

```
Switch#show controller atm 5/0/0
Redundancy NOT Enabled on interface
IF Name: ATM5/0/0    Chip Base Address(es): A8B08000, 0 Port type: OC3
  Port rate: 155
Mbps    Port medium: SM Fiber
Port status:Good Signal    Loopback:None    Flags:8308
TX Led: Traffic Pattern    RX Led: Traffic Pattern  TX clock source:
  network-derived
Framing mode:  sts-3c
Cell payload scrambling on
Sts-stream scrambling on
```

Setting SONET APS Priority Requests: Preventing or Causing Automatic Protection Switching

APS priority requests are used to manually control the relationship between two APS ports from EXEC mode. The APS priority levels, lockout (1), force (2), and manual (5), are defined in the "*Telcordia GR-253-CORE"* document.

To set the APS priority requests, use the following commands in EXEC mode:

```
Switch#aps lockout atm slot/subslot/port
```

This APS priority level 1 request prevents a working interface from switching to a protection interface.

```
Switch#aps force atm slot/subslot/port from [protection | working]
```

This APS priority level 2 request manually forces the specified interface to the protection or working interface unless a request of equal or higher priority is in effect.

Use the **working** option to force operation from the working channel to the protection channel. Use the **protection** option to force operation from the protection channel to the working channel. For instance, in the following example, the system is forced to use the protection channel associated with ATM interface 1/0/0:

```
Switch#aps force atm 1/0/0 from working
```

The following APS priority level 5 request manually switches an interface to the protection or working interface unless a request of equal or higher priority is in effect. The **working** and **protection** options are available for this command as well:

```
Switch#aps manual atm slot/subslot/port from [protection | working]
```

The **Switch#aps clear atm** *slot/subslot/port* command manually clears all posted APS priority requests created by any of the APS priority commands.

To verify that you successfully set the APS priority requests, you can use the **show aps** EXEC command:

```
Switch#aps force atm 5/1/0 from working
Switch#show aps
      ATM5/1/0:APS Lin NR Uni, Failure channel:Working
      Active Channel:CHANNEL6/1/0, Channel stat:Force Switch
      Port stat (w,p):(Good, Good)
```

Setting SONET APS Signal Thresholds

You can configure the APS signal bit error rate (BER) thresholds at which the system announces signal degradation or signal failure.

The **Switch(config-if)#aps signal-degrade BER threshold** ***value*** command sets the interface's BER threshold value for signal degradation. This controls the BER value at which signal degradation is announced, indicating an unstable or error-prone connection. This BER threshold can be in the range of 10^{-5} to 10^{-9}. There is no default threshold, although the generally accepted telecom industry standard is 10^{-7}.

The **Switch(config-if)#aps signal-fail BER threshold** ***value*** command sets the interface's BER threshold value for signal failure. This controls the BER value at which a signal failure is announced, indicating a broken connection. This BER threshold can be in the range of 10^{-3} to 10^{-5}, with a default threshold of 10^{-3}.

The *value* argument represents the exponent of the BER threshold. For instance, a value of 5 sets the threshold to 10^{-5}. For example, here is how to set the APS signal degradation and signal failure thresholds for ATM interface 1/0/0:

```
Switch(config)#interface atm 1/0/0
Switch(config-if)#aps signal-degrade BER threshold 7
Switch(config-if)#aps signal-fail BER threshold 5
```

Verifying SONET APS Signal Thresholds

To display an interface's current BER threshold settings, use the **show interface atm** command:

```
Switch#show interface atm 1/0/0

interface ATM1/0/0
 description la1
 no ip address
 no ip redirects
 no ip proxy-arp
 no atm auto-configuration
 no atm ilmi-keepalive
 atm uni version 4.0
 aps mode linear 1+1 nonreverting unidirectional
 aps signal-fail BER threshold 3
 aps signal-degrade BER threshold 9
```

Primary and Secondary Role Switching

The Cisco 6400 allows you to manually force the primary and secondary devices in a redundant pair to switch roles. This capability can be important for upgrade or debug activities.

To reverse the primary and secondary roles in redundant modules, use the following command in EXEC mode:

```
Switch#redundancy force-failover {slot | slot/subslot | main-cpu}
```

This command forces the system to switch the current primary and secondary devices of the redundant pair.

Now that you have seen the redundancy commands, you can begin configuring the 6400 to accept and manage the incoming connections at the Layer 2 and Layer 3 levels.

Configurations for ATM Switching

Here is the most basic pair of commands that direct ATM connections through the Cisco 6400. To map PVC 1/100 coming from int NLC 1/0/0 to NRP 7/0/0, where it would change identification to become PVC 1/101, the 6400 commands would be

```
NSP(config)#int atm 1/0/0
NSP(config-if)#atm pvc 1 100 interface atm7/0/0 1 101
```

NOTE You can also use PVPs to map whole groups of individual PVCs. Just substitute **pvp** for **pvc** and leave off the virtual channel identifier after the path number.

The first step is to address the ATM interface 1/0/0. This particular NLC identifier can relate to any of the following:

- A coaxial, half-height NLC (DS3 or E3) that addresses the BNC connectors on the 6400's backplane coaxial I/O board. The interface identification 1/0/0 would denote the connection in the first slot, the upper subslot, and the first of two ports for that upper subslot.
- A fiber-connected, half-height NLC (OC-3 or STM1) with optical fiber connections on the front of the card. The identifier 1/0/0 would denote the first slot's upper subslot's first port.
- A fiber-connected, full-height NLC (OC-12 or STM4) occupying the full first slot, obviously starting with the upper subslot, with one and only one port for the relatively large optical fiber connection on the front.

