About This Manual

Document Objectives

This publication provides internetworking design and implementation case studies and examples, with the intent to help you identify and implement practical internetworking strategies that are both flexible and scalable.

This publication was developed to assist professionals preparing for Cisco Certified Internetwork Expert (CCIE) candidacy, though it is a valuable resource for all internetworking professionals. It is designed for use in conjunction with other Cisco manuals or as a standalone reference. You may find it helpful to refer to the *Cisco CCIE Fundamentals: Network Design*, which provides detailed descriptions of the internetworking strategies and technologies used in this publication.

Audience

This publication is intended to support the network administrator who designs and implements router- or switched-based internetworks, and describes practical examples of how to apply Cisco features to meet internetworking needs. Readers should know how to configure a Cisco router and should be familiar with the protocols and media that their routers have been configured to support.

Readers will better understand the material in this publication if they are familiar with networking terminology. The Cisco *Internetworking Terms and Acronyms* publication is a useful reference for those with minimal knowledge of networking terms.

Document Organization

This manual contains twelve chapters, which are described below:

Chapter 1 “RIP and OSPF Redistribution,” which addresses the issue of integrating Routing Information Protocol (RIP) networks with Open Shortest Path First (OSPF) networks

Chapter 2, “Dial-on-Demand Routing,” which addresses the dial-on-demand routing (DDR) feature that allows you to use existing telephone lines to form a wide-area network (WAN).

Chapter 3, “Increasing Security on IP Networks,” which addresses the broad topic of network security.

Chapter 4, “Integrating Enhanced IGRP into Existing Networks,” which addresses the Enhanced Interior Gateway Routing Protocol (IGRP).

Chapter 5, “Reducing SAP Traffic in Novell IPX Networks,” which addresses how to deal with the nuances of Novel IPX networks.

Chapter 6, “UDP Broadcast Flooding,” which addresses he interworkings of broadcast data packets.

Chapter 7, “STUN for Front-End Processors,” which addresses serial tunneling (STUN) and the integration of traditional *systems network architecture* (SNA) networks with multiprotocol networks.

Chapter 8, “Using ISDN Effectively in Multiprotocol Networks,” which addresses how, as telephone companies make Integrated Services Digital Network (ISDN) services available, ISDN is becoming an increasingly popular way of connecting remote sites.

Chapter 9, “Using HSRP for Fault-Tolerant IP Routing,”

which addresses Cisco’s Hot Standby Routing Protocol (HSRP), which provides automatic router backup when you configure it on Cisco routers that run the Internet Protocol (IP) over Ethernet, Fiber Distributed Date Interface (FDDI), and Token Ring local-area networks (LANs).

Chapter 10, “LAN Switching,” which addresses how to deal with the fact that today’s local-area networks LANs) are becoming increasingly congested and overburdened.

Chapter 11, “Multicasting in IP and AppleTalk Networks,” which addresses the concept of end-users being able to send and receive audio and video (known collectively as *multimedia*) at the desktop has gained considerable attention and acceptance that has become increasingly common in the past few years.

Chapter 12, “Scaling Dial-on-Demand Routing,” which addresses the design of an access network that allows a large number of remote sites to communicate with an existing central-site network.

Document Conventions

In this publication, the following conventions are used:

- Commands and keywords are in **boldface**.
- New, important terms are *italicized* when accompanied by a definition or discussion of the term.
- Protocol names are *italicized* at their first use in each chapter.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

RIP and OSPF Redistribution

This case study addresses the issue of integrating Routing Information Protocol (RIP) networks with Open Shortest Path First (OSPF) networks. Most OSPF networks also use RIP to communicate with hosts or to communicate with portions of the internetwork that do not use OSPF. Cisco supports both the RIP and OSPF protocols and provides a way to exchange routing information between RIP and OSPF networks. This case study provides examples of how to complete the following phases in redistributing information between RIP and OSPF networks, including the following topics:

- Configuring a RIP Network
- Adding OSPF to the Center of a RIP Network
- Adding OSPF Areas
- Setting Up Mutual Redistribution

Configuring a RIP Network

Figure 1-1 illustrates a RIP network. Three sites are connected with serial lines. The RIP network uses a Class B address and an 8-bit subnet mask. Each site has a contiguous set of network numbers.

Figure 1-1 A RIP network.

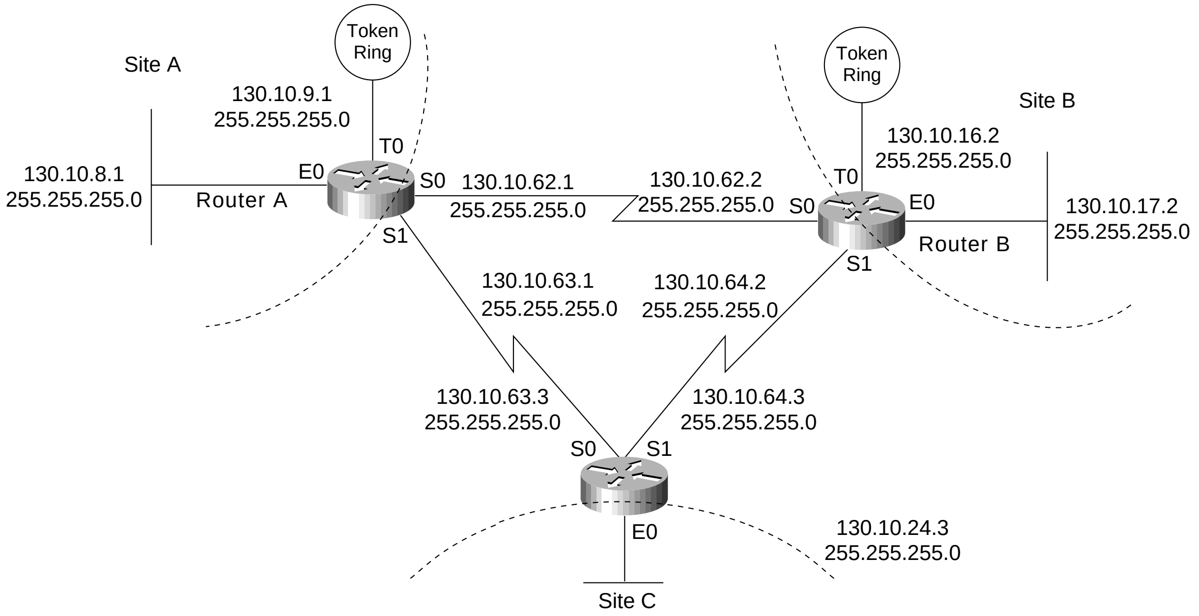


Table 1-1 lists the network address assignments for the RIP network, including the network number, subnet range, and subnet masks. All interfaces indicate network 130.10.0.0; however, the specific address includes the subnet and subnet mask. For example, serial interface 0 on Router C has an IP address of 130.10.63.3 with a subnet mask of 255.255.255.0.

Table 1-1 RIP Network Address Assignments

Network Number	Subnets	Subnet Masks
130.10.0.0	**Site A:** 8 through 15	255.255.255.0
130.10.0.0	**Site B:** 16 through 23	255.255.255.0
130.10.0.0	**Site C:** 24 through 31	255.255.255.0
130.10.0.0	**Serial Backbone:** 62 through 64	255.255.255.0

Configuration File Examples

The following commands in the configuration file for Router A determine the IP address for each interface and enable RIP on those interfaces:

```
interface serial 0
ip address 130.10.62.1 255.255.255.0
interface serial 1
ip address 130.10.63.1 255.255.255.0
interface ethernet 0
ip address 130.10.8.1 255.255.255.0
interface tokenring 0
ip address 130.10.9.1 255.255.255.0
router rip
network 130.10.0.0
```

The following commands in the configuration file for Router B determine the IP address for each interface and enable RIP on those interfaces:

```
interface serial 0
ip address 130.10.62.2 255.255.255.0
interface serial 1
ip address 130.10.64.2 255.255.255.0
interface ethernet 0
ip address 130.10.17.2 255.255.255.0
interface tokenring 0
ip address 130.10.16.2 255.255.255.0
router rip
network 130.10.0.0
```

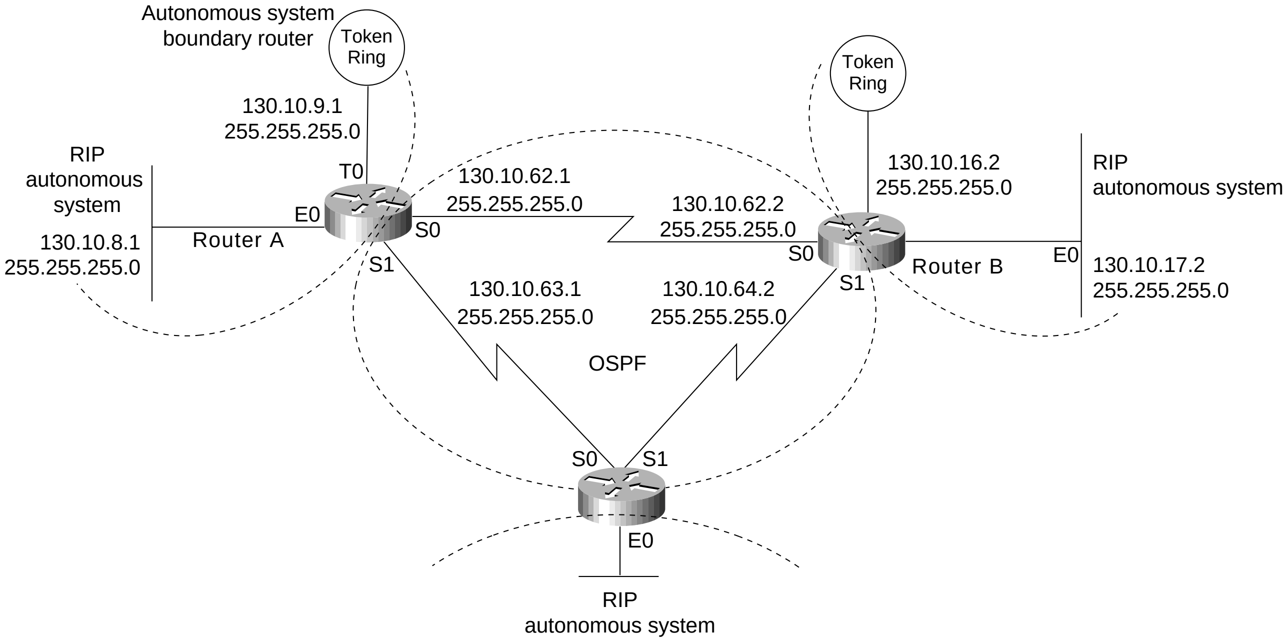
The following commands in the configuration file for Router C determine the IP address for each interface and enable RIP on those interfaces:

```
interface serial 0
ip address 130.10.63.3 255.255.255.0
interface serial 1
ip address 130.10.64.3 255.255.255.0
interface ethernet 0
ip address 130.10.24.3 255.255.255.0
router rip
network 130.10.0.0
```

Adding OSPF to the Center of a RIP Network

A common first step in converting a RIP network to OSPF is to add backbone routers that run both RIP and OSPF, while the remaining network devices run RIP. These backbone routers are OSPF autonomous system boundary routers. Each autonomous system boundary router controls the flow of routing information between OSPF and RIP. In Figure 1-2, Router A is configured as the autonomous system boundary router.

Figure 1-2 RIP network with OSPF at the center.



RIP does not need to run between the backbone routers; therefore, RIP is suppressed on Router A with the following commands:

```
router rip
passive-interface serial 0
passive-interface serial 1
```

The RIP routes are redistributed into OSPF by all three routers with the following commands:

```
router ospf 109
redistribute rip subnets
```

The **subnets** keyword tells OSPF to redistribute all subnet routes. Without the **subnets** keyword, only networks that are not subnetted will be redistributed by OSPF. Redistributed routes appear as external type 2 routes in OSPF. Each RIP domain receives information about networks in other RIP domains and in the OSPF backbone area from the following commands that redistribute OSPF routes into RIP:

```
router rip
redistribute ospf 109 match internal external 1 external 2
default-metric 10
```

The **redistribute** command uses the **ospf** keyword to specify that OSPF routes are to be redistributed into RIP. The keyword **internal** indicates the OSPF intra-area and interarea routes: External 1 is the external route type 1, and external 2 is the external route type 2. Because the command in the example uses the default behavior, these keywords may not appear when you use the **write terminal** or **show configuration** commands.

Because metrics for different protocols cannot be directly compared, you must specify the default metric in order to designate the cost of the redistributed route used in RIP updates. All routes that are redistributed will use the default metric.

In Figure 1-2, there are no paths directly connecting the RIP clouds. However, in typical networks, these paths, or “back doors,” frequently exist, allowing the potential for feedback loops. You can use access lists to determine the routes that are advertised and accepted by each router. For example, access list 11 in the configuration file for Router A allows OSPF to redistribute information learned from RIP only for networks 130.10.8.0 through 130.10.15.0:

```
router ospf 109
redistribute rip subnet
distribute-list 11 out rip
access-list 11 permit 130.10.8.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.255
```

These commands prevent Router A from advertising networks in other RIP domains onto the OSPF backbone, thereby preventing other boundary routers from using false information and forming a loop.

Configuration File Examples

The full configuration for Router A follows:

```
interface serial 0
ip address 130.10.62.1 255.255.255.0
interface serial 1
ip address 130.10.63.1 255.255.255.0
interface ethernet 0
ip address 130.10.8.1 255.255.255.0
interface tokenring 0
ip address 130.10.9.1 255.255.255.0
!
router rip
default-metric 10
network 130.10.0.0
passive-interface serial 0
passive-interface serial 1
redistribute ospf 109 match internal external 1 external 2
!
router ospf 109
network 130.10.62.0 0.0.0.255 area 0
network 130.10.63.0 0.0.0.255 area 0
redistribute rip subnets
distribute-list 11 out rip
!
access-list 11 permit 130.10.8.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.255
```

The full configuration for Router B follows:

```
interface serial 0
ip address 130.10.62.2 255.255.255.0
interface serial 1
ip address 130.10.64.2 255.255.255.0
interface ethernet 0
ip address 130.10.17.2 255.255.255.0
interface tokenring 0
ip address 130.10.16.2 255.255.255.0
!
router rip
default-metric 10
network 130.10.0.0
passive-interface serial 0
passive-interface serial 1
redistribute ospf 109 match internal external 1 external 2
!
router ospf 109
network 130.10.62.0 0.0.0.255 area 0
network 130.10.64.0 0.0.0.255 area 0
redistribute rip subnets
distribute-list 11 out rip
access-list 11 permit 130.10.16.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.255
```

The full configuration for Router C follows:

```
interface serial 0
ip address 130.10.63.3 255.255.255.0
interface serial 1
ip address 130.10.64.3 255.255.255.0
interface ethernet 0
ip address 130.10.24.3 255.255.255.0
!
router rip
default-metric 10
!
network 130.10.0.0
passive-interface serial 0
passive-interface serial 1
redistribute ospf 109 match internal external 1 external 2
!
router ospf 109
network 130.10.63.0 0.0.0.255 area 0
network 130.10.64.0 0.0.0.255 area 0
redistribute rip subnets
distribute-list 11 out rip
access-list 11 permit 130.10.24.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.255
```

Adding OSPF Areas

Figure 1-3 illustrates how each of the RIP clouds can be converted into an OSPF area. All three routers are area border routers. Area border routers control network information distribution between OSPF areas and the OSPF backbone. Each router keeps a detailed record of the topology of its area and receives summarized information from the other area border routers on their respective areas.