After addressing the NLC, the second command maps the VC to the ATM interface on the NRP card in slot 7/0/0 and renumbers the virtual channel identifier to 101 as it will be handled on the NRP. You can think of this mapping as across the 6400's backplane, after the NLC has stepped down the incoming high-speed signal (coaxial or optical).

Configurations for Layer 3 Terminations

After the PVC arrives at the NRP, more sophisticated coding is required to identify and manage the connection. This coding defines the Layer 3 terminations (bridging, PPPoE, PPPoA) as well as security options (AAA, RADIUS, or TACACS+) and address management options such as DHCP. Each of these is explained in the following sections with examples.

PPPoA

The following is the simple, manual configuration for a PPPoA connection on the 6400's NRP:

```
router(config)#username cisco password Cisco
router(config)#interface atm0/0/0.1 point-to-point
router(config-if)#pvc 1/101
router(config-if-atm-vc)#encapsulation aal5mux ppp virtual-template 1
router(config-if-atm-vc)#ubr 384
router(config-if-atm-vc)#exit
```

The **username cisco password Cisco** command identifies the remote ATU-R host name (cisco) and password (Cisco) used for PPP CHAP authentication.

The **interface atm0/0/0.1 point-to-point** command identifies the subinterface (logical division of a physical interface) atm0/0/0 as the target interface for the incoming connection. This subinterface is a point-to-point connection.

The **pvc 1/101** command addresses the connection that will terminate on this subinterface as a PVC, with the newly renumbered identification of VPI=1, VCI=101.

The **encapsulation aal5mux ppp virtual-template 1** command sets the encapsulation method for the PVC 1/101 as AAL5MUX. With AAL5MUX (as opposed to AAL5SNAP), multiple protocols (such as IP and IPX) can be carried inside the PPP frames and terminated on this interface. This command line also establishes that virtual template 1 will be used for additional configuration information for this PVC. You will see shortly the virtual template's labor-saving, commonly-shared parameters.

ubr 384 is a simple command that defines the unspecified bit rate (UBR) PCR as 384 Kbps bidirectionally (both upstream and downstream). The PCR is the only parameter that can be set with UBR, and it is optional. If no PCR is defined, each connection is theoretically free to occupy all the available bandwidth.

The previous example is an example of when a single VC within a subinterface is configured for PPPoA encapsulation (AAL5MUX). Although Template 1 is used to define AAA configuration for this VC, you can see that configuring each individual VC is not the best way to conduct large-scale implementations. Still, you should learn this simplest way so that you can appreciate the shortcuts shown later in this section.

The NRP's virtual templates assign PPP features (or other architecture characteristics) to a PVC. As each PPP session comes online, a virtual-access interface is cloned from the virtual template. This virtual-access interface inherits all the configuration specified in the virtual template. When the virtual template is changed, the changes are automatically propagated to all virtual-access interfaces cloned from that particular virtual template. Here is an example of a configured virtual template:

```
(lines deleted)
 pvc 1/101
  encapsulation aal5mux ppp Virtual-Template1

(lines deleted)

interface virtual-template1
 description PPPoATM
 ip unnumbered gigabit-ethernet0/0/0

 peer default ip address pool dsl
 ppp authentication chap
!
ip local pool dsl 192.168.40.20 192.168.40.50
```

In this configuration, it is assumed that all PPPoA VCs (DSL users) cloned from virtual template 1 will use CHAP authentication and will be allocated an IP address from the pool named dsl configured on the router. When the virtual template is changed, the changes are automatically propagated to all virtual-access interfaces cloned from that particular virtual template. To configure a different class of users on the same router, you can provision a separate virtual template interface. You can have up to 25 virtual templates.

An interesting characteristic of the virtual template is that the local end of the PPPoA connection runs without an explicitly-defined IP address. Instead, the IP address of the NRP-2's Gigabit Ethernet interface is used for addressability. This could also be the Fast Ethernet interface on the first-generation NRP or a loopback interface on either type of NRP. For reasons of memory allocation and addressing beyond the scope of this course, the virtual-access interface must have some sort of interface IP address linked with it syntactically. Therefore, this command essentially says, "Because we *must* link some sort of IP address to this virtual-access interface, but we do not want to waste a real IP address on it, just refer to it as having the same IP address as the already-designated Gigabit Ethernet interface."

In any case, do not use a static IP assignment within a virtual template; routing problems can occur, and you might end up working late if you forget this caution. Always use the **ip unnumbered** command when configuring a virtual template.

Following are explanations of the commands used:

- **interface virtual-template** *number*—Associates a virtual template with a virtual template interface.
- **description PPPoATM**—An optional plain-language description, which could also be the name of the marketed class of service, perhaps similar to "Business-class Gold" or "Residential-class Regular Plan."
- **ip unnumbered gigabit-ethernet 0/0/0**—Enables IP on the interface without assigning a specific IP address.
- **no ip directed-broadcast**—Depending on the Cisco IOS Software version, you might not see this command line, because it is now a default value. It is explained here only in the interest of completeness. This subcommand disables forwarding of directed broadcasts on the interface. The default is to forward directed broadcasts. A directed broadcast is a packet sent to a specific network or series of networks, whereas a flooded broadcast packet is sent to every network. A directed-broadcast address includes the network or subnet fields.
- **peer default ip address {pool [***poolname***] | dhcp }**—Specifies a dynamic IP address assignment method, in this case from a named pool of IP addresses. Another option is to assign addresses from a DHCP server.
- **ppp authentication {pap | chap} [pap | chap]**—Selects the authentication protocol, CHAP in this case, and an optional secondary protocol.
- **ip local pool dsl 192.168.40.20 192.168.40.50**—Defines the range of 31 IP addresses available through the pool named dsl. These addresses may be reused in instant succession as soon as one host device (DSL subscriber) relinquishes an address by logging off the DSL network.