Figure 1-3 Configuring route summarization between OSPF areas.

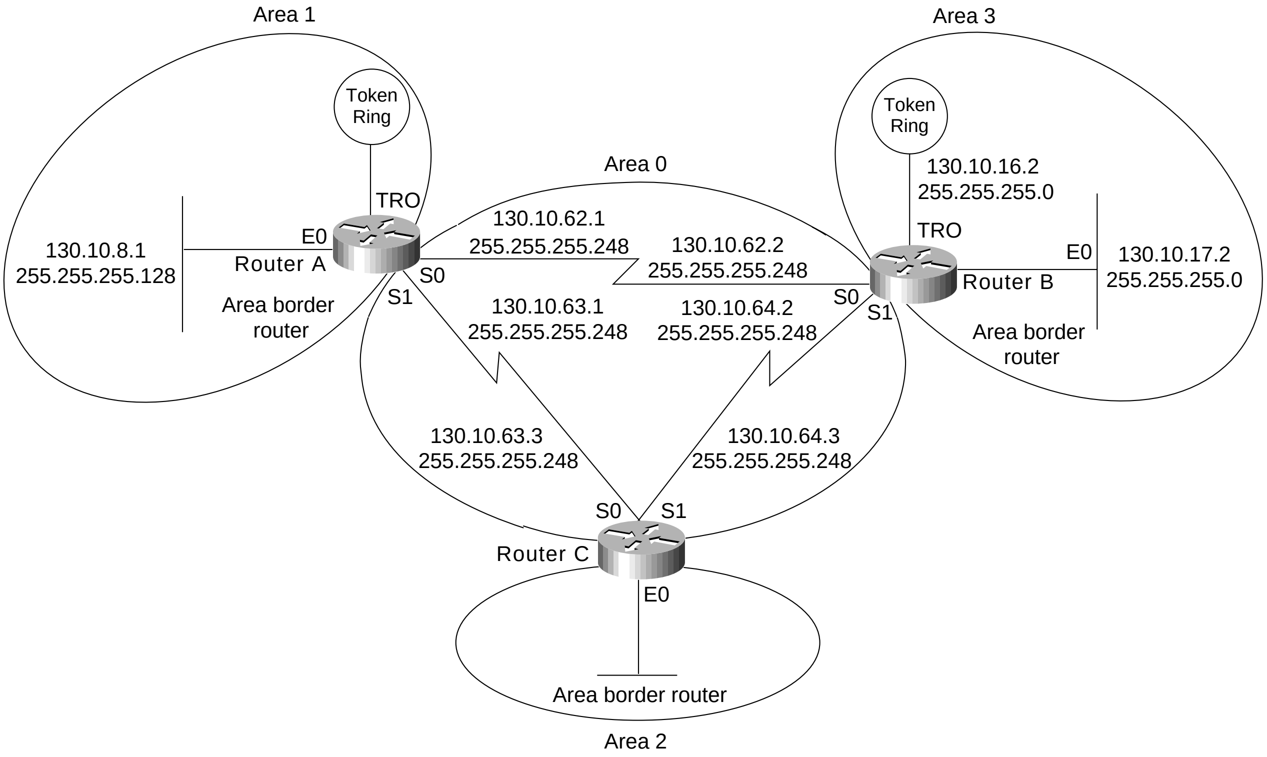


Figure 1-3 also illustrates *variable-length subnet masks* (VLSMs). VLSMs use different size network masks in different parts of the network for the same network number. VLSM conserves address space by using a longer mask in portions of the network that have fewer hosts. Table 1-2 lists the network address assignments for the network, including the network number, subnet range, and subnet masks. All interfaces indicate network 130.10.0.0.

Table 1-2 OSPF Address Assignments

Network Number	Subnets	Subnet Masks
130.10.0.0	**Area 0:** 62 through 64	255.255.255.248
130.10.0.0	**Area 1:** 8 through 15	255.255.255.0
130.10.0.0	**Area 2:** 16 through 23	255.255.255.0
130.10.0.0	**Area 3:** 24 through 31	255.255.255.0

To conserve address space, a mask of 255.255.255.248 is used for all the serial lines in area 0. If an area contains a contiguous range of network numbers, an area border router uses the **range** keyword with the **area** command to summarize the routes that are injected into the backbone:

```
router ospf 109
network 130.10.8.0 0.0.7.255 area 1
area 1 range 130.10.8.0 255.255.248.0
```

These commands allow Router A to advertise one route, 130.10.8.0 255.255.248.0, which covers all subnets in Area 1 into Area 0. Without the **range** keyword in the **area** command, Router A would advertise each subnet individually; for example, one route for 130.10.8.0 255.255.255.0, one route for 130.10.9.0 255.255.255.0, and so forth.

Because Router A no longer needs to redistribute RIP routes, the **router rip** command can now be removed from the configuration file; however, it is common in some environments for hosts to use RIP to discover routers. When RIP is removed from the routers, the hosts must use an alternative technique to find the routers. Cisco routers support the following alternatives to RIP:

- *ICMP Router Discovery Protocol (IRDP)*—This technique is illustrated in the example at the end of this section. IRDP is the recommended method for discovering routers. The **ip irdp** command enables IRDP on the router. Hosts must also run IRDP.
- *Proxy Address Resolution Protocol (ARP)*—If the router receives an ARP request for a host that is not on the same network as the ARP request sender, and if the router has the best route to that host, the router sends an ARP reply packet giving the router's own local data link address. The host that sent the ARP request then sends its packets to the router, which forwards them to the intended host. Proxy ARP is enabled on routers by default. Proxy ARP is transparent to hosts.

Configuration File Examples

The full configuration for Router A follows:

```
interface serial 0
ip address 130.10.62.1 255.255.255.248
interface serial 1
ip address 130.10.63.1 255.255.255.248
interface ethernet 0
ip address 130.10.8.1 255.255.255.0
ip irdp
interface tokenring 0
ip address 130.10.9.1 255.255.255.0
ip irdp
router ospf 109
network 130.10.62.0 0.0.0.255 area 0
network 130.10.63.0 0.0.0.255 area 0
network 130.10.8.0 0.0.7.255 area 1
area 1 range 130.10.8.0 255.255.248.0
```

The full configuration for Router B follows:

```
interface serial 0
ip address 130.10.62.2 255.255.255.248
interface serial 1
ip address 130.10.64.2 255.255.255.248
interface ethernet 0
ip address 130.10.17.2 255.255.255.0
ip irdp
interface tokenring 0
ip address 130.10.16.2 255.255.255.0
ip irdp
router ospf 109
network 130.10.62.0 0.0.0.255 area 0
network 130.10.64.0 0.0.0.255 area 0
network 130.10.16.0 0.0.7.255 area 2
area 2 range 130.10.16.0 255.255.248.0
```

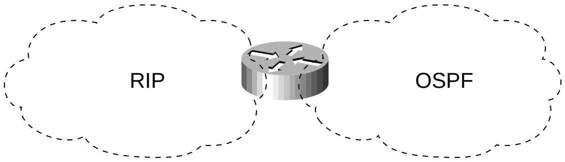
The full configuration for Router C follows:

```
interface serial 0
ip address 130.10.63.2 255.255.255.248
interface serial 1
ip address 130.10.64.2 255.255.255.248
interface ethernet 0
ip address 130.10.24.3 255.255.255.0
ip irdp
router ospf 109
network 130.10.63.0 0.0.0.255 area 0
network 130.10.64.0 0.0.0.255 area 0
network 130.10.24.0 0.0.0.255 area 3
area 3 range 130.10.24.0 255.255.248.0
```

Setting Up Mutual Redistribution

It is sometimes necessary to accommodate more complex network topologies such as independent RIP and OSPF clouds that must perform mutual redistribution. In this scenario, it is critically important to prevent potential routing loops by filtering routes. The router in Figure 1-4 is running both OSPF and RIP.

Figure 1-4 Mutual redistribution between RIP and OSPF networks.



With the following commands, OSPF routes will be redistributed into RIP. You must specify the default metric to designate the cost of the redistributed route in RIP updates. All routes redistributed into RIP will have this default metric.

```
! passive interface subcommand from previous example is left out for clarity!
router rip
default-metric 10
network 130.10.0.0
redistribute ospf 109
```

It is a good practice to strictly control which routes are advertised when redistribution is configured. In the following example, a **distribute-list out** command causes RIP to ignore routes coming from the OSPF that originated from the RIP domain.

```
router rip
distribute-list 10 out ospf 109
!
access-list 10 deny 130.10.8.0 0.0.7.255
access-list 10 permit 0.0.0.0 255.255.255.255
```

Router A

The full configuration for the router follows:

```
interface serial 0
ip add 130.10.62.1 255.255.255.0
!
interface serial 1
ip add 130.10.63.1 255.255.255.0
!
interface ethernet 0
ip add 130.10.8.1 255.255.255.0
!
interface tokenring 0
ip add 130.10.9.1 255.255.255.0
!
router rip
default-metric 10
network 130.10.0.0
passive-interface serial 0
passive-interface serial 1
redistribute ospf 109
distribute-list 10 out ospf 109
!
router ospf 109
network 130.10.62.0 0.0.0.255 area 0
network 130.10.63.0 0.0.0.255 area 0
redistribute rip subnets
distribute-list 11 out rip
!
access-list 10 deny 130.10.8.0 0.0.7.255
access-list 10 permit 0.0.0.0 255.255.255.255
access-list 11 permit 130.10.8.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.255
```

Summary

Because it is common for OSPF and RIP to be used together, it is important to use the practices described here in order to provide functionality for both protocols on an internetwork. You can configure autonomous system boundary routers that run both RIP and OSPF and redistribute RIP routes into the OSPF and vice versa. You can also create OSPF areas using area border routers that provide route summarizations. Use VLSM to conserve address space.

Dial-on-Demand Routing

Cisco's dial-on-demand routing (DDR) feature allows you to use existing telephone lines to form a wide-area network (WAN). While using existing telephone lines, you can analyze traffic patterns to determine whether the installation of leased lines is appropriate. DDR provides significant cost savings over leased lines for links that are utilized for only a few hours each day or that experience low traffic flow.

DDR over serial lines requires the use of dialing devices that support V.25bis. V.25bis is an International Telecommunication Union Telecommunication (ITU-T) Standardization Sector standard for in-band signaling to bit synchronous data communications equipment (DCE) devices. A variety of devices support V.25bis, including analog V.32 modems, ISDN terminal adapters, and inverse multiplexers. Cisco's implementation of V.25bis supports devices that use the 1984 version of V.25bis (which requires the use of odd parity), as well as devices that use the 1988 version of V.25bis (which does not use parity).

Note The ITU-T carries out the functions of the former Consultative Committee for International Telegraph and Telephone (CCITT).

This case study describes the use of DDR to connect a worldwide network that consists of a central site located in San Francisco and remote sites located in Tokyo, Singapore, and Hong Kong. The following scenarios and configuration file examples are described:

- Having the Central Site Dial Out

  Describes the central and remote site configurations for three setups: a central site with one interface per remote site, a single interface for multiple remote sites, and multiple interfaces for multiple remote sites. Includes examples of the usage of rotary groups and access lists.

- Having the Central and Remote Sites Dial In and Dial Out

  Describes the central and remote site configurations for three setups: central site with one interface per remote site, a single interface for multiple remote sites, and multiple interfaces for multiple remote sites. Also describes the usage of Point-to-Point Protocol (PPP) encapsulation and the Challenge Handshake Authentication Protocol (CHAP).

- Having Remote Sites Dial Out

  A common configuration is one in which the remote sites place calls to the central site but the central site does not dial out. In a "star" topology, it is possible for all of the remote routers to have their serial interfaces on the same subnet as the central site serial interface.

- Using DDR as a Backup to Leased Lines

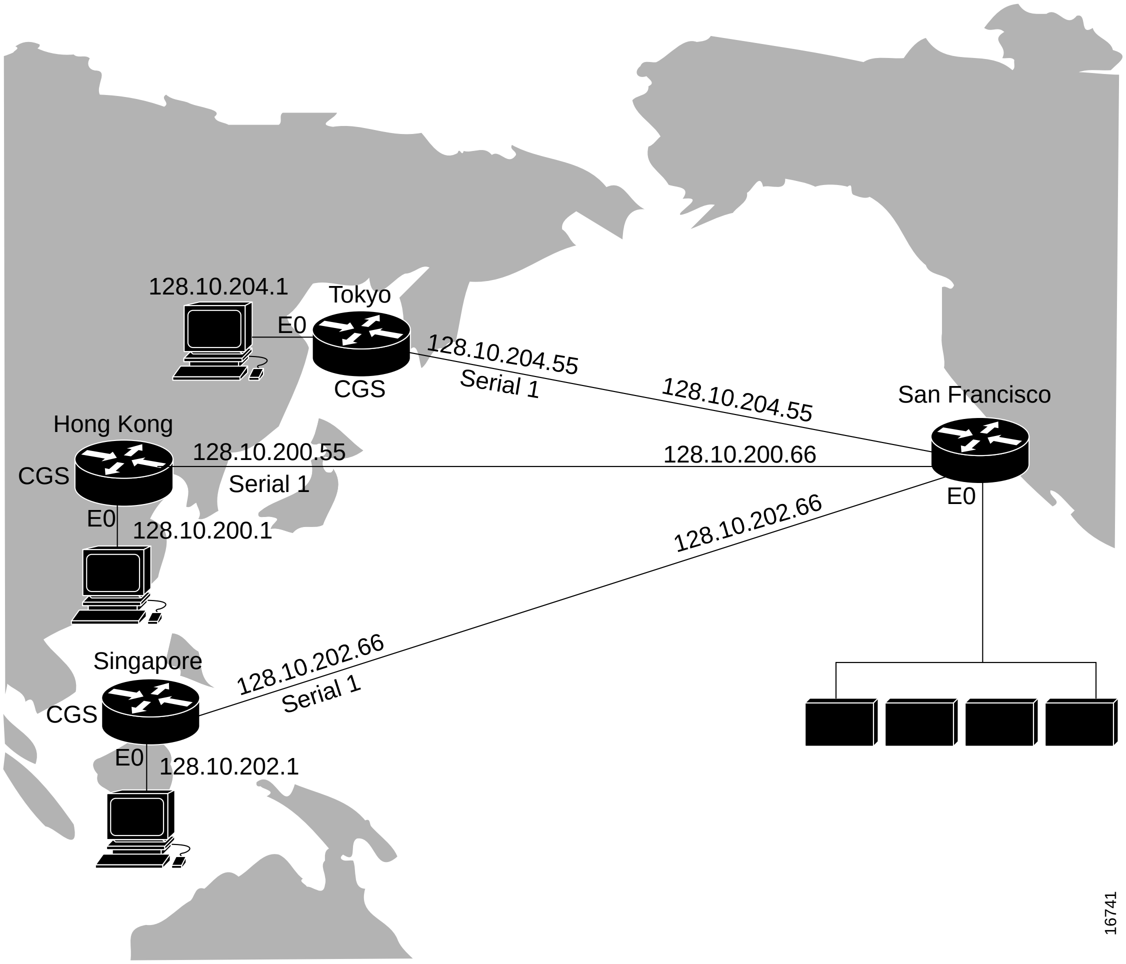
Describes the use of DDR as a backup method to leased lines and provides examples of how to use floating static routes on single and shared interfaces.

- Using Leased Lines and Dial Backup

  Describes the use of Data Terminal Ready (DTR) dialing and V.25bis dialing with leased lines.

Figure 2-1 shows the topology of the DDR network that is the subject of this case study.

Figure 2-1 DDR internetwork topology.



Note All examples and descriptions in this case study refer to features available in Software Release 9.1(9) or later. Some features are available in earlier releases. Features that are available only in Software Release 9.21 are indicated as such.

Having the Central Site Dial Out

In this example, the central site calls the remote sites. The cost of initiating a call from the United States to international sites is often lower than if the remote sites initiate the call, and it is expected that remote offices need to connect to the central site network only periodically. This section provides the following configuration examples in which the central site is configured to dial out:

- Configuring One Interface Per Remote Site
- Configuring a Single Interface for Multiple Remote Sites
- Configuring Multiple Interfaces for Multiple Remote Sites

Configuring One Interface Per Remote Site

For the initial configuration, the San Francisco central site is configured to have one interface per remote site.

Central Site: Dial Out Only

In the following configuration, the central site places the calls with a separate interface configured for each remote site. There is no support for answering calls in this configuration.

```
interface serial 5
description DDR connection to Hong Kong
ip address 128.10.200.66 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 0118527351625
pulse-time 1
dialer-group 1
!
interface serial 6
description DDR connection to Singapore
ip address 128.10.202.66 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 011653367085
pulse-time 1
dialer-group 1
!
interface serial 7
description DDR connection to Tokyo
ip address 128.10.204.66 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 0118127351625
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
redistribute static
! route to Hong Kong
ip route 128.10.200.0 255.255.255.192 128.10.200.65
! route to Singapore
ip route 128.10.202.0 255.255.255.192 128.10.202.65
! route to Tokyo
ip route 128.10.204.0 255.255.255.192 128.10.204.65
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
```

Interface Configuration

The configuration of the individual interfaces and Internet Protocol (IP) addresses is straightforward. The IP address for each interface is provided. The example uses a 6-bit host portion in IP addresses. The **dialer in-band** command enables DDR and V.25bis dialing on the interface. V.25bis is a ITU-T standard for in-band signaling to bit synchronous DCE devices. A variety of devices support V.25bis, ranging from analog V.32 modems to ISDN terminal adapters to inverse multiplexers.

The **dialer wait-for-carrier-time** command is set to 60 seconds. When using V.25bis, the router does not parse any responses it receives from the DCE. Instead, the router depends on the modem's Carrier Detect (CD) signal to indicate that a call has been connected. If the modem's CD signal is not activated before the time allotted with the **dialer wait-for-carrier-time** command, the router assumes that the call has failed and disconnects the line. Because the calls are international, and thus take longer to connect than local calls, the wait for carrier time is set to 60 seconds. Even for local calls, analog modems can take 20 to 30 seconds to synchronize to each other, including the time to dial and answer.

The **dialer string** command identifies the telephone number of the targeted destination. Because the central site is calling only a single destination, this dialer string is the simplest possible configuration. The **pulse-time** command specifies how long Data Terminal Ready (DTR) is held inactive. When using DDR and V.25bis modems, the router disconnects calls by deactivating DTR. This command is automatically inserted into the configuration when the **dialer in-band** command is entered.

The **dialer-group** command is used to identify each interface with a dialer list set. The **dialer-list** command associates each interface with access lists that determine which packets are "interesting" versus "uninteresting" for an interface. For details on access lists and dialer lists, see the "Access List Configuration" section that follows.

Routing Configuration

The Interior Gateway Routing Protocol (IGRP) is used to route traffic on the network. The first two commands in the routing section of the configuration file are **router igrp** and **network**. These define the IGRP number and the network over which IGRP runs.

The **redistribute** command causes the static route information (defined with the **ip route** commands shown in the configuration example) to be sent to other routers in the same IGRP area. Without this command, other routers connected to the central site will not have routes to the remote routers. The three static routes define the subnets on the Ethernet backbone of the remote routers. DDR tends to use static routes extensively because routing updates are not received when the dial-up connection is not active.

Access List Configuration

The last section of the configuration file provides the access lists that DDR uses to classify "interesting" and "uninteresting" packets. Interesting packets are packets that pass the restrictions of the access lists. These packets either initiate a call (if one is not already in progress) or reset the idle timer if a call is in progress. Uninteresting packets are transmitted if the link is active, but dropped if the link is not active. Uninteresting packets do not initiate calls or reset the idle timer. Access list 101 provides the following filters:

- IGRP packets that are sent to the broadcast address (255.255.255.255) do not cause dialing.
- All other IP packets are interesting and thus may cause dialing and reset the idle timer.

Remote Sites: Dial In Only

Except for the IP address and the default route, each of the remote sites is configured identically as an answer-only site. The following example lists Hong Kong's configuration:

```
interface serial 1
description interface to answer calls from San Francisco
ip address 128.10.200.65 255.255.255.192
dialer in-band
!
ip route 0.0.0.0 0.0.0.0 128.10.200.66
```

The answering site will not disconnect the call. It is up to the calling site to disconnect the call when the line is idle. In this case, the answering site is using static routing. The default route points to the serial interface at the central site.

Configuring a Single Interface for Multiple Remote Sites

It is possible to use a single interface to call multiple destinations, such as a site in Hong Kong and a site in Paris, France. Because of the time differences, these sites would never need to be connected at the same time. Therefore, a single interface could be used for both sites without the possibility of contention for the interface and without the cost of dedicating a serial port and modem to each destination.

Central Site: Dial Out Only

In the following configuration, the central site places the calls. A single interface is configured to call multiple remote sites. There is no support for answering calls in this configuration.

```
interface serial 5
description DDR connection to Hong Kong and Singapore
ip address 128.10.200.66 255.255.255.192
ip address 128.10.202.66 255.255.255.192 secondary
dialer in-band
dialer wait-for-carrier-time 60
! map Hong Kong to a phone number
dialer map ip 128.10.200.65 0118527351625
! map Singapore to a phone number
dialer map ip 128.10.202.65 011653367085
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
passive-interface serial 5
redistribute static
! route to Hong Kong
ip route 128.10.200.0 255.255.255.192 128.10.200.65
! route to Singapore
ip route 128.10.202.0 255.255.255.192 128.10.202.65
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
```

Interface Configuration

The configuration of the interface in this example is slightly more complicated than the configuration described in the "Configuring One Interface Per Remote Site" section. In addition to the original IP address, there is a secondary IP address configured for serial interface 5 because the Singapore and Hong Kong offices are on different subnets.

The **dialer in-band**, **dialer wait-for-carrier-time**, **pulse-time**, and **dialer-group** commands are used in the same manner as described previously in the "Configuring One Interface Per Remote Site" section. However, the previous **dialer string** command has been removed and replaced with two **dialer map** commands.

The first **dialer map** command maps the telephone number for Hong Kong to its next hop address, which is the IP address of the serial port of the router in Hong Kong. The second **dialer map** command maps the telephone number for the Singapore router to the next hop address for Singapore.

Routing Configuration

The IP static routes define the next hops used in the **dialer map** commands. When a packet is received for a host on network 128.10.200.0, it is routed to a next hop address of 128.10.200.65. This route goes out serial interface 5. DDR uses the next hop address to obtain the telephone number of the destination router.

Note The use of the **passive-interface** command states that routing updates are not to be sent out serial interface 5. Because the remote sites are using a default route, there is no need to send routing updates over the wire.

Access List Configuration

The use of **dialer map** commands provides an additional level of filtering. When a packet is received for a host on network 128.10.200.0, it is routed to a next hop address of 128.10.200.65. This route goes out serial interface 5. The packet is compared to the access lists. If the packet is deemed "interesting," the packet's next hop address is compared to the **dialer map** commands defined for that interface. If a match is found, the interface is checked to determine whether it is connected to the telephone number for that next hop address. If the interface is not connected, a call is placed to the telephone number. If the interface is currently connected to that number, the idle timer is reset. If the interface is connected to another number (from another **dialer map** command), the fast-idle timer is started due to contention for the interface. If there is no match of the next hop address to any of the dialer maps and there is no dialer string defined (which matches all next hop addresses), the packet is dropped.

This additional layer of filtering for the next hop address causes problems for broadcast packets such as routing updates. Because a broadcast packet is transmitted with a next hop address of the broadcast address, the check against the **dialer map** commands will fail. If you want broadcast packets transmitted to telephone numbers defined by **dialer map** commands, additional **dialer map** commands must specify the broadcast address as the next hop address with the same telephone number. For example, you might add the following **dialer map** commands:

```
dialer map ip 255.255.255.255 0118527351625
dialer map ip 255.255.255.255 011653367085
```

If the interface is currently connected to one of these telephone numbers, and if it receives an IGRP broadcast packet, that packet will now be transmitted because it matches a **dialer map** command to an already connected telephone number. (If the connection is already established, both "interesting" and “uninteresting” packets are sent.) If a connection is not already established, adding the **dialer map** commands will not cause an IGRP packet sent to the broadcast address to cause dialing because the access lists determine that the IGRP packet is uninteresting.

Note In the configuration example described in the “Configuring a Single Interface for Multiple Remote Sites” section, the **dialer string** command permits broadcast packets to be sent when the link is connected because the dialer string matches all next hop addresses that did not have a dialer map.

Remote Sites: Dial In Only

Except for the IP address and the default route, each of the remote sites is configured identically as an answer-only site. The following example illustrates the Hong Kong configuration:

```
interface serial 1
description interface to answer calls from San Francisco
ip address 128.10.200.65 255.255.255.192
dialer in-band
!
ip route 0.0.0.0 0.0.0.0 128.10.200.66
```

The answering site will not disconnect the call. It is up to the calling site to disconnect the call when the line is idle. A default route is defined back to the central site.

Configuring Multiple Interfaces for Multiple Remote Sites

When using a single interface with dialer maps, contention for the interface can occur. This contention starts a fast-idle timer that causes lines to remain connected for a shorter idle time than usual, allowing other destinations to use the interface. Dialer rotary groups prevent contention by creating a set of interfaces that can be used to dial out. Rather than statically assigning an interface to a destination, dialer rotary groups allow dynamic allocation of interfaces to telephone numbers. Before a call is placed, the rotary group is searched for an interface that is not in use to place the call. It is not until all of the interfaces in the rotary group are in use that the fast-idle timer is started.

Note The following configurations appear as they would be entered at the command line. Due to the way dialer rotary groups function, the output from a **write terminal** command on the router may differ slightly from what is shown here.

Central Site: Dial Out Only

The following configuration defines dialer rotary groups on the central site router:

```
interface dialer 1
description rotary group for Hong Kong, Tokyo, and Singapore
ip address 128.10.200.66 255.255.255.192
ip address 128.10.202.66 255.255.255.192 secondary
ip address 128.10.204.66 255.255.255.192 secondary
dialer in-band
dialer wait-for-carrier-time 60
! map Hong Kong to a phone number
dialer map ip 128.10.200.65 0118527351625
! map Singapore to a phone number
dialer map ip 128.10.202.65 011653367085
! map Tokyo to a phone number
dialer map ip 128.10.204.65 0118127351625
pulse-time 1
dialer-group 1
!
interface serial 5
dialer rotary-group 1
!
interface serial 6
dialer rotary-group 1
!
router igrp 1
network 128.10.0.0
passive-interface dialer 1
redistribute static
!
! route to Hong Kong
ip route 128.10.200.0 255.255.255.192 128.10.200.65
! route to Singapore
ip route 128.10.202.0 255.255.255.192 128.10.202.65
! route to Tokyo
ip route 128.10.204.0 255.255.255.192 128.10.204.65
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
```

Interface Configuration

Specifying a dialer interface is the first step in defining a dialer rotary group. While a dialer interface is not a physical interface, all of the configuration commands that can be specified for a physical interface can be used for a dialer interface. For example, the commands listed under the **interface dialer** command are identical to those used for physical serial interface 5 as described in the "Configuring a Single Interface for Multiple Remote Sites" section. Also, an additional **dialer map** command has been added to map the next hop address for Tokyo to the telephone number.

The **dialer rotary-group** command places physical serial interface 5 and serial interface 6 in the rotary group. Either of these interfaces can be used to dial any of the destinations defined by the **interface dialer** command.

As mentioned earlier, when you look at the configuration on the router using the **write terminal** command, the configuration may look slightly different from your input. For example, the **pulse-time** command associated with the dialer interface will appear with all of the serial interfaces that were added with the **dialer rotary-group** command. Certain configuration information associated with the dialer interface is propagated to all of the interfaces that are in the rotary group.

Routing Configuration

The routing section of this configuration has not changed from the example in the "Configuring a Single Interface for Multiple Remote Sites" section. But if you were to examine the routing table for one of the remote networks using the **show ip route** command (for example, **show ip route 128.10.200.0**), you would see that the output interface for packets sent to this subnet is interface dialer 1. The actual physical interface over which the packet will be transmitted is not determined until the DDR steps described in the following paragraph are performed.

Before a packet is sent out the dialer interface, DDR checks to determine whether the packet is "interesting" or "uninteresting." DDR then checks the dialer map. Next, all of the physical interfaces in the rotary group are checked to determine whether they are connected to the telephone number. If an appropriate interface is found, the packet is sent out that physical interface. If an interface is not found and the packet is deemed interesting, the rotary group is scanned for an available physical interface. The first available interface found is used to place a call to the telephone number.

Note To use dynamic routing, in which two of the remote sites communicate with each other via the central site, the **no ip split-horizon** command is required and the **passive-interface** command must be removed.

Access List Configuration

This configuration uses the same access lists as the example in the "Configuring a Single Interface for Multiple Remote Sites" section. A default route is defined back to the central site.

Remote Sites: Dial In Only

Except for the IP address and the default route, each of the remote sites is configured identically as an answer-only site. The following example illustrates the Hong Kong configuration:

```
interface serial 1
description interface to answer calls from San Francisco
ip address 128.10.200.65 255.255.255.192
dialer in-band
!
ip route 0.0.0.0 0.0.0.0 128.10.200.66
```

The answering site will not disconnect the call. It is up to the calling site to disconnect the call when the line is idle.

Having the Central and Remote Sites Dial In and Dial Out

It is often more convenient to have the remote sites call the central site as its users require, instead of depending on the central site to poll the remote sites. This section provides the following configuration examples in which both the central site and the remote sites are placing calls:

- Configuring One Interface Per Remote Site
- Configuring a Single Interface for Multiple Remote Sites
- Configuring Multiple Interfaces for Multiple Remote Sites

Configuring One Interface Per Remote Site

In order to support dial-in and dial-out for both the central and remote sites using one interface per remote site, each remote site must call in on the specific central site interface that has the dialer string corresponding to the respective remote site telephone number.

Central Site: Dial In and Dial Out

In the following example, the central San Francisco site is configured to place and answer calls. One interface is configured per remote site.

```
interface serial 5
description DDR connection to Hong Kong
ip address 128.10.200.66 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 0118527351625
pulse-time 1
dialer-group 1
!
interface serial 6
description DDR connection to Singapore
ip address 128.10.202.66 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 011653367085
pulse-time 1
dialer-group 1
!
interface serial 7
description DDR connection to Tokyo
ip address 128.10.204.66 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 0118127351625
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
redistribute static
!
! route to Hong Kong
ip route 128.10.200.0 255.255.255.192 128.10.200.65
! route to Singapore
ip route 128.10.202.0 255.255.255.192 128.10.202.65
! route to Tokyo
ip route 128.10.204.0 255.255.255.192 128.10.204.65
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
```

Remote Sites: Dial In and Dial Out

All of the remote configurations are similar. Each defines a default route back to the central site and a dialer string that contains the telephone number of the central site.