Although instructive, this method is incredibly cumbersome, because you must define each PVC in turn, defining the encapsulation type for each PVC, referring to the same virtual template for each one, and so on. And this very simple example doesn't differentiate between service levels by bit rate, and so on.

How can you reduce the manual configuration for each PVC, saving labor costs and minimizing the chance of human error? You can group characteristics in a class of parameters and then reference the class repeatedly. Here is a configuration that does just that:

```
router(config)#vc-class atm ppp-atm !names the type of class, in this case, an ATM
  Point-to-Point Protocol over ATM class!
router(config-vc-class)#encapsulation aal5mux ppp
virtual-template 1
router(config-vc-class)#ubr 384
router(config-vc-class)#exit


router(config)#interface ATM 0/0/0
router(config-if)#class-int ppp-atm !Associates the new class with the interface
  ATM 0/0/0!
router(config-if)#pvc 1/101
router(config-if-atm-vc)#exit
router(config-if)#pvc 1/102
router(config-if-atm-vc)#exit
```

In this example, a virtual connection class (VC class) called PPP-ATM is defined that specifies how VCs will be encapsulated and secured. This VC class is then associated with the interface ATM0/0/0 on the NRP. Remember that ATM0/0/0 is the NRP's main (backplane) connection on the 6400. By associating the VC class with the interface, every individual PVC that is identified on that interface automatically takes on the characteristics of that class. This automation eliminates the need to repeat the characteristics line by line for each PVC.

This virtual class refers to the baseline virtual template, constituting a reference within a reference, meaning that when these VCs are assigned the values of the virtual class, they also inherit all the values of the virtual template. For instance, you already saw that the virtual template uses the PPP protocol CHAP to provide for AAA (Authentication, Authorization, and Accounting). This eliminates typing the same CHAP definitions for every PVC.

The VC class can be applied to an entire interface or subinterface, but there is still flexibility for different service levels. Even if the VC class is associated with an interface, an individual PVC on that same interface can be associated with a different VC class without affecting the overall association of the interface class for other PVCs.

It should be obvious that the VC class method of user configuration is much more efficient than repeated, individual VC configuration for large-scale service deployment. However, this method still requires manually identifying individual PVCs on a particular interface, even if no further definition is needed. Therefore, this is still a labor-intensive configuration to accommodate the tens of thousands of PVCs in a large DSL network. In the interest of working smarter, not harder, there should be a still-easier way. This easier way makes use of autodiscovery, as explained next.

This method does not require any individual definition of PVCs after they arrive on the NRP. With careful planning and precise advance configuration, the incoming PVCs are automatically switched to certain interfaces, according to the particular service levels, where the associated

VC class has been associated on that interface. Here is a simple example of the basic commands to establish this autodiscovery and routing:

```
router(config)#vc-class atm ppp-atm
router(config-vc-class)#encapsulation aal5mux ppp
virtual-template 1
router(config-vc-class) #exit

router(config)#interface ATM 0/0/0
router(config-if)#atm ilmi-enable
router(config-if)#atm ilmi-pvc-discovery sub-interface
router(config-if)#interface ATM 0/0/0.101
router(config-if)#class-int ppp-atm

router(config-if) #exit
```

Here are the steps required for this most efficient method of handling PVCs, in their suggested order of implementation:

Step 1 The individual PVCs must be configured with ordered VPIs that correspond to the individualized service levels. This can be done at the IP DSL Switch or when mapping the PVCs across the 6400 backplane. For instance, a very robust and well-differentiated marketing plan for the DSL service provider might offer six DSL service levels: regular, fast, and fastest for both business and residential customers. Therefore, the corresponding VPIs could be numbered 101, 102, and 103 for business-class customers, and the VPIs of the residential-class customers could be numbered 201, 202, and 203. As mentioned in Appendix B, the Network-Network Interface (NNI) topology has 4096 available VPI numbers.

Step 2 You must create NRP subinterfaces that correspond to the VPI numbers. In this example, these subinterfaces would be designated as follows:

int ATM 0/0/0.101	Regular business-class service
int ATM 0/0/0.102	Fast business-class service
int ATM 0/0/0.103	Fastest business-class service
int ATM 0/0/0.201	Regular residential-class service
int ATM 0/0/0.202	Fast residential-class service
int ATM 0/0/0.203	Fastest residential-class service

Step 3 In this example, you must define six virtual classes of service, each of which might reference as few as a single virtual template, as explained earlier. For simplicity, you could name each ATM VC class according to the marketed service level.

Step 4 Following this scenario, you would associate each of the virtual classes with the matching subinterface using the **int-class** command. This means that each of the six subinterfaces would have associated with it a particular VC class, with parameters varying by the end user's subscription.

Step 5 The fifth and suggested last step is to enable ILMI and **ilmi-pvc-discovery**. ILMI is one of ATM's internal signaling and control protocols.

As PVCs arrive on the NRP's main interface, int ATM0/0/0, they are discovered by the ILMI protocol. The incoming PVC's VPI is identified and matched with a corresponding subinterface that was configured in advance. In the present scenario with six different service levels and six different subinterfaces, the PVCs whose VPI is 101 would automatically be switched to subinterface 0/0/0.101, where they would be assigned the parameters that were associated with that subinterface through the VC class. The PVCs whose VPI is 102 would automatically be funneled to the subinterface 0/0/0.102, where the group characteristics would include a medium-fast peak cell rate. The PVCs from the business-class customers who had paid for the fastest service available, those whose PVCs came into the NRP with a VPI of 103, would automatically be steered by ILMI to the subinterface of 0/0/0.103. The continuous process of autodiscovery by ILMI and the assignment of the appropriate class characteristics is simultaneous, meaning that the residential customers' PVCs are also switched to their appropriate target subinterfaces, there to assume their own characteristics.

Consider this great news: You are not required to specify any individual PVC definitions, such as encapsulation and bit rate, for these tens of thousands of incoming connections! Of course, you should carefully define the VC classes and subinterfaces in advance and then test the ILMI discovery on a few sample PVCs in advance.

As many subinterfaces may be set up as there are VPIs. For instance, all incoming PVCs whose VPI is 2 would be routed automatically to subinterface 0/0/0.2, all incoming PVCs whose VPI is 3 would be routed automatically to subinterface 0/0/0.3, and so on. Each of these subinterfaces would presumably have a unique virtual class associated with it. However, in conformance with the good practice of a limited number of profiles on the IP DSL Switch, you should not need more than a dozen different service levels, and therefore, no more than a dozen different subinterfaces. This minimizes, to almost nothing, excessive ATM overhead and processing demands that could degrade performance if you had an unrealistically high number of subinterfaces, such as a different subinterface (and service level) for every PVC.

Now that you have configured the ultra-efficient autodiscovery of PVCs on the NRP, you must know how to verify the ATM traffic's status. You can do this with the command **show atm pvc**

ppp, which precisely displays the status of the ATM PVCs of the PPP type. Here is an example of this command's output:

```
NRP-1-8#sho atm pvc ppp
         VCD /
ATM Int.  Name VPI VCI Type  VCSt VA  VASt
0/0/0     17   1   34  PVC   UP   13  DOWN UBR 155000  UP
0/0/0     18   1   36  PVC   UP   10    UP UBR 155000  UP
0/0/0     19   1   38  PVC   UP   14  DOWN UBR 155000  UP
0/0/0      6   1   40  PVC   UP    8  DOWN UBR 155000  UP
0/0/0      1   1   42  PVC   UP    6  DOWN UBR 155000  UP
0/0/0      5   1   44  PVC-L UP    7  DOWN UBR 155000  UP
0/0/0      8   2   36  PVC   UP    9  DOWN UBR 155000  UP
0/0/0     11   3   38  PVC-M UP   11  DOWN UBR 155000  UP
0/0/0     10   3   40  PVC-L UP   12  DOWN UBR 155000  UP
0/0/0.100 14 100  100  PVC-L UP   21  DOWN UBR 155000  UP
```

This command also causes each PVC in turn to be analyzed in detail and reported, as shown for this single PVC:

```
Open: IPCP
  Bound to ATM4/0 VCD: 2, VPI: 0, VCI: 34
  Cloned from virtual-template: 1
  Last input 01:04:26, output never, output hang never
  Last clearing of "show interface" counters 5d02h
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     782 packets input, 30414 bytes, 0 no buffer
     Received 3 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     395 packets output, 5540 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions


NRP-1-8#
```

In this output, the router first displays the status of all PVCs configured for PPP in the ATM interface. The name of the VC, the VC VPI/VCI, the VC state (up/down, and the Virtual Access (VA) number are displayed, along with VA interface-specific configuration and status. Then the second portion (which shows only one PVC to avoid repetition) provides quick insight into the status of a particular PVC.

Overall, the **show atm pvc ppp** command's details might vary with the Cisco IOS Software version, so you should consult the particular version's release notes for full details of the columns.

Along with configuring the PVC mapping and verifying PVCs as needed, you also need to configure options for IP address management (DHCP) and security (AAA, RADIUS), as shown in the next section.

Configuring DHCP, AAA, and RADIUS

How can you configure the IP address allocation scheme, as well as security, which are required for PPPoA and the other, sophisticated topologies on the 6400?

To enable DHCP on the NRP, you can use this command set:

```
router(config)#ip dhcp-server <server name/ip> !the DHCP server IP address
 is the variable at the end of this line!
router(config)#interface virtual template-template 2
router(config-if)#ip unnumbered ethernet 1/0 !this is the management Ethernet port
  to be used as the address for the virtual template; this is also called the
  Backplane Ethernet (BPE)!

router(config-if)#peer default ip address dhcp
router(config-if)#ppp authentication chap
```

In this example, a DHCP server location (by IP address) and authentication method (CHAP) are defined under a virtual template (virtual template 2). The virtual template is then associated with a VC during the VC configuration process. Notice that the virtual template takes the address of the management Ethernet port. This is usually the next-hop gateway router address assigned to the CPE router during authentication.