Hong Kong

In the following example, the remote Hong Kong site is configured to place and answer calls. Hong Kong's configuration file contains a dialer string of 14155551212, which should call serial interface 5 in San Francisco.

```
interface serial 1
description DDR connection to San Francisco
ip address 128.10.200.65 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551212
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
!
ip route 128.10.0.0 255.255.0.0 128.10.200.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
```

Singapore

In the following example, the remote Singapore site is configured to place and answer calls. The Singapore configuration file contains a dialer string of 14155551213, which should call serial interface 6 in San Francisco.

```
interface serial 1
description DDR connection to San Francisco
ip address 128.10.202.65 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551213
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
!
ip route 128.10.0.0 255.255.0.0 128.10.202.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
```

Tokyo

In the following example, the remote Tokyo site is configured to place and answer calls. The Tokyo configuration file contains a dialer string of 14155551214, which should call serial interface 7 in San Francisco.

```
interface serial 1
description DDR connection to San Francisco
ip address 128.10.204.65 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551214
pulse-time 1
dialer-group 1
router igrp 1
network 128.10.0.0
!
ip route 128.10.0.0 255.255.0.0 128.10.204.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
```

Because all incoming calls are assumed to be from the telephone number configured with the **dialer string** command, it is important to configure the central and remote sites correctly. For example, if the Singapore dialer string uses the telephone number that Hong Kong uses to call the central site, packets from the central site intended for Hong Kong would be sent to Singapore whenever Singapore called in because Singapore called in using the Hong Kong interface.

Configuring a Single Interface for Multiple Remote Sites

When multiple sites are calling into a central site, an authentication mechanism must be used unless that central site has one interface dedicated to each incoming call. Without the authentication mechanism, the central site router has no way of identifying the sites to which it is currently connected and cannot ensure that additional calls are not made. Point-to-Point Protocol (PPP) encapsulation with CHAP or Password Authentication Protocol (PAP) provides the mechanism to identify the calling party.

Note A router with a built-in ISDN port may be able to use calling party identification. Because calling party identification is not available everywhere, PPP with CHAP provides the identification mechanism. In Software Release 9.21, PPP and Password Authentication Protocol (PAP) can be used in place of CHAP, although PAP is less secure than CHAP. The configuration of PAP would differ slightly from the configuration for CHAP illustrated in this section.

Central Site: Dial In and Dial Out

In the following example, the central San Francisco site is configured to place and answer calls. A single interface is configured for multiple remote sites.

```
hostname SanFrancisco
interface serial 5
description DDR connection to Hong Kong and Singapore
ip address 128.10.200.66 255.255.255.192
ip address 128.10.202.66 255.255.255.192 secondary
encapsulation ppp
ppp authentication chap
dialer in-band
dialer wait-for-carrier-time 60
dialer map ip 128.10.200.65 name HongKong 0118527351625
dialer map ip 128.10.202.65 name Singapore 011653367085
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
passive-interface serial 5
redistribute static
!
! route to Hong Kong
ip route 128.10.200.0 255.255.255.192 128.10.200.65
! route to Singapore
ip route 128.10.202.0 255.255.255.192 128.10.202.65
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username HongKong password password1
username Singapore password password2
```

The command **encapsulation ppp** enables PPP encapsulation. The command **ppp authentication chap** enables CHAP authentication. In addition, **username** commands are entered for each of the remote sites that place calls. The **username** command defines the name of the remote router and a password to be associated with that router. When **ppp authentication chap** is configured, authentication must be verified or else network traffic will not be transmitted.

The **dialer map** command contains the host name of the remote router. This associates the remote router with a next hop address and a telephone number. When a packet is received for a host on network 128.10.200.0, it is routed to a next hop address of 128.10.200.65 via serial interface 5. The packet is compared to the access lists and then the packet's next hop address is compared to the **dialer map** commands for serial interface 5.

If the packet is "interesting" and a connection to the number in the **dialer map** command is already active on the interface, the idle timer is reset. If a match is found, DDR checks the interface to determine whether it is connected to the telephone number for the next hop address. The comparison to the telephone number is useful only if the router placed the call or if the telephone number was received via calling party ID on an ISDN router. With CHAP and the **name** keyword included in the **dialer map** command, both the telephone number and the name for a given next hop address are compared to the names of the routers already connected. In this way, calls to destinations to which connections are already established can be avoided.

Remote Sites: Dial In and Dial Out

In the following configuration examples, the remote sites are configured to place and receive calls to or from a single interface at the central site.

Hong Kong

The following configuration allows Hong Kong to place and receive calls to and from the central site in San Francisco:

```
hostname HongKong
interface serial 1
description DDR connection to SanFrancisco
ip address 128.10.200.65 255.255.255.192
encapsulation ppp
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551212
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
!
ip route 128.10.0.0 255.255.0.0 128.10.200.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username SanFrancisco password password1
```

Singapore

The following configuration allows Singapore to place and receive calls to and from the central site in San Francisco:

```
hostname Singapore
interface serial 1
description DDR connection to San Francisco
ip address 128.10.202.65 255.255.255.192
encapsulation ppp
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551212
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
ip route 128.10.0.0 255.255.0.0 128.10.202.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username SanFrancisco password password2
```

Unlike the central site, the remote sites do not contain the **ppp authentication chap** command. This is because only one site, the central site, is calling in to the remote sites. If only one site is calling in, DDR assumes the call is from the number defined with the **dialer string** command; therefore, the command **ppp authentication chap** is not required.

Note If the remote sites use **dialer map** commands instead of **dialer string**, the **ppp authentication chap** command is required, and the **dialer map** commands require the **name** keyword. This is because the assumption is made that if the **dialer map** command is used, multiple sites either can be called or can call in.

Also, the remote sites have a **username** entry for the San Francisco router, and the San Francisco router contains the username passwords for Singapore and Hong Kong.

Configuring Multiple Interfaces for Multiple Remote Sites

The configurations in this section are similar to the examples provided in the earlier "Configuring a Single Interface for Multiple Remote Sites" section. The encapsulation is set to PPP and CHAP authentication is required.

Central Site: Dial In and Dial Out

The following example configures the central site router to dial in and dial out on multiple interfaces to multiple remote sites:

```
hostname SanFrancisco
interface dialer 1
description rotary group for Hong Kong, Tokyo, and Singapore
ip address 128.10.200.66 255.255.255.192
ip address 128.10.202.66 255.255.255.192 secondary
ip address 128.10.204.66 255.255.255.192 secondary
encapsulation ppp
ppp authentication chap
dialer in-band
dialer wait-for-carrier-time 60
dialer map ip 128.10.200.65 name HongKong 0118527351625
dialer map ip 128.10.202.65 name Singapore 011653367085
dialer map ip 128.10.204.65 name Tokyo 0118127351625
pulse-time 1
dialer-group 1
!
interface serial 5
dialer rotary-group 1
!
interface serial 6
dialer rotary-group 1
!
router igrp 1
network 128.10.0.0
passive-interface dialer 1
redistribute static
! route to Hong Kong
ip route 128.10.200.0 255.255.255.192 128.10.200.65
! route to Singapore
ip route 128.10.202.0 255.255.255.192 128.10.202.65
! route to Tokyo
ip route 128.10.204.0 255.255.255.192 128.10.204.65
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username HongKong password password1
username Singapore password password2
username Tokyo password password3
```

Remote Sites: Dial In and Dial Out

In the following configuration examples, the remote sites are configured to place and receive calls to or from multiple interfaces at the central site. All of the remote sites dial the same telephone number. At the San Francisco site, that single telephone number will connect to either serial interface 5 or serial interface 6. This capability is provided by the telephone service provider.

Hong Kong

The following configuration allows Hong Kong to place and receive calls to and from the central site in San Francisco:

```
hostname HongKong
interface serial 1
description DDR connection to SanFrancisco
ip address 128.10.200.65 255.255.255.192
encapsulation ppp
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551212
pulse-time 1
dialer-group 1
router igrp 1
network 128.10.0.0
ip route 128.10.0.0 255.255.0.0 128.10.200.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username SanFrancisco password password1
```

Singapore

The following configuration allows Singapore to place and receive calls to and from the central site in San Francisco:

```
hostname Singapore
interface serial 1
description DDR connection to San Francisco
ip address 128.10.202.65 255.255.255.192
encapsulation ppp
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551212
pulse-time 1
dialer-group 1
router igrp 1
network 128.10.0.0
ip route 128.10.0.0 255.255.0.0 128.10.202.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username SanFrancisco password password2
```

Tokyo

The following configuration allows Tokyo to place and receive calls to and from the central site in San Francisco:

```
hostname Tokyo
interface serial 1
description DDR connection to San Francisco
ip address 128.10.204.65 255.255.255.192
encapsulation ppp
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551212
pulse-time 1
dialer-group 1
router igrp 1
network 128.10.0.0
ip route 128.10.0.0 255.255.0.0 128.10.204.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username SanFrancisco password password3
```

The remote sites do not use the **ppp authentication chap**. This is because only one site, the central site, is calling in to the remote sites. If only one site is calling in, DDR assumes the call is from the number defined with the **dialer string** command; therefore, the command **ppp authentication chap** is not required. However, if the remote sites use **dialer map** commands instead of **dialer string**, the **ppp authentication chap** command is required, and the **dialer map** commands require the **name** keyword.

Also, each remote site has a **username SanFrancisco** entry containing the same password that the central San Francisco site uses to identify the remote site.

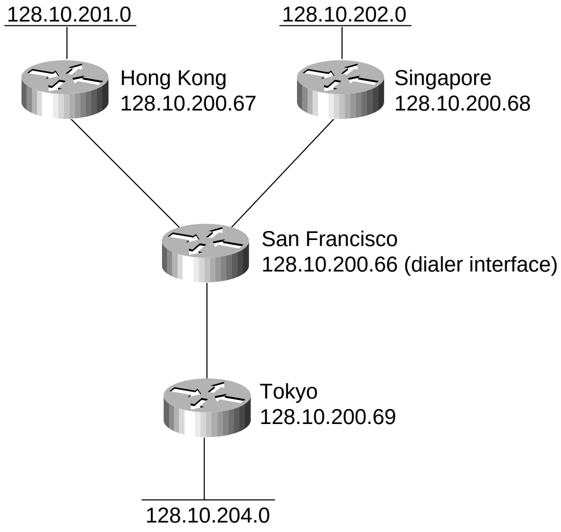
Having Remote Sites Dial Out

A common configuration is to have the remote sites place calls to the central site, which does not dial out.

Configuring Multiple Interfaces for Multiple Remote Sites

In a “star” topology, all the remote routers can have their serial interfaces on the same subnet as the central site serial interface. (See Figure 2-2.)

Figure 2-2 Remote sites dial out (star topology).



Central Site: Dial In Only

The following example configures the central site router to accept dial-ins on multiple interfaces:

```
hostname SanFrancisco
interface dialer 1
description rotary group for inbound calls
ip address 128.10.200.66 255.255.255.192
encapsulation ppp
ppp authentication chap
dialer in-band
dialer wait-for-carrier-time 60
dialer map ip 128.10.200.67 name HongKong
dialer map ip 128.10.200.68 name Singapore
dialer map ip 128.10.200.69 name Tokyo
pulse-time 1
dialer-group 1
!
interface serial 5
dialer rotary-group 1
!
interface serial 6
dialer rotary-group 1
!
router igrp 1
network 128.10.0.0
passive-interface dialer 1
redistribute static
! route to Hong Kong
ip route 128.10.201.0 255.255.255.192 128.10.200.67
! route to Singapore
ip route 128.10.202.0 255.255.255.192 128.10.200.68
! route to Tokyo
ip route 128.10.204.0 255.255.255.192 128.10.200.69
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username HongKong password password1
username Singapore password password2
username Tokyo password password3
```

Remote Sites: Dial Out Only

In the following configurations, the remote sites are configured to place calls to multiple interfaces at the central site. The assumption here is that a single telephone number on the central site will get any one of two possible inbound serial interfaces (serial interface 5 or serial interface 6).

Hong Kong

The following configuration allows Hong Kong to place calls to the central site in San Francisco:

```
hostname HongKong
interface ethernet 0
ip address 128.10.201.1 255.255.255.192
interface serial 1
description DDR connection to SanFrancisco
ip address 128.10.200.67 255.255.255.192
encapsulation ppp
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551212
pulse-time 1
dialer-group 1
router igrp 1
network 128.10.0.0
ip route 128.10.0.0 255.255.0.0 128.10.200.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username SanFrancisco password password1
```

Singapore

The following configuration allows Singapore to place calls to the central site in San Francisco:

```
hostname Singapore
interface ethernet 0
ip address 128.10.202.1 255.255.255.192
interface serial 1
description DDR connection to San Francisco
ip address 128.10.200.68 255.255.255.192
encapsulation ppp
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551212
pulse-time 1
dialer-group 1
router igrp 1
network 128.10.0.0
ip route 128.10.0.0 255.255.0.0 128.10.200.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username SanFrancisco password password2
```

Tokyo

The following configuration allows Tokyo to place calls to the central site in San Francisco:

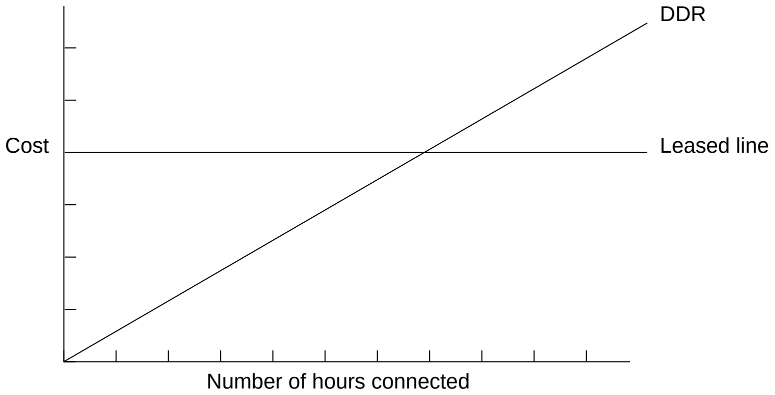
```
hostname Tokyo
interface ethernet 0
ip address 128.10.204.1 255.255.255.192
interface serial 1
description DDR connection to San Francisco
ip address 128.10.200.69 255.255.255.192
encapsulation ppp
dialer in-band
dialer wait-for-carrier-time 60
dialer string 14155551212
pulse-time 1
dialer-group 1
router igrp 1
network 128.10.0.0
ip route 128.10.0.0 255.255.0.0 128.10.200.66
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
!
username SanFrancisco password password3
```

Using DDR as a Backup to Leased Lines

DDR allows you to quickly enable a WAN connection through the use of existing analog telephone lines. Also, DDR provides cost savings because the line is used on an as-needed basis, whereas a leased line is paid for when the line is not in use. However, there are times when a leased line may provide benefits.

Figure 2-3 shows that there can be a point (when a connection needs to be maintained for more than a certain number of hours per day) at which a DDR link no longer has cost savings, and a leased line may be more cost effective. Additionally, DDR links have a variable cost. It is difficult to predict what a DDR link may cost per month, given that users can initiate traffic at any time.

Figure 2-3 DDR-to-Leased Line Cutover.



With leased lines, you can still continue to use dial-up lines as a backup by using either of the following methods:

- Floating static routes (single and shared interfaces) and DDR
- DTR dialing or V.25bis dialing

Floating Static Routes

Floating static routes are static routes that have an administrative distance greater than the administrative distance of dynamic routes. Administrative distances can be configured on a static route so that the static route is less desirable than a dynamic route. In this manner, the static route is not used when the dynamic route is available. However, if the dynamic route is lost, the static route can take over, and traffic can be sent through this alternative route. If this alternative route is provided by a DDR interface, DDR can be used as a backup mechanism.