Large-scale deployment of PPP user services requires the use of a central database such as RADIUS to ease the configuration burden. RADIUS servers, providing AAA functionality, contain the per-user configuration database, including password authentication and authorization information. To enable the functionality of AAA and define the application of the RADIUS protocol on the NRP, you can use the global config command set here:

```
aaa new-model
aaa authentication login default radius
aaa authentication ppp default radius
aaa authorization network radius
radius-server host 192.168.1.1 auth-port 1645 acct-port 1646 !These are the
  standard authentication and accounting ports for the RADIUS daemon!
radius-server timeout 20 !measure in seconds!
radius-server key root !the word root is the shared key between the Network Access
  Server (NAS) and the RADIUS server!

aaa new-model
aaa authentication login default none
aaa authentication ppp default local group radius
aaa authorization network default local group radius none
aaa accounting network default wait-start group radius

!
username cisco password 0 cisco
!

interface ATM0/0/0.132 point-to-point

 pvc 1/32
  encapsulation aal5mux
!
interface Virtual-Template1
 ip unnumbered FastEthernet0/0/0
```

```
 peer default ip address pool dsl-pool
 ppp authentication chap
!
radius-server host 192.168.2.20 auth-port 1645 acct-port 1646
radius-server key cisco
```

Following are the explanations for the pertinent commands:

- **aaa new-model**—Enables the AAA access control model.
- **aaa authentication login default none**—Ensures that if the user cannot be authenticated with the various methods defined in the next command, there is no default login. (All users must be authenticated according to at least one defined method.)
- **aaa authentication ppp {default |** *list-name***}** *method1* **[***method2***...]**—Specifies one or more AAA authentication methods for use on interfaces running PPP. The *list-name* option refers to the name of this particular method list (or the default list, as in the previous example), and the *method* option is a list of methods to be tried in turn. You can use the command **aaa authentication ppp** with the *method* keyword **local** to specify that the Cisco router or access server should use the local username database for authentication. In this case, the local username database is used first, and then RADIUS.
- The **aaa authorization network radius** command sets RADIUS for network authorization, address assignment, and access lists. It follows the order of methods defined for individual PPP login.

  You can further control access and accounting by using the **wait-start** keyword, which ensures that the RADIUS security server acknowledges the start notice before granting the user's process request. In this case, it is applied only to network activities. To stop all accounting activities on this line or interface, use the **none** keyword.
- The RADIUS-specific commands in the second half of the configuration are as follows:
  - **radius-server host {***hostname* **|** *ip-address***} [***auth-port port-number***][***acct-port port-number***]**—Specifies a RADIUS server host.
  - **radius-server key cisco**—Sets the encryption key to match that used on the RADIUS server—in this case, the keyword **cisco**.

The AAA and RADIUS commands also apply to PPPoE, which is discussed in the next section.

PPPoE

As you learned in Chapter 3, PPPoE requires a PPP software client at the user location. The client (either the desktop PC or the Cisco 820 series router) initiates a PPP session by encapsulating PPP frames into a MAC frame and then bridging the frame (over ATM/DSL) to the gateway router. From this point, PPP sessions can be established, authenticated, addressed, and so on. VCs and associated PPP sessions can also be associated with VPDN groups, as shown here:

```
NRP-1(config)#username cisco password 0 cisco
NRP-1(config)#vpdn enable
```

```
NRP-1(config)#vpdn-group 1
NRP-1(config-vpdn)#accept dialin
NRP-1(config-vpdn)#protocol pppoe
NRP-1(config-vpdn)#virtual-template 1
NRP-1(config-vpdn)#pppoe limit per-vc 20

NRP-1(config)#int virtual-template 4
NRP-1(config-if)#ip unnumbered FastEthernet0/0/0
NRP-1(config-if)#ip mtu 1492

NRP-1(config-if)#peer default ip address pool pppoe-pool
NRP-1(config-if)#ppp authentication chap

NRP-1(config)#ip local pool pppoe-pool 192.168.5.100 192.168.5.150

NRP-1(config)#int atm 0/0/0.143 point-to-point
NRP-1(config-subif)#
NRP-1(config-subif)#pvc 3/143
NRP-1(config-if-atm-vc)#encapsulation aal5snap
NRP-1(config-if-atm-vc)#protocol pppoe



ip cef
!
```

In this example, the username and password (both **cisco**) are defined in the first line.

The following PPPoE termination command sets up VPDN:

```
vpdn enable
```

The following defines the VPDN group:

```
pppoe limit per-vc number
```

The command **vpdn-group 1** identifies the particular VPDN group whose characteristics follow. The command **accept dialin** configures the router to accept dial-in calls. The command **protocol pppoe** defines the VPDN's protocol as PPPoE. The command **virtual-template 1** associates this VPDN group with a particular virtual template, which is quite like the virtual template you created and referenced for PPPoA in the previous section. The optional command **pppoe limit per-vc 20** limits the number of PPPoE sessions that can be established on a virtual circuit. This limitation is a great help in managing the licensing of user accounts. The default is 100 sessions per VC.

Notice that the encapsulation type is aal5snap. This is the standard for both PPPoE and bridging.

The **interface virtual-template** *number* command selects the virtual-access interface to be configured.

The **ip unnumbered fastethernet 0/0/0** command enables IP on the interface without specifying a new address. In this case, the only address that might be associated with this virtual-template interface is the one already defined for the Fast Ethernet interface.

The **ip mtu 1492** command sets the IP MTU size to 1492, which is required for PPPoE.