Central Site

The following example outlines a configuration of a central site using leased lines for primary connectivity and DDR for backup:

```
interface serial 1
description Leased connection to Hong Kong
ip address 128.10.200.66 255.255.255.192
!
interface serial 2
description leased connection to Singapore
ip address 128.10.202.66 255.255.255.192
!
interface serial 5
description backup DDR connection to Hong Kong
ip address 128.10.200.130 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 0118527351625
pulse-time 1
dialer-group 1
!
interface serial 6
description backup DDR connection to Singapore
ip address 128.10.202.130 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 011653367085
pulse-time 1
dialer-group 1
!
interface serial 7
description DDR connection to Tokyo
ip address 128.10.204.66 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 0118127351625
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
redistribute static
!
! route to Hong Kong with administrative distance
ip route 128.10.200.0 255.255.255.192 128.10.200.129 150
! route to Singapore with administrative distance
ip route 128.10.202.0 255.255.255.192 128.10.202.129 150
! route to Tokyo
ip route 128.10.204.0 255.255.255.192 128.10.204.65
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
```

Serial interfaces 1 and 2 are used as leased lines to Hong Kong and Singapore. Serial interface 5 backs up serial interface 1; serial interface 6 backs up serial interface 2; and serial interface 7 is used for DDR to Tokyo.

Remote Sites

Each remote sites has a leased line as a primary link and a DDR line as a backup. For example:

```
interface serial 0
description leased line from San Francisco
ip address 128.10.200.65 255.255.255.192
!
interface serial 1
description interface to answer backup calls from San Francisco
ip address 128.10.200.129 255.255.255.192
dialer in-band
!
router igrp 1
network 128.10.0.0
! route back to San Francisco with administrative distance
ip route 128.10.0.0 255.255.0.0 128.10.200.130 150
```

The first serial interface is the leased line, whereas the second answers calls from the central site in case the central site needs to use DDR as a backup method.

Floating Static Routes on Shared Interfaces

The central site configuration requires a large number of serial ports because each primary port has its own backup. For true redundancy, backup is a requirement. But in many cases, an interface or a set of interfaces can be shared as backup for a set of primary lines. The following configuration shows how to set up a single interface to back up all of the primary lines:

```
interface serial 1
description Leased connection to Hong Kong
ip address 128.10.200.66 255.255.255.192
!
interface serial 2
description leased connection to Singapore
ip address 128.10.202.66 255.255.255.192
!
interface serial 5
description backup DDR connection for all destinations except Tokyo
ip address 128.10.200.130 255.255.255.192
ip address 128.10.202.130 255.255.255.192 secondary
dialer in-band
dialer wait-for-carrier-time 60
! map Hong Kong to a phone number
dialer map ip 128.10.200.129 0118527351625
! map Singapore to a phone number
dialer map ip 128.10.202.129 011653367085
pulse-time 1
dialer-group 1
!
interface serial 7
description DDR connection to Tokyo
ip address 128.10.204.66 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 0118127351625
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
passive-interface serial 5
redistribute static
!
! route to Hong Kong with administrative distance
ip route 128.10.200.0 255.255.255.192 128.10.200.129 150
! route to Singapore with administrative distance
ip route 128.10.202.0 255.255.255.192 128.10.202.129 150
! route to Tokyo
ip route 128.10.204.0 255.255.255.192 128.10.204.65
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
```

Serial interface 5 is the DDR backup interface for all destinations and is configured with multiple IP addresses for routing. The **dialer map** commands map the next hop addresses to the telephone numbers for each of the destinations. If a dynamic route is lost, the floating static route takes over. The next hop address sends the packets to serial interface 5, where the **dialer map** commands place the telephone call.

If two primary lines fail at the same time, there will be contention to use serial interface 5. The fast-idle timer may disconnect the calls. If serial interface 5 were in constant use, one of the primary lines would be disconnected and packets would be dropped. The fact that the backup route is unavailable is not communicated because there is no way to announce that one of the two IP addresses on the interface are unavailable. If you use a dialer rotary group, the contention problem can be avoided.

Using Leased Lines and Dial Backup

This section describes how to use the following two methods for dial backup with leased lines:

- DTR Dialing
- V.25bis Dialing

DTR Dialing

Since Software Release 8.3, a dial backup capability has been provided. Although it is somewhat more restrictive than floating static routes, dial backup can be used if V.25bis modems are not available or if protocols that do not have support for floating static routes are used.

Central Site

Dial backup requires that the modems place a call when the Data Terminal Ready (DTR) signal is raised. The telephone number is configured into the modem or other DCE device. That number is called when DTR is raised. The call is disconnected when DTR is lowered. The following configuration illustrates how to take advantage of dial backup and DTR dialing:

```
interface serial 1
description Leased connection to Hong Kong
ip address 128.10.200.66 255.255.255.192
backup interface serial 4
backup delay 0 20
!
interface serial 2
description leased connection to Singapore
ip address 128.10.202.66 255.255.255.192
backup interface serial 5
backup delay 0 20
!
interface serial 4
description backup connection for Hong Kong
ip address 128.10.200.67 255.255.255.192
pulse-time 10
!
interface serial 5
description backup connection for Singapore
ip address 128.10.202.67 255.255.255.192
pulse-time 10
!
interface serial 7
description DDR connection to Tokyo
ip address 128.10.204.66 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 0118127351625
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
```

This solution requires one serial port per primary line. Because the backup ports are placed on the same subnet as the primary serial port, no static routes are required. The **backup delay** command is used to specify how long to wait after the primary has failed before activating the backup line, and how long to delay before deactivating the backup line after the primary line comes back up. In this case, the primary link will be active for 20 seconds before disabling the backup line. This delay allows for flapping in the primary link when it returns to functioning.

Remote Sites

For the remote sites, the floating static route is not needed. The IP address of the backup interface must be on the same subnet as the primary interface. The following example illustrates the Hong Kong router configuration. Serial interface 0 is the leased line, whereas serial interface 1 answers calls as a backup method:

```
interface serial 0
description leased line from San Francisco
ip address 128.10.200.65 255.255.255.192
!
interface serial 1
description interface to answer backup calls from San Francisco
ip address 128.10.200.68 255.255.255.192
!
router igrp 1
network 128.10.0.0
```

V.25bis Dialing

V.25bis dialing capability can be preferable to DTR dialing when multiple telephone numbers are required. Using DTR dialing, most devices will call only a single number. With V.25bis, the router can attempt to call several numbers if the first number does not answer. The following configuration illustrates V.25bis dialing:

```
interface serial 1
description Leased connection to Hong Kong
ip address 128.10.200.66 255.255.255.192
backup interface serial 4
backup delay 0 20
!
interface serial 2
description leased connection to Singapore
ip address 128.10.202.66 255.255.255.192
backup interface serial 5
backup delay 0 20
!
interface serial 4
description backup connection for Hong Kong
ip address 128.10.200.67 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer map IP 128.10.200.68 0118527351625
dialer map IP 128.10.200.68 0118527351872
dialer-group 1
pulse-time 1
!
interface serial 5
description backup connection for Singapore
ip address 128.10.202.67 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 011653367085
dialer-group 1
pulse-time 1
!
interface serial 7
description DDR connection to Tokyo
ip address 128.10.204.66 255.255.255.192
dialer in-band
dialer wait-for-carrier-time 60
dialer string 0118127351625
pulse-time 1
dialer-group 1
!
router igrp 1
network 128.10.0.0
redistribute static
!
! route to Hong Kong
ip route 128.10.200.0 255.255.255.192 128.10.200.68
! route to Singapore
ip route 128.10.202.0 255.255.255.192 128.10.202.68
! route to Tokyo
ip route 128.10.204.0 255.255.255.192 128.10.204.65
!
dialer-list 1 protocol IP PERMIT
```

Multiple telephone numbers are configured for serial interface 4. The two **dialer map** commands have the same next hop address. The software first attempts to call the telephone number specified in the first **dialer map** command. If this number fails—that is, if no connection is made before the wait-for-carrier timer expires—the second number is dialed. Each of the other backup interfaces uses a dialer string for the backup telephone number. When using V.25bis with dial backup, the **dialer-list protocol** command shown in the preceding example should be used. The dialer list states that all IP traffic is interesting and will, therefore, cause dialing. Routing updates are included. When a serial line is used as a backup, it is normally the state of the primary link, not the fast-idle timer, that determines when to disconnect the call.

Summary

As this case study indicates, there are many ways that dial-on-demand routing (DDR) can be used both for primary access and backup access. Sites can place calls, receive calls, and both place and receive calls. Additionally, using dialer rotary groups provides increased flexibility.

Increasing Security on IP Networks

Network security is a broad topic that can be addressed at the *data* link, or media, level (where packet snooping and encryption problems can occur), at the *network,* or protocol, layer (the point at which Internet Protocol (IP) packets and routing updates are controlled), and at the *application* layer (where, for example, host-level bugs become issues).

As more users access the Internet and as companies expand their networks, the challenge to provide security for internal networks becomes increasingly difficult. Companies must determine which areas of their internal networks they must protect, learn how to restrict user access to these areas, and determine which types of network services they should filter to prevent potential security breaches.

Cisco Systems provides several network, or protocol, layer features to increase security on IP networks. These features include controls to restrict access to routers and communication servers by way of console port, Telnet, Simple Network Management Protocol (SNMP), Terminal Access Controller Access Control System (TACACS), vendor token cards, and access lists. Firewall architecture setup is also discussed.

Caution Although this case study addresses network-layer security issues, which are the most relevant in the context of an Internet connection, ignoring host-level security, even with network-layer filtering in place, can be dangerous. For host-level security measures, refer to your application's documentation and the recommended reading list at the end of this case study.

Understanding Cisco's Approach to Network Security

When most people talk about security, they mean ensuring that users can only perform tasks they are authorized to do, can only obtain information they are authorized to have, and cannot cause damage to the data, applications, or operating environment of a system.

The word *security* connotes protection against malicious attack by outsiders. Security also involves controlling the effects of errors and equipment failures. Anything that can protect against a deliberate, intelligent, calculated attack will probably prevent random misfortune as well.

Security measures keep people honest in the same way that locks do. This case study provides specific actions you can take to improve the security of your network. Before going into specifics, however, it will help if you understand the following basic concepts that are essential to any security system:

- *Know your enemy*

  This case study refers to *attackers* or *intruders*. Consider who might want to circumvent your security measures and identify their motivations. Determine what they might want to do and the damage that they could cause to your network.

Security measures can never make it impossible for a user to perform unauthorized tasks with a computer system. They can only make it harder. The goal is to make sure the network security controls are beyond the attacker's ability or motivation.

- *Count the cost*

  Security measures almost always reduce convenience, especially for sophisticated users. Security can delay work and create expensive administrative and educational overhead. It can use significant computing resources and require dedicated hardware.

  When you design your security measures, understand their costs and weigh those costs against the potential benefits. To do that, you must understand the costs of the measures themselves and the costs and likelihoods of security breaches. If you incur security costs out of proportion to the actual dangers, you have done yourself a disservice.

- *Identify your assumptions*

  Every security system has underlying assumptions. For example, you might assume that your network is not tapped, or that attackers know less than you do, that they are using standard software, or that a locked room is safe. Be sure to examine and justify your assumptions. Any hidden assumption is a potential security hole.

- *Control your secrets*

  Most security is based on secrets. Passwords and encryption keys, for example, are secrets. Too often, though, the secrets are not really all that secret. The most important part of keeping secrets is knowing the areas you need to protect. What knowledge would enable someone to circumvent your system? You should jealously guard that knowledge and assume that everything else is known to your adversaries. The more secrets you have, the harder it will be to keep all of them. Security systems should be designed so that only a limited number of secrets need to be kept.

- *Remember human factors*

  Many security procedures fail because their designers do not consider how users will react to them. For example, because they can be difficult to remember, automatically generated "nonsense" passwords are often found written on the undersides of keyboards. For convenience, a "secure" door that leads to the system's only tape drive is sometimes propped open. For expediency, unauthorized modems are often connected to a network to avoid onerous dial-in security measures.

  If your security measures interfere with essential use of the system, those measures will be resisted and perhaps circumvented. To win compliance, you must make sure that users can get their work done, and you must sell your security measures to users. Users must understand and accept the need for security.

  Any user can compromise system security, at least to some degree. Passwords, for instance, can often be found simply by calling legitimate users on the telephone, claiming to be a system administrator, and asking for them. If your users understand security issues, and if they understand the reasons for your security measures, they are far less likely to make an intruder's life easier.

  At a minimum, users should be taught never to release passwords or other secrets over unsecured telephone lines (especially cellular telephones) or electronic mail (email). Users should be wary of questions asked by people who call them on the telephone. Some companies have implemented formalized network security training for their employees; that is, employees are not allowed access to the Internet until they have completed a formal training program.

- *Know your weaknesses*

  Every security system has vulnerabilities. You should understand your system's weak points and know how they could be exploited. You should also know the areas that present the largest danger and prevent access to them immediately. Understanding the weak points is the first step toward turning them into secure areas.

- *Limit the scope of access*

  You should create appropriate barriers inside your system so that if intruders access one part of the system, they do not automatically have access to the rest of the system. The security of a system is only as good as the weakest security level of any single host in the system.

- *Understand your environment*

  Understanding how your system normally functions, knowing what is expected and what is unexpected, and being familiar with how devices are usually used, help you to detect security problems. Noticing unusual events can help you to catch intruders before they can damage the system. Auditing tools can help you to detect those unusual events.

- Limit your trust

  You should know exactly which software you rely on, and your security system should not have to rely upon the assumption that all software is bug-free.

- *Remember physical security*

  Physical access to a computer (or a router) usually gives a sufficiently sophisticated user total control over that computer. Physical access to a network link usually allows a person to tap that link, jam it, or inject traffic into it. It makes no sense to install complicated software security measures when access to the hardware is not controlled.

- *Security is pervasive*

  Almost any change you make in your system may have security effects. This is especially true when new services are created. Administrators, programmers, and users should consider the security implications of every change they make. Understanding the security implications of a change is something that takes practice. It requires lateral thinking and a willingness to explore every way in which a service could potentially be manipulated.

Controlling Access to Cisco Routers

It is important to control access to your Cisco routers. You can control access to the router using the following methods:

- Console Access
- Telnet Access
- Simple Network Management Protocol (SNMP) Access
- Controlling Access to Network Servers That Contain Configuration Files

You can secure the first three of these methods by employing features within the router software. For each method, you can permit nonprivileged access and privileged access for a user (or group of users). Nonprivileged access allows users to monitor the router, but not to configure the router. Privileged access allows the user to fully configure the router.

For console port and Telnet access, you can set up two types of passwords. The first type of password, the login password, allows the user nonprivileged access to the router. After accessing the router, the user can enter privileged mode by entering the **enable** command and the proper password. Privileged mode provides the user with full configuration capabilities.

SNMP access allows you to set up different SNMP community strings for both nonprivileged and privileged access. Nonprivileged access allows users on a host to send the router SNMP get-request and SNMP get-next-request messages. These messages are used for gathering statistics from the router. Privileged access allows users on a host to send the router SNMP set-request messages in order to make changes to the router's configurations and operational state.

Console Access

A console is a terminal attached directly to the router via the console port. Security is applied to the console by asking users to authenticate themselves via passwords. By default, there are no passwords associated with console access.