The **peer default ip address pool pppoe-pool** command specifies that the dynamic IP addresses will come from the pool called pppoe-pool. Notice that the range of IP addresses in this pool, 51 in all, is defined in the following authentication definition line:

```
ppp authentication chap
```

This command sets the authentication protocol.

The **NRP-1(config)#interface ATM0/0/0.143 point-to-point** command defines a subinterface of the point-to-point type.

The **NRP-1(config-subif)#** subcommand disables the forwarding of directed broadcasts on the interface. The default is to forward directed broadcasts. A directed broadcast is a packet sent to a specific network or series of networks, whereas a flooded broadcast packet is sent to every network. A directed-broadcast address includes the network or subnet fields.

The **NRP-1(config-subif)#pvc 3/143** command defines a PVC on the subinterface.

The **NRP-1(config-if-atm-vc)#encapsulation aal5snap** command defines the type of encapsulation for PVC 3/143 as aal5snap, which is required for PPPoE (and RFC 2684 bridging as well, as you will see in the next section).

The command **NRP-1(config-if-atm-vc)#protocol pppoe** explicitly defines the protocol type as PPPoE, which avoids confusion with the much simpler bridging type of PVC, which is also based on RFC 2684.

The **ip cef** command enables Cisco Express Forwarding (CEF), because CEF is required for PPPoE.

This concludes the explanation of PPPoE on the Cisco 6400's node route processor.

RFC 2684 Bridging

This simplest type of architecture is frequently found in legacy DSL networks. It requires setting up a translation between the bridged network, the DSL users themselves, and the larger routed network, the Internet outside the local service area. This translation on the NRP is a bridge group virtual interface, meaning that you arbitrarily convert one of the Layer 3 interfaces to a Layer 2 bridging interface on the NRP. Here is an example of the configuration required:

```
router(config)#bridge irb
router(config)#bridge 1 route ip

router(config)#interface atm0/0/0.1 point-to-point
router(config-subif)#bridge-group 1
router(config-subif)#pvc 10/101
router(config-if-atm-vc)#encapsulation aal5snap
router(config-if-atm-vc)#ubr 384
router(config-if-atm-vc)#protocol bridge broadcast
router(config-if-atm-vc)#exit

router(config)#bridge 1 protocol ieee

router(config)#int bvi 1
router(config-if)#ip add 192.168.1.1 255.255.255.0
```

The following are explanations of the commands used in this example:

- **(config)#bridge irb**—Defines IRB. Usually for general Internet access, bridging may be combined with IRB to terminate the bridged traffic and route the traffic to an IP or IPX network. Issuing the **bridge irb** command enables the IRB feature in IOS, but you still need to specify the protocol to be used from the bridged domain to the routed domain. That is the result of the **bridge 1 route ip** command. When you configure **bridge 1 route ip**, a BVI is created. The BVI number corresponds to the bridge group. All other protocols are bridged. You can also route multiple protocols over a BVI.
- **(config)#interface atm0/0/0.1 point-to-point**—Defines a subinterface and specifies that it is a point-to-point type.
- **(config-if)#bridge-group 1**—Associates bridge group 1 with the subinterface.
- **(config-if)#pvc 10/101**—Creates a PVC on the subinterface.
- **router(config-if-atm-vc)#encapsulation aal5snap**—Defines the encapsulation type for this PVC as aal5snap, the same as for PPPoE.
- **router(config-if-atm-vc)#ubr 384**—(Optional) Sets the peak cell rate for this UBR connection at 384 Kbps or any other reasonable value.
- **router(config-if-atm-vc)#exit**—Exits PVC configuration mode.
- **Router(config)#bridge 1 protocol ieee**—Sets the Spanning Tree Protocol type as **ieee**, which is by far the most common type.
- **(config)#interface bvi 1**—Opens interface configuration mode, allowing you to define an IP address in the next line. When you intend to bridge and route a given protocol in the same bridge group, you must configure the network-layer attributes (Layer 3) of the protocol on the BVI. Do not configure protocol attributes on the BVIs. No bridging attributes can be configured on them.

  Although it is generally the case that all bridged segments belonging to a bridge group are represented as a single segment or network to the routing protocol, there are situations in which several individual networks coexist within the same bridged segment. To make it possible for the routed domain to learn about the other networks behind the BVI, configure a secondary address on the BVI to add the corresponding network to the routing process.

A more-scalable and less-insecure form of RFC 2684 bridging is available—RBE. Configuring the 6400's NRP for RBE is requires only the addition of the command **atm route-bridged** to the PVC's definition. As you can see in the following, you also might need to define host routes if you use unnumbered interfaces:

```
(config)#interface ATM0/0/0.133 point-to-point
  ip unnumbered Loopback0

     atm route-bridged ip
 pvc 1/33
     encapsulation aal5snap
```

```
! only need host routes when using unnumbered interfaces


ip route 172.168.1.2 255.255.255.255 ATM0/0/0.132
ip route 172.168.1.3 255.255.255.255 ATM0/0/0.133
```

This concludes the discussion of terminating PVCs with RFC 2684 bridging on the 6400's NRP.

VPN Configurations on the Cisco 6400

Configuring a virtual private network on the 6400's NRP requires more work outside the DSL network, and the 6400 itself, than any of the other architecture types. Although configuring the other end of the tunnel is far beyond the DSL network configuration, you have the opportunity now to learn these important procedures.