Nonprivileged Mode Password

You configure a password for nonprivileged mode by entering the following commands in the router's configuration file. Passwords are case-sensitive. In this example, the password is “1forAll.”

```
line console 0
login
password 1forAll
```

When you log in to the router, the router login prompt is as follows:

```
User Access Verification
Password:
```

You must enter the password “1forAll” to gain nonprivileged access to the router. The router response is as follows:

```
router>
```

Nonprivileged mode is signified on the router by the > prompt. At this point, you can enter a variety of commands to view statistics on the router, but you cannot change the configuration of the router. Never use “cisco,” or other obvious derivatives, such as “pancho,” for a Cisco router password. These will be the first passwords intruders will try if they recognize the Cisco login prompt.

Privileged Mode Password

Configure a password for privileged mode by entering the following commands in the router's configuration file. In this example, the password is “san-fran.”

```
enable-password san-fran
```

To access privileged mode, enter the following command:

```
router> enable
Password:
```

Enter the password “san-fran” to gain privileged access to the router. The router responds as follows:

```
router#
```

Privileged mode is signified by the `#` prompt. In privileged mode, you can enter all commands to view statistics and configure the router.

Session Timeouts

Setting the login and enable passwords may not provide enough security in some cases. The timeout for an unattended console (by default 10 minutes) provides an additional security measure. If the console is left unattended in privileged mode, any user can modify the router's configuration. You can change the login timeout via the command **exec-timeout** *mm ss* where *mm* is minutes and *ss* is seconds. The following commands change the timeout to 1 minute and 30 seconds:

```
line console 0
exec-timeout 1 30
```

Password Encryption

All passwords on the router are visible via the **write terminal** and **show configuration** privileged mode commands. If you have access to privileged mode on the router, you can view all passwords in cleartext by default.

There is a way to hide cleartext passwords. The command **service password-encryption** stores passwords in an encrypted manner so that anyone performing a **write terminal** and **show configuration** will not be able to determine the cleartext password. However, if you forget the password, regaining access to the router requires you to have physical access to the router.

Note Although encryption is helpful, it can be compromised and thus should not be your only network-security strategy.

Telnet Access

You can access both nonprivileged and privileged mode on the router via Telnet. As with the console port, Telnet security is provided when users are prompted by the router to authenticate themselves via passwords. In fact, many of the same concepts described in the "Console Access" section earlier in this chapter apply to Telnet access. You must enter a password to go from nonprivileged mode to privileged mode, and you can encrypt passwords and specify timeouts for each Telnet session.

Nonprivileged Mode Password

Each Telnet port on the router is known as a *virtual terminal*. There are a maximum of five virtual terminal (VTY) ports on the router, allowing five concurrent Telnet sessions. (The communication server provides more VTY ports.) On the router, the virtual terminal ports are numbered from 0 through 4. You can set up nonprivileged passwords for Telnet access via the virtual terminal ports with the following configuration commands. In this example, virtual terminal ports 0 through 4 use the password "marin":

```
line vty 0 4
login
password marin
```

When a user telnets to a router IP address, the router provides a prompt similar to the following:

```
% telnet router
Trying ...
Connected to router.
Escape character is '^]'.
User Access Verification
Password:
```

If the user enters the correct nonprivileged password, the following prompt appears:

```
router>
```

Privileged Mode Password

The user now has nonprivileged access to the router and can enter privileged mode by entering the **enable** command as described in the "Privileged Mode Password" section earlier in this chapter.

Restricting Telnet Access to Particular IP Addresses

If you want to allow only certain IP addresses to use Telnet to access the router, you must use the **access-class** command. The command **access-class** *nn* **in** defines an access list (from 1 through 99) that allows access to the virtual terminal lines on the router. The following configuration commands allow incoming Telnet access to the router only from hosts on network 192.85.55.0:

```
access-list 12 permit 192.85.55.0 0.0.0.255
line vty 0 4
access-class 12 in
```

Restricting Telnet Access to Cisco Products via TCP Ports

It is possible to access Cisco products via Telnet to specified TCP ports. The type of Telnet access varies, depending upon the following Cisco software releases:

- Software Release 9.1 (11.4) and earlier and 9.21 (3.1) and earlier
- Software Release 9.1 (11.5), 9.21 (3.2), and 10.0 and later

Earlier Software Releases

For Software Release 9.1 (11.4) and earlier and Software Release 9.21 (3.1) and earlier, it is possible, by default, to establish TCP connections to Cisco products via the TCP ports listed in Table 3-1.

Table 3-1 TCP Port Telnet Access to Cisco Products (Earlier Releases)

TCP Port Number	Access Method
7	Echo
9	Discard
23	Telnet (to virtual terminal VTY ports in rotary fashion)
79	Finger
1993	SNMP over TCP
2001 through 2999	Telnet to auxiliary (AUX) port, terminal (TTY) ports, and virtual terminal (VTY) ports
3001 through 3999	Telnet to rotary ports (access via these ports is only possible if the rotaries have been explicitly configured first with the **rotary** command)
4001 through 4999	Telnet (stream mode) mirror of 2000 range
5001 through 5999	Telnet (stream mode) mirror of 3000 range (access via these ports is possible only if the rotaries have been explicitly configured first)
6001 through 6999	Telnet (binary mode) mirror of 2000 range
7001 through 7999	Telnet (binary mode) mirror of 3000 range (access via these ports is possible only if the rotaries have been explicitly configured first)
8001 through 8999	Xremote (communication servers only)
9001 through 9999	Reverse Xremote (communication servers only)
10001 through 19999	Reverse Xremote rotary (communication servers only; access via these ports is possible only if the ports have been explicitly configured first)

Caution Because Cisco routers have no TTY lines, configuring access (on communication servers) to terminal ports 2002, 2003, 2004, and greater could potentially provide access (on routers) to virtual terminal lines 2002, 2003, 2004, and greater. To provide access only to TTY ports, you can create access lists to prevent access to VTYs.

When configuring rotary groups, keep in mind that access through any available port in the rotary group is possible (unless access lists are defined). Cisco recommends that if you are using firewalls that allow in-bound TCP connection to high-number ports, remember to apply appropriate in-bound access lists to Cisco products.

The following is an example illustrating an access list denying all in-bound Telnet access to the auxiliary port and allowing Telnet access to the router only from IP address 192.32.6.7:

```
access-class 51 deny 0.0.0.0 255.255.255.255
access-class 52 permit 192.32.6.7
line aux 0
access-class 51 in
line vty 0 4
access-class 52 in
```

To disable connections to the echo and discard ports, you must disable these services completely with the **no service tcp-small-servers** command.

Caution If the **ip alias** command is enabled on Cisco products, TCP connections to any destination port are considered valid connections. You may want to disable the **ip alias** command.

You might want to create access lists to prevent access to Cisco products via these TCP ports. For information on how to create access lists for routers, see the “Configuring the Firewall Router” section later in this chapter. For information on how to create access lists for communication servers, see the “Configuring the Firewall Communication Server” section later in this chapter.

Software Releases 9.1 (11.5), 9.21 (3.2), and 10.0 and Later

With Software Release 9.1 (11.5), 9.21 (3.2), and any version of Software Release 10, the following enhancements have been implemented:

- Direct access to virtual terminal lines (VTYs) through the 2000, 4000, and 6000 port ranges has been disabled. If you want to keep access open, you can set up one-to-one mapping of VTY-to-rotary ports.
- Connections to echo and discard ports (7 and 9) can be disabled with the **no service tcp-small-servers** command.
- All Cisco products allow connections to IP alias devices only on destination port 23.

For later releases, a Cisco router accepts TCP connections on the ports listed in Table 3-2 by default.

Table 3-2 TCP Port Telnet Access to Cisco Products (Later Releases)

TCP Port Number	Access Method
7	Echo
9	Discard
23	Telnet
79	Finger
1993	SNMP over TCP

TCP Port Number	Access Method
2001	Auxiliary (AUX) port
4001	Auxiliary (AUX) port (stream)
6001	Auxiliary (AUX) port (binary)

Access via port 23 can be restricted by creating an access list and assigning it to virtual terminal lines. Access via port 79 can be disabled with the **no service finger** command. Access via port 1993 can be controlled with SNMP access lists. Access via ports 2001, 4001, and 6001 can be controlled with an access list placed on the auxiliary port.

Terminal Access Controller Access Control System (TACACS)

Nonprivileged and privileged mode passwords are global and apply to every user accessing the router from either the console port or from a Telnet session. As an alternative, the Terminal Access Controller Access Control System (TACACS) provides a way to validate every user on an individual basis before they can gain access to the router or communication server. TACACS was derived from the United States Department of Defense and is described in Request For Comments (RFC) 1492. TACACS is used by Cisco to allow finer control over who can access the router in nonprivileged and privileged mode.

With TACACS enabled, the router prompts the user for a username and a password. Then, the router queries a TACACS server to determine whether the user provided the correct password. A TACACS server typically runs on a UNIX workstation. Public domain TACACS servers can be obtained via anonymous ftp to *ftp.cisco.com* in the */pub* directory. Use the */pub/README* file to find the filename. A fully supported TACACS server is bundled with CiscoWorks Version 3.

The configuration command **tacacs-server host** specifies the UNIX host running a TACACS server that will validate requests sent by the router. You can enter the **tacacs-server host** command several times to specify multiple TACACS server hosts for a router.

Nonprivileged Access

If all servers are unavailable, you may be locked out of the router. In that event, the configuration command **tacacs-server last-resort [password | succeed]** allows you to determine whether to allow a user to log in to the router with no password (**succeed** keyword) or to force the user to supply the standard login password (**password** keyword).

The following commands specify a TACACS server and allow a login to succeed if the server is down or unreachable:

```
tacacs-server host 129.140.1.1
tacacs-server last-resort succeed
```

To force users who access the router via Telnet to authenticate themselves using TACACS, enter the following configuration commands:

```
line vty 0 4
login tacacs
```

Privileged Access

This method of password checking can also be applied to the privileged mode password with the **enable use-tacacs** command. If all servers are unavailable, you may be locked out of the router. In that event, the configuration command **enable last-resort [succeed | password]** allows you to determine whether to allow a user to log in to the router with no password (**succeed** keyword) or to force the user to supply the enable password (**password** keyword). There are significant risks to using the **succeed** keyword. If you use the **enable use-tacacs** command, you must also specify the **tacacs-server authenticate enable** command.

The **tacacs-server extended** command enables a Cisco device to run in extended TACACS mode. The UNIX system must be running the extended TACACS daemon, which can be obtained via anonymous ftp to *ftp.cisco.com*. The filename is *xtacacsd.shar.* This daemon allows communication servers and other equipment to talk to the UNIX system and update an audit trail with information on port usage, accounting data, or any other information the device can send.

The command **username <user> password [0 | 7] <password>** allows you to store and maintain a list of users and their passwords on a Cisco device instead of on a TACACS server. The number 0 stores the password in cleartext in the configuration file. The number 7 stores the password in an encrypted format. If you do not have a TACACS server and still want to authenticate users on an individual basis, you can set up users with the following configuration commands:

```
username steve password 7 steve-pass
username allan password 7 allan-pass
```

The two users, Steve and Allan, will be authenticated via passwords that are stored in encrypted format.

Token Card Access

Using TACACS service on routers and communications servers, support for physical card key devices, or token cards, can also be added. The TACACS server code can be modified to provide support for this without requiring changes in the setup and configuration of the routers and communication servers. This modified code is not directly available from Cisco.

The token card system relies on a physical card that must be in your possession in order to provide authentication. By using the appropriate hooks in the TACACS server code, third-party companies can offer these enhanced TACACS servers to customers. One such product is the Enigma Logic SafeWord security software system. Other card-key systems, such as Security Dynamics SmartCard, can be added to TACACS as well.

Simple Network Management Protocol (SNMP) Access

SNMP is another method you can use to access your routers. With SNMP, you can gather statistics or configure the router. Gather statistics with get-request and get-next-request messages, and configure routers with set-request messages. Each of these SNMP messages has a community string that is a cleartext password sent in every packet between a management station and the router (which contains an SNMP agent). The SNMP community string is used to authenticate messages sent between the manager and agent. Only when the manager sends a message with the correct community string will the agent respond.

The SNMP agent on the router allows you to configure different community strings for nonprivileged and privileged access. You configure community strings on the router via the configuration command **snmp-server community** <*string*> **[RO | RW]** [*access-list*]. The following sections explore the various ways to use this command.

Unfortunately, SNMP community strings are sent on the network in cleartext ASCII. Thus, anyone who has the ability to capture a packet on the network can discover the community string. This may allow unauthorized users to query or modify routers via SNMP. For this reason, using the **no snmp-server trap-authentication** command may prevent intruders from using trap messages (sent between SNMP managers and agents) to discover community strings.

The Internet community, recognizing this problem, greatly enhanced the security of SNMP version 2 (SNMPv2) as described in RFC 1446. SNMPv2 uses an algorithm called *MD5* to authenticate communications between an SNMP server and agent. MD5 verifies the integrity of the communications, authenticates the origin, and checks for timeliness. Further, SNMPv2 can use the data encryption standard (DES) for encrypting information.

Nonprivileged Mode

Use the **RO** keyword of the **snmp-server community** command to provide nonprivileged access to your routers via SNMP. The following configuration command sets the agent in the router to allow only SNMP get-request and get-next-request messages that are sent with the community string "public":

```
snmp-server community public RO 1
```

You can also specify a list of IP addresses that are allowed to send messages to the router using the *access-list* option with the **snmp-server community** command. In the following configuration example, only hosts 1.1.1.1 and 2.2.2.2 are allowed nonprivileged mode SNMP access to the router:

```
access-list 1 permit 1.1.1.1
access-list 1 permit 2.2.2.2
snmp-server community public RO 1
```

Privileged Mode

Use the **RW** keyword of the **snmp-server community** command to provide privileged access to your routers via SNMP. The following configuration command sets the agent in the router to allow only SNMP set-request messages sent with the community string "private":

```
snmp-server community private RW 1
```

You can also specify a list of IP addresses that are allowed to send messages to the router by using the *access-list* option of the **snmp-server community** command. In the following configuration example, only hosts 5.5.5.5 and 6.6.6.6 are allowed privileged mode SNMP access to the router:

```
access-list 1 permit 5.5.5.5
access-list 1 permit 6.6.6.6
snmp-server community private RW 1
```

Controlling Access to Network Servers That Contain Configuration Files

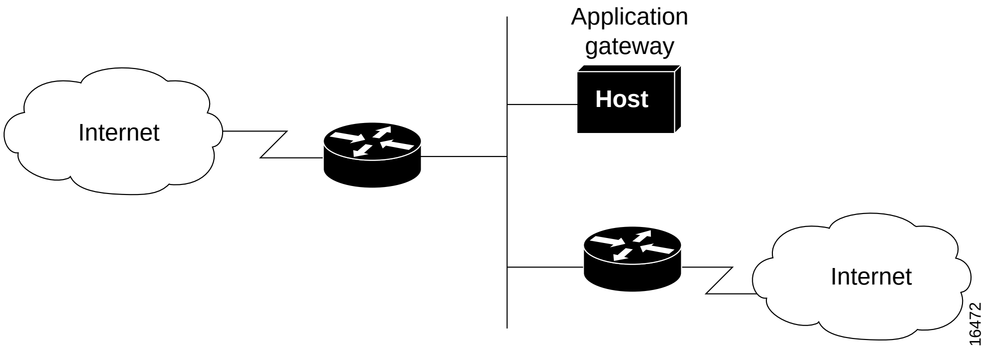
If a router regularly downloads configuration files from a Trivial File Transfer Protocol (TFTP) or Maintenance Operations Protocol (MOP) server, anyone who can access the server can modify the router configuration files stored on the server.

Communication servers can be configured to accept incoming local area transport (LAT) connections. Protocol translators and their translating router brethren can accept X.29 connections. These different types of access should be considered when creating a firewall architecture.

Setting Up Your Firewall Architecture

A firewall architecture is a structure that exists between you and the outside world to protect you from intruders. In most circumstances, intruders are represented by the global Internet and the thousands of remote networks it interconnects. Typically, a network firewall consists of several different machines as shown in Figure 3-1.

Figure 3-1 Typical firewall architecture.



In this architecture, the router that is connected to the Internet (exterior router) forces all incoming traffic to go to the application gateway. The router that is connected to the internal network (interior router) accepts packets only from the application gateway.

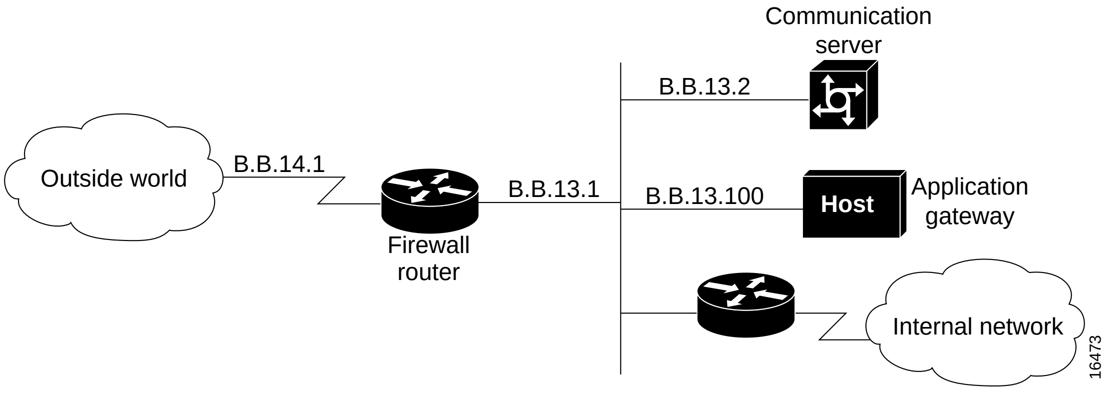
The application gateway institutes per-application and per-user policies. In effect, the gateway controls the delivery of network-based services both into and from the internal network. For example, only certain users might be allowed to communicate with the Internet, or only certain applications are permitted to establish connections between an interior and exterior host.

The route and packet filters should be set up to reflect the same policies. If the only application that is permitted is mail, only mail packets should be allowed through the router. This protects the application gateway and avoids overwhelming it with packets that it would otherwise discard.

Controlling Traffic Flow

This section uses the scenario illustrated in Figure 3-2 to describe the use of access lists to restrict traffic to and from a firewall router and a firewall communication server.

Figure 3-2 Controlling traffic flow via the firewall router.



In this case study, the firewall router allows incoming new connections to one or more communication servers or hosts. Having a designated router act as a firewall is desirable because it clearly identifies the router's purpose as the external gateway and avoids encumbering other routers with this task. In the event that the internal network needs to isolate itself, the firewall router provides the point of isolation so that the rest of the internal network structure is not affected.

Connections to the hosts are restricted to incoming file transfer protocol (FTP) requests and email services as described in the "Configuring the Firewall Router" section later in this chapter. The incoming Telnet, or modem, connections to the communication server are screened by the communication server running TACACS username authentication, as described in the "Configuring the Firewall Communication Server" section later in this chapter.

Note Connections from one communication server modem line to another outgoing modem line (or to the outside world) should be disallowed to prevent unauthorized users from using your resources to launch an attack on the outside world. Because intruders have already passed the communication server TACACS authentication at this point, they are likely to have someone's password. It is an excellent idea to keep TACACS passwords and host passwords distinct from one another.

Configuring the Firewall Router

In the firewall router configuration that follows, subnet 13 of the Class B network is the firewall subnet, whereas subnet 14 provides the connection to the worldwide Internet via a service provider:

```
interface ethernet 0
ip address B.B.13.1 255.255.255.0
interface serial 0
ip address B.B.14.1 255.255.255.0
router igrp
network B.B.0.0
```

This simple configuration provides *no security* and allows all traffic from the outside world onto all parts of the network. To provide security on the firewall router, use access lists and access groups as described in the next section.

Defining Access Lists

Access lists define the actual traffic that will be permitted or denied, whereas an access group applies an access list definition to an interface. Access lists can be used to deny connections that are known to be a security risk and then permit all other connections, or to permit those connections that are considered acceptable and deny all the rest. For firewall implementation, the latter is the more secure method.

In this case study, incoming email and news are permitted for a few hosts, but FTP, Telnet, and rlogin services are permitted only to hosts on the firewall subnet. IP *extended* access lists (range 100 to 199) and transmission control protocol (TCP) or user datagram protocol (UDP) port numbers are used to filter traffic. When a connection is to be established for email, Telnet, FTP, and so forth, the connection will attempt to open a service on a specified port number. You can, therefore, filter out selected types of connections by denying packets that are attempting to use that service. For a list of well-known services and ports, see the "Filtering TCP and UDP Services" section later in this chapter.

An access list is invoked after a routing decision has been made but before the packet is sent out on an interface. The best place to define an access list is on a preferred host using your favorite text editor. You can create a file that contains the **access-list** commands, place the file (marked *readable*) in the default TFTP directory, and then network load the file onto the router.

The network server storing the file must be running a TFTP daemon and have TCP network access to the firewall router. Before network loading the access control definition, any previous definition of this access list is removed by using the following command:

```
no access-list 101
```

The **access-list** command can now be used to permit any packets returning to machines from already established connections. With the **established** keyword, a match occurs if the TCP datagram has the acknowledgment (ACK) or reset (RST) bits set.

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 established
```

If any firewall routers share a common network with an outside provider, you may want to allow access from those hosts to your network. In this case study, the outside provider has a serial port that uses the firewall router Class B address (B.B.14.2) as a source address as follows:

```
access-list 101 permit ip B.B.14.2 0.0.0.0 0.0.0.0 255.255.255.255
```

The following example illustrates how to deny traffic from a user attempting to spoof any of your internal addresses from the outside world (*without* using 9.21 input access lists):

```
access-list 101 deny ip B.B.0.0 0.0.255.255 0.0.0.0 255.255.255.255
```

The following commands allow domain name system (DNS) and network time protocol (NTP) requests and replies:

```
access-list 101 permit udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 53
access-list 101 permit udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 123
```

The following command denies the network file server (NFS) user datagram protocol (UDP) port:

```
access-list 101 deny udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 2049
```

The following commands deny OpenWindows on ports 2001 and 2002 and deny X11 on ports 6001 and 6002. This protects the first two screens on any host. If you have any machine that uses more than the first two screens, be sure to block the appropriate ports.

```
access-list 101 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 6001
access-list 101 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 6002

access-list 101 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 2001
access-list 101 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 2002
```

The following command permits Telnet access to the communication server (B.B.13.2):

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.2 0.0.0.0 eq 23
```

The following commands permit FTP access to the host on subnet 13:

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 eq 21
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 eq 20
```

For the following examples, network B.B.1.0 is on the internal network. Figure 3-2The following commands permit TCP and UDP connections for port numbers greater than 1023 to a very limited set of hosts. Make sure no communication servers or protocol translators are in this list.

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 gt 1023
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.1.100 0.0.0.0 gt 1023
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.1.101 0.0.0.0 gt 1023
access-list 101 permit udp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 gt 1023
access-list 101 permit udp 0.0.0.0 255.255.255.255 B.B.1.100 0.0.0.0 gt 1023
access-list 101 permit udp 0.0.0.0 255.255.255.255 B.B.1.101 0.0.0.0 gt 1023
```

Note Standard FTP uses ports above 1023 for its data connections; therefore, for standard FTP operation, ports above 1023 must all be open. For more details, see the “File Transfer Protocol (FTP) Port” section that follows.

The following commands permit DNS access to the DNS server(s) listed by the Network Information Center (NIC):

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 eq 53
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.1.100 0.0.0.0 eq 53
```

The following commands permit incoming simple mail transfer protocol (SMTP) email to only a few machines:

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 eq 25
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.1.100 0.0.0.0 eq 25
```

The following commands allow internal network news transfer protocol (NNTP) servers to receive NNTP connections from a list of authorized peers:

```
access-list 101 permit tcp 16.1.0.18 0.0.0.1 B.B.1.100 0.0.0.0 eq 119
access-list 101 permit tcp 128.102.18.32 0.0.0.0 B.B.1.100 0.0.0.0 eq 119
```

The following command permits Internet control message protocol (ICMP) for error message feedback:

```
access-list 101 permit icmp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

Every access list has an implicit "deny everything else" statement at the end of the list to ensure that attributes that are not expressly permitted are in fact denied.

File Transfer Protocol (FTP) Port

Many sites today choose to block incoming TCP sessions originated from the outside world while allowing outgoing connections. The trouble with this is that blocking incoming connections kills traditional FTP client programs because these programs use the "PORT" command to tell the server where to connect to send the file. The client opens a "control" connection to the server, but the server then opens a "data" connection to an effectively arbitrarily chosen (> 1023) port number on the client.

Fortunately, there is an alternative to this behavior that allows the client to open the "data" socket and allows you to have the firewall and FTP too. The client sends a PASV command to the server, receives back a port number for the data socket, opens the data socket to the indicated port, and finally sends the transfer.

In order to implement this method, the standard FTP client program must be replaced with a modified one that supports the PASV command. Most recent implementations of the FTP server already support the PASV command. The only trouble with this idea is that it breaks down when the server site has also blocked arbitrary incoming connections.

Source files for a modified FTP program that works through a firewall are now available via anonymous FTP at *ftp.cisco.com*. The file is */pub/passive-ftp.tar.Z*. This is a version of BSD 4.3 FTP with the PASV patches. It works through a firewall router that allows only incoming established connections.

Caution Care should be taken in providing anonymous FTP service on the host system. Anonymous FTP service allows anyone to access the hosts, without requiring an account on the host system. Many implementations of the FTP server have severe bugs in this area. Also, take care in the implementation and setup of the anonymous FTP service to prevent any obvious access violations. For most sites, anonymous FTP service is disabled.

Applying Access Lists to Interfaces

After this access list has been loaded onto the router and stored into nonvolatile random-access memory (NVRAM), assign it to the appropriate interface. In this case study, traffic coming from the outside world via serial 0 is filtered before it is placed on subnet 13 (ethernet 0). Therefore, the **access-group** command, which assigns an access list to filter incoming connections, must be assigned to Ethernet 0 as follows:

```
interface ethernet 0
ip access-group 101
```

To control outgoing access to the Internet from the network, define an access list and apply it to the outgoing packets on serial 0 of the firewall router. To do this, returning packets from hosts using Telnet or FTP must be allowed to access the firewall subnetwork B.B.13.0.

Filtering TCP and UDP Services

Some well-known TCP and UDP port numbers include the services listed in Table 3-3.

Table 3-3 Well-Known TCP and UDP Services and Ports

Service	Port Type	Port Number
File Transfer Protocol (FTP)—Data	TCP	20
FTP—Commands	TCP	21
Telnet	TCP	23
Simple Mail Transfer Protocol (SMTP)—Email	TCP	25
Terminal Access Controller Access Control System (TACACS)	UDP	49
Domain Name Server (DNS)	TCP and UDP	53
Trivial File Transfer Protocol (TFTP)	UDP	69
finger	TCP	79
SUN Remote Procedure Call (RPC)	UDP	111
Network News Transfer Protocol (NNTP)	TCP	119
Network Time Protocol (NTP)	TCP and UDP	123
NeWS	TCP	144
Simple Management Network Protocol (SNMP)	UDP	161
SNMP (traps)	UDP	162
Border Gateway Protocol (BGP)	TCP	179
rlogin	TCP	513
rexec	TCP	514
talk	TCP and UDP	517
ntalk	TCP and UDP	518
Open Windows	TCP and UDP	2000
Network File System (NFS)	UDP	2049
X11	TCP and UDP	6000

CERT Advisory

The Computer Emergency Response Team (CERT) recommends filtering the services listed in Table 3-4.

Table 3-4 CERT Advisory on TCP and UDP Services and Ports

Service	Port Type	Port Number
DNS zone transfers	TCP	53
TFTP daemon (tftpd)	UDP	69
link—commonly used by intruders	TCP	87
SUN RPC	TCP and UDP	111[1]
NFS	UDP	2049
BSD UNIX **r** commands (**rsh**, **rlogin**, and so forth)	TCP	512 through 514
line printer daemon (lpd)	TCP	515
UNIX-to-UNIX copy program daemon (uucpd)	TCP	540
Open Windows	TCP and UDP	2000
X Windows	TCP and UDP	6000+

[1] Port 111 is only a directory service. If you can guess the ports on which the actual data services are provided, you can access them. Most RPC services do not have fixed port numbers. You should find the ports on which these services can be found and block them. Unfortunately, because ports can be bound anywhere, Cisco recommends blocking all UDP ports except DNS where practical.

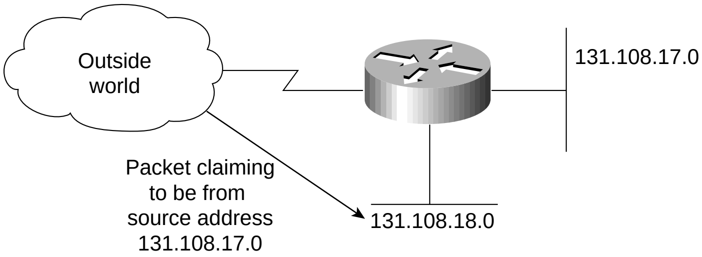
Note Cisco recommends that you filter the finger TCP service at port 79 to prevent outsiders from learning about internal user directories and the names of hosts from which users log in.

Input Access Lists

In Software Release 9.21, Cisco introduces the ability to assign input access lists to an interface. This allows a network administrator to filter packets before they enter the router, instead of as they leave the router. In most cases, input access lists and output access lists accomplish the same functionality; however, input access lists are more intuitive to some people and can be used to prevent some types of IP address “spoofing” where output access lists will not provide sufficient security.

Figure 3-3 illustrates a host that is “spoofing,” or illegally claiming to be an address that it is not. Someone in the outside world is claiming to originate traffic from network 131.108.17.0. Although the address is spoofed, the router interface to the outside world assumes that the packet is coming from 131.108.17.0. If the input access list on the router allows traffic coming from 131.108.17.0, it will accept the illegal packet. To avoid this spoofing situation, an input access list should be applied to the router interface to the outside world. This access list would not allow any packets with addresses that are from the internal networks of which the router is aware (17.0 and 18.0).

Figure 3-3 A host that is spoofing.



If you have several internal networks connected to the firewall router and the router is using output filters, traffic between internal networks will see a reduction in performance created by the access list filters. If input filters are used only on the interface going from the router to the outside world, internal networks will not see any reduction in performance.

Note If an address uses source routing, it can send and receive traffic through the firewall router. For this reason, you should always disable source routing on the firewall router with the **no ip source-route** command.

Configuring the Firewall Communication Server

In this case study, the firewall communication server has a single inbound modem on line 2:

```
interface Ethernet0
ip address B.B.13.2 255.255.255.0
!
access-list 10 deny B.B.14.0 0.0.0.255
access-list 10 permit B.B.0.0 0.0.255.255
!
access-list 11 deny B.B.13.2 0.0.0.0
access-list 11 permit B.B.0.0 0.0.255.255
!
line 2
login tacacs
location FireWallCS#2
!
access-class 10 in
access-class 11 out
!
modem answer-timeout 60
modem InOut
telnet transparent
terminal-type dialup
flowcontrol hardware
stopbits 1
rxspeed 38400
txspeed 38400
!
tacacs-server host B.B.1.100
tacacs-server host B.B.1.101
tacacs-server extended
!
line vty 0 15
login tacacs
```

Defining Access Lists

In this example, the network number is used to permit or deny access; therefore, standard IP access list numbers (range 1 through 99) are used. For incoming connections to modem lines, only packets from hosts on the internal Class B network and packets from those hosts on the firewall subnetwork are permitted:

```
access-list 10 deny B.B.14.0 0.0.0.255
access-list 10 permit B.B.0.0 0.0.255.255
```

Outgoing connections are allowed only to internal network hosts and to the communication server. This prevents a modem line in the outside world from calling out on a second modem line:

```
access-list 11 deny B.B.13.2 0.0.0.0
access-list 11 permit B.B.0.0 0.0.255.255
```

Applying Access Lists to Lines

Apply an access list to an asynchronous line with the **access-class** command. In this case study, the restrictions from access list 10 are applied to incoming connections on line 2. The restrictions from access list 11 are applied to outgoing connections on line 2.