First, you should review these terms related to VPN:

- **Tunnel**—A virtual pipe between the LAC and the LNS that carries multiple PPP sessions. It consists of user traffic and header information necessary to support the tunnel. The tunnel profile can be in the local router configuration or on a remote RADIUS server.
- **L2TP access concentrator (LAC)**—The client directly connects to the LAC, which resides between the home network (Cisco in this example) and the remote user. Its job is to tunnel PPP frames through the Internet to the local L2TP network server (LNS). It may tunnel any protocol carried within PPP. The LAC initiates incoming calls and receives outgoing calls. For our examples, the LAC is typically the Cisco 6400's NRP.
- **L2TP network server (LNS)**—The termination point for the L2TP tunnel where the home LAN is located. It is the home LAN's access point where PPP frames are processed and passed to higher-layer protocols. An LNS can operate on any platform capable of PPP termination. The LNS handles the server side of the L2TP protocol, although it can initiate the outgoing call to create a tunnel.
- **Session**—A single tunneled PPP session. Also referred to as a call.
- **AAA**—The authentication, authorization, and accounting server, used to store domain and user information. At the LAC, the AAA server stores domain information necessary to identify and establish the tunnel to the remote LNS. At the LNS, the AAA server stores user information needed to authenticate the tunnel user.

L2TP can support either PPPoA or PPPoE encapsulation on the PVC coming from the CPE. The LAC accepts this PPP session and establishes the L2TP tunnel to the LNS. After Link Control Protocol (LCP) has been negotiated, the LAC partially authenticates the end user with CHAP or PAP but does not process PPP packets. The user is authenticated on the LNS where the call terminates. Information necessary to identify the remote LNS can be stored in the AAA server or can be entered directly into the LAC's configuration.

The username@domain name is used to verify that the user is a VPDN client and to provide a mapping to a specific endpoint LNS. The tunnel endpoints (LAC and LNS) authenticate each other, and the tunnel opens. As soon as the tunnel exists, an L2TP session is created for the end

user. The LAC propagates the LCP negotiated options and the partially authenticated CHAP/PAP information to the LNS.

L2TP utilizes two types of messages—control messages and data messages. Control messages are used to establish, maintain, and clear a tunnel and to set up and clear sessions. Data messages are used to encapsulate PPP frames being carried over the tunnel.

L2TP guarantees the delivery of control messages through a control channel. Messages in the control channel have sequence numbers used to detect loss or out-of-order delivery. Lost control messages are retransmitted. Data messages may also use sequence numbers to reorder packets and detect lost packets.

To begin configuring VPDN, follow these steps:

Step 1 Start with configuring authentication for the L2TP tunnel. This causes the LAC to check for tunnel authentication using either a RADIUS server or a local database. To use local authorization, a local database of usernames and passwords can be defined on the LAC.

Step 2 Enable VPDN with the command **vpdn enable**. This extends remote access to a private network across a shared infrastructure, such as the Internet.

Step 3 Define a VPDN group, as you did earlier with PPPoE. You apply all VPDN attributes for the LAC through this group. You can use the command **vpdn-group** *number*. This VPDN group contains attributes for initiating the L2TP tunnel on the LAC. Typically, you need one VPDN group for each LAC. For an LNS that services many LACs, the configuration can become cumbersome. However, you can use the default VPDN group configuration if all the LACs will share the same tunnel attributes. An example of this scenario is an LNS that services a large department with many Windows NT L2TP clients that are colocated with the LAC. Each of the Windows NT devices is an L2TP client as well as a LAC. Each of these devices demands a tunnel to the LNS. If all the tunnels will share the same tunnel attributes, you can use a default VPDN group configuration, which simplifies the configuration process.

Step 4 Enable the LAC to initiate the L2TP tunnels using the **initiate-to** command. The configuration looks like this:

```
(config)#vpdn-group 2
 request-dialin
protocol l2tp
       domain Cisco.com
 initiate-to ip 192.168.2.2
 local name NRP-2
 l2tp tunnel password 7 060506324F41
```

In this example, the LAC is configured to initiate an L2TP tunnel to the LNS (whose IP address is 192.168.2.2) if the login contains the Cisco.com domain name. The LAC local host name and the shared secret password used for tunnel authentication between the LAC and the LNS can also be configured under the VPDN group.

Step 5 Move to the LNS and duplicate the security definitions, username, and password, as you did in Step 1 for the LAC.

Step 6 The next step on the LNS is to enable VPDN, again duplicating what you did on the LAC. You should also define a VPDN group to which you will apply all VPDN attributes for the LNS. This VPDN group contains attributes for accepting the L2TP tunnel on the LNS.

Step 7 Enable the LNS to accept L2TP tunnels, using the **terminate-from** command, as shown in the following, along with the other commands:

```
(config)#vpdn-group team1
 accept-dialin
 protocol l2tp
 virtual-template 1
 terminate-from hostname NRP-2
 l2tp tunnel password 7 0822455D0
```

The command **accept-dialin** specifies the local name to use for authenticating and the virtual template to use for cloning new virtual-access interfaces when an incoming L2TP tunnel connection is requested from a specific peer.

The command **terminate-from hostname NRP-2** defines the attributes needed to find the LNS for the given domain name. You can enter the LNS's IP address, or in this case, a predefined host name (NRP-2) instead of the IP address.

In the example, the LNS is configured to accept an L2TP tunnel from the LAC with a host name of NRP-2. The shared secret password used for tunnel authentication between the LAC and the LNS can also be configured under the vpdn-group. The virtual-template 1 interface is used for creating the virtual-access interface.

Step 8 A final task on the LNS, although not necessarily performed in strict order, is to define the virtual-template interface and an IP address pool for dynamic IP address assignment over IPCP, as shown in the following:

```
interface Virtual-Template1
     ip address 192.168.7.1 255.255.255.0
     peer default ip address pool L2TP_pool
     ppp authentication chap
ip local pool L2TP_pool 192.168.7.100 192.168.7.120
```

You may continue to configure the virtual template interface with configuration parameters you want applied to the virtual-access interfaces.

Now that you have configured the LAC and LNS, you can use the **show vpdn** command to display information about active tunnels and message identifiers in a VPDN. Here is an example of the output:

```
lac1#show vpdn
L2TP Tunnel and Session Information
Total tunnels 1 sessions 1
```

```
LocID RemID Remote Name  State Remote Address Port Sessions
11984 36217 ciscoemp     est   172.30.248.10  1701 1

LocID RemID TunID Intf Username   State LastChg Fastswitch
9     5     11984 Vi6 joe@cisco.com  est  00:31:39  enabled

%No active L2F tunnels

%No active PPPoE tunnels
```

This concludes the explanation of configuring L2TP on the Cisco 6400 and other devices in the larger network.

Summary

In this vital chapter, you configured all three components of the DSL network with the CLI Cisco IOS Software. Moving from the Cisco 827 and 827-4V, with their basic data and added voice configurations, you then followed the DSL traffic upstream to the Cisco 6000 series of IP DSL Switches. There you learned not only how to configure the IP DSL Switch for redundancy, but also how to map the incoming DSL traffic to outbound ATM traffic and to adjust configurations for particular models of IP DSL Switches and line cards. Finally, you learned how APS and other redundancy options work on the Cisco 6400 UAC, followed by configuration of the basic architectures, such as PPPoA on the NRP.

Review Questions

1 What does the command **ip unnumbered gigabit ethernet 0/0/0** accomplish?

   A It enables IP on the interface without assigning a specific IP address.

   B It selects the authentication protocol and an optional secondary protocol.

   C It associates a virtual template with a virtual template interface.

   D It specifies a dynamic IP address assignment method, from either an IP address pool or a DHCP server.

2 Which of the following apply to the NRP?

   A Layer 3 services

   B High-speed ATM switch

   C Central control for the 6400

   D Traffic shaping

3 Which of the following apply to the NSP?

- **A** Central control of the 6400
- **B** Layer 3 packet services
- **C** Terminates PPP
- **D** End-to-end transport authentication

4 Which of the following apply to the NLC?

- **A** Optical interface
- **B** Automatic protection switching
- **C** Layer 3 services
- **D** Terminates PPP

5 True or false: The 6400 must be configured with redundant NSP cards.

6 The IP address for the subscriber's PC can be provided by what? (Select all that apply.)

- **A** Static configuration
- **B** DHCP server on the ATU-R
- **C** DHCP server on the aggregation router
- **D** DHCP server on the RADIUS server

7 The interface designation **int atm 0/0/0** on the 6400 designates what?

- **A** ATM interface slot 0, subslot 0, card 0
- **B** Backplane connection, subslot 0, port 0
- **C** ATM interface port 0, subslot 0, slot 0
- **D** ATM interface slot 0, port 0, subslot 0

8 From the NSP, interface designation **atm 1/0/0** indicates what?

- **A** Interface to the card in port 1, subslot 0, slot 0
- **B** Interface to the card in slot 1, subslot 0, card 0
- **C** Interface to the card in slot 1, subslot 0, port 0
- **D** Interface to the card in slot 1, card 0, port 0

9 What is the NSP switch default ATM protocol?

- **A** LAPD
- **B** X.25
- **C** UNI 4.0
- **D** ISDN

10 Which of the following best describes a BVI?

- **A** It provides the interface between a bridge group and the routed network.
- **B** It assembles packets coming from the bridge group into FR frames.
- **C** It restricts broadcast messages within a bridge group.
- **D** None of the above

11 Which of the following best describes SONET APS on the 6400 system?

- **A** Redundant NLC cards supplying OC-3 traffic to the NSP.
- **B** Redundant NLC ports supplying OC-3 traffic to the NRP.
- **C** The 6400 system does not support SONET APS.
- **D** Redundant NSP configured for failover.

12 **ilmi-pvc-discovery** refers to the process of what?

- **A** The automatic discovery of configured PVCs on the NRP card
- **B** The automatic discovery of configured PVCs on the NLC card by the NRP card
- **C** The automatic discovery of configured PVCs on the NSP card by the NRP card
- **D** None of the above

13 When two NRP-1s are configured in a redundant configuration, which of the following occurs?

- **A** Both NRP-1s provide redundant traffic to the NSP.
- **B** Both NRP-1s provide redundant traffic to the NLC.
- **C** Both NRP-1s provide redundant traffic to the NPP.
- **D** None of the above

14 True or false: The function of the **encapsulation aal5mux ppp dialer** command is to specify the encapsulation type for the PPPoA PVC on the 827.

15 Which protocol contains information about higher-layer protocols, including IP and IPX, and their control protocols (IPCP for IP)?

- **A** NCP
- **B** LLC
- **C** PCP
- **D** NLP