```
access-class 10 in
access-class 11 out
```

Using Banners to Set Up Unauthorized Use Notifications

It is also wise to use the **banner exec** global configuration command to provide messages and unauthorized use notifications, which will be displayed on all new connections. For example, on the communication server, you can enter the following message:

```
banner exec ^C
If you have problems with the dial-in lines, please send mail to helpdesk@Corporation
X.com. If you get the message “% Your account is expiring”, please send mail with name
and voicemail box to helpdesk@CorporationX.com, and someone will contact you to renew
your account. Unauthorized use of these resources is prohibited.
```

Securing Nonstandard Services

There are a number of nonstandard services available from the Internet that provide value-added services when connecting to the outside world. In the case of a connection to the Internet, these services can be very elaborate and complex. Examples of these services are World Wide Web (WWW), Wide Area Information Service (WAIS), gopher, and Mosaic. Most of these systems are concerned with providing a wealth of information to the user in some organized fashion and allowing structured browsing and searching.

Most of these systems have their own defined protocol. Some, such as Mosaic, use several different protocols to obtain the information in question. Use caution when designing access lists applicable to each of these services. In many cases, the access lists will become interrelated as these services become interrelated.

Summary

Although this case study illustrates how to use Cisco network layer features to increase network security on IP networks, in order to have comprehensive security, you must address all systems and layers.

Recommended Reading

This section contains a list of publications that provide internetwork security information.

Books and Periodicals

Cheswick, B. and Bellovin, S. *Firewalls and Internet Security.* Addison-Wesley.

Comer, D.E and Stevens, D.L., *Internetworking with TCP/IP.* Volumes I-III. Englewood Cliffs, New Jersey: Prentice Hall; 1991-1993.

Curry, D. *UNIX System Security—A Guide for Users and System Administrators.*

Garfinkel and Spafford. *Practical UNIX Security.* O'Reilly & Associates.

Quarterman, J. and Carl-Mitchell, S. *The Internet Connection*, Reading, Massachusetts: Addison-Wesley Publishing Company; 1994.

Ranum, M. J. *Thinking about Firewalls*, Trusted Information Systems, Inc.

Stoll, C. *The Cuckoo's Egg*. Doubleday.

Treese, G. W. and Wolman, A. *X through the Firewall and Other Application Relays.*

Requests For Comments (RFCs)

RFC 1118. "The Hitchhiker's Guide to the Internet." September 1989.

RFC 1175. "A Bibliography of Internetworking Information." August 1990.

RFC1244. "Site Security Handbook." July 1991.

RFC 1340. "Assigned Numbers." July 1992.

RFC 1446. "Security Protocols for SNMPv2." April 1993.

RFC 1463. "FYI on Introducing the Internet—A Short Bibliography of Introductory Internetworking Readings for the Network Novice." May 1993.

RFC 1492. "An Access Control Protocol, Sometimes Called TACACS." July 1993.

Internet Directories

Documents at *gopher.nist.gov.*

The "Computer Underground Digest" in the */pub/cud* directory at *ftp.eff.org*.

Documents in the */dist/internet_security* directory at *research.att.com.*

Integrating Enhanced IGRP into Existing Networks

The Enhanced Interior Gateway Routing Protocol (IGRP) combines the ease of use of traditional routing protocols with the fast rerouting capabilities of link-state protocols, providing advanced capabilities for fast convergence and partial updates. When a network topology change occurs, the Diffusing Algorithm (DUAL) used with Enhanced IGRP provides convergence in less than five seconds in most cases. This is equivalent to the convergence achieved by link-state protocols such as Open Shortest Path First (OSPF), Novell Link Services Protocol (NLSP), and Intermediate System-to-Intermediate System (IS-IS). In addition, Enhanced IGRP sends routing update information only when changes occur, and only the changed information is sent to affected routers.

Enhanced IGRP supports three network level protocols: IP, AppleTalk, and Novell Internetwork Packet Exchange (IPX). Each of these has protocol-specific, value-added functionality. IP Enhanced IGRP supports variable-length subnet masks (VLSMs). IPX Novell Enhanced IGRP supports incremental Service Advertisement Protocol (SAP) updates, removes the Routing Information Protocol (RIP) limitation of 15 hop counts, and provides optimal path use. A router running AppleTalk Enhanced IGRP supports partial, bounded routing updates and provides load sharing and optimal path use.

The case study provided here discusses the benefits and considerations involved in integrating Enhanced IGRP into the following types of internetworks:

- *IP*—The existing IP network is running IGRP
- *Novell IPX*—The existing IPX network is running RIP and SAP
- *AppleTalk*—The existing AppleTalk network is running the Routing Table Maintenance Protocol (RTMP)

When integrating Enhanced IGRP into existing networks, plan a phased implementation. Add Enhanced IGRP at the periphery of the network by configuring Enhanced IGRP on a boundary router on the backbone off the core network. Then integrate Enhanced IGRP into the core network.

Note For a discussion of Enhanced IGRP network design considerations and details on DUAL convergence, see the *Internetwork Design Guide*.

Caution If you are using *candidate default route* in IP Enhanced IGRP and have installed multiple releases of Cisco router software within your internetwork that include any versions prior to September 1994, contact your Cisco technical support representative for version compatibility and software upgrade information. Refer to your software release notes for details. If you plan to implement Enhanced IGRP over a Frame Relay network, you should ensure that your network is hierarchical in design and adheres to sound design principles.

IP Network

This case study illustrates the integration of Enhanced IGRP into an IGRP internetwork in two phases: configuring an IGRP network and adding Enhanced IGRP to the network. The key considerations for integrating Enhanced IGRP into an IP network running IGRP are as follows:

- Route selection
- Metric handling
- Redistribution from IGRP to Enhanced IGRP and vice versa
- Route summarization

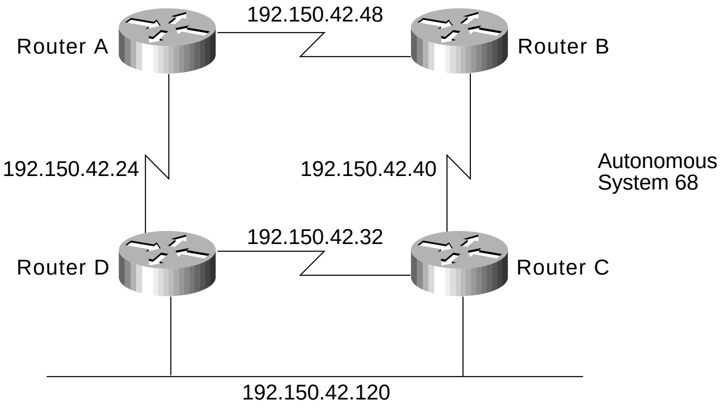
Configuring an IGRP Network

IGRP is a dynamic distance vector routing protocol designed by Cisco Systems in the mid-1980s for routing in an autonomous system (AS) containing large, arbitrarily complex networks with diverse media.

An autonomous system is a collection of interconnected routers under common management control, or with similar routing policies and requirements. Typically, an autonomous system consists of routers connecting multiple IP network numbers. Routes originating from one autonomous system that need to be advertised into other autonomous systems must be redistributed.

In Figure 4-1, Routers A, B, C, and D are configured to run IGRP in autonomous system 68.

Figure 4-1 Configuring an IGRP network.



The configuration commands to enable IGRP routing for Routers A, B, C, and D are as follows:

```
router igrp 68
network 192.150.42.0
```

Adding Enhanced IGRP to IGRP Networks

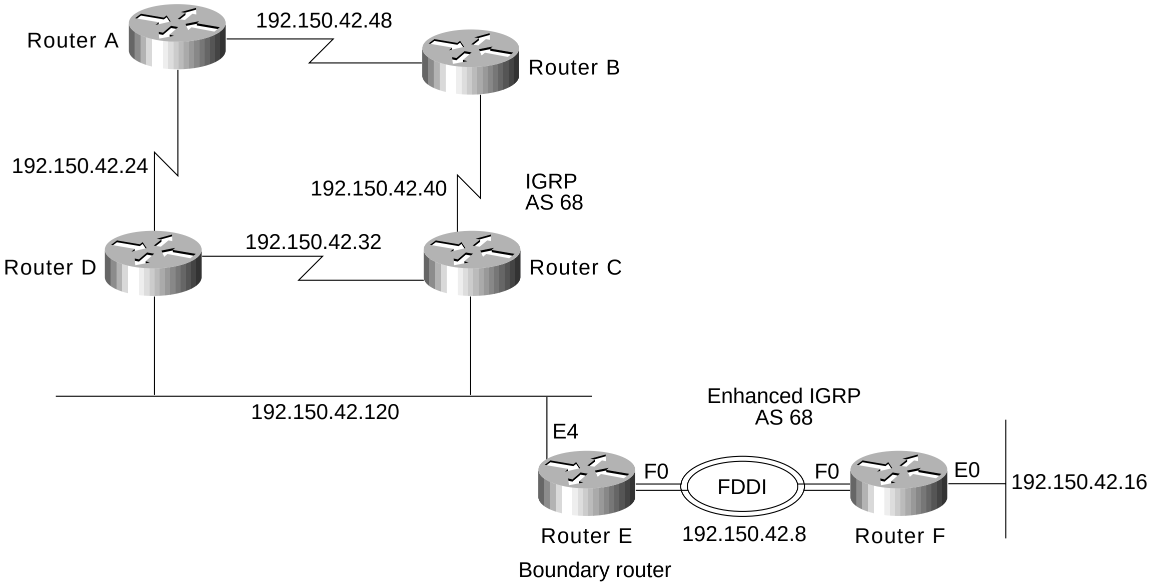
This section provides two examples of adding Enhanced IGRP to IGRP networks:

- Adding Enhanced IGRP to a Single IGRP Network
- Adding Enhanced IGRP to Multiple IGRP Networks

Adding Enhanced IGRP to a Single IGRP Network

In Figure 4-2, Router E acts as the boundary router, running both IGRP and Enhanced IGRP, and redistributing information between IGRP autonomous system 68 into the Enhanced IGRP autonomous system 68.

Figure 4-2 Adding Enhanced IGRP to a single IGRP network.



Router E, the boundary router, is configured to run both IGRP and Enhanced IGRP as follows:

```
router igrp 68
network 192.150.42.0
router eigrp 68
network 192.150.42.0
```

Note Redistribution is automatic because the autonomous system number for IGRP and Enhanced IGRP are the same.

Router F runs Enhanced IGRP only:

```
router eigrp 68
network 192.150.42.0
```

A **show ip route** command on Router E shows networks that are directly connected (C), routes learned from IGRP (I), and routes learned from Enhanced IGRP (D):

```
192.150.42.0 is subnetted (mask is 255.255.255.248), 7 subnets
C       192.150.42.120 is directly connected, Ethernet4
I       192.150.42.48 [100/2860] via 192.150.42.123, 0:00:08, Ethernet4
I       192.150.42.40 [100/2850] via 192.150.42.121, 0:00:08, Ethernet4
I       192.150.42.32 [100/2850] via 192.150.42.121, 0:00:08, Ethernet4
I       192.150.42.24 [100/2760] via 192.150.42.123, 0:00:08, Ethernet4
D       192.150.42.16 [90/30720] via 192.150.42.10, 0:00:38, Fddi0
C       192.150.42.8 is directly connected, Fddi0
```

A **show ip route** command on Router F shows that all routes are learned via enhanced IGRP (D) or are directly connected (C):

```
192.150.42.0 is subnetted (mask is 255.255.255.248), 7 subnets
D       192.150.42.120 [90/729600] via 192.150.42.9, 0:01:16, Fddi0
D EX    192.150.42.48 [170/757760] via 192.150.42.9, 0:01:16, Fddi0
D EX    192.150.42.40 [170/755200] via 192.150.42.9, 0:01:16, Fddi0
D EX    192.150.42.32 [170/755200] via 192.150.42.9, 0:01:16, Fddi0
D EX    192.150.42.24 [170/732160] via 192.150.42.9, 0:01:16, Fddi0
C       192.150.42.16 is directly connected, Ethernet0
C       192.150.42.8 is directly connected, Fddi0
```

Subnetwork 120 is seen as an internal route. All other routes are external (EX) because they were learned via IGRP in Router E and redistributed into Enhanced IGRP.

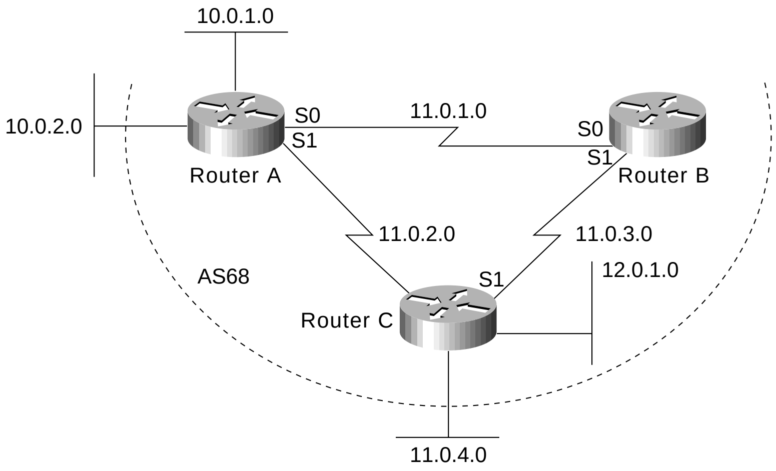
A **show ip eigrp topology** command on Router F shows that the state of each of the networks is passive (P) and that each network has one successor and lists the feasible distance (FD) of each successor via a neighbor to the destination. The computed/advertised metric is listed. Then the interface through which the neighbor network is available is provided.

```
IP-EIGRP Topology Table for process 68
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status
P 192.150.42.120 255.255.255.248, 1 successors, FD is 2172416
         via 192.150.42.9 (2172416/2169856), Fddi0
P 192.150.42.8 255.255.255.248, 1 successors, FD is 28160
         via Connected, Fddi0
P 192.150.42.48 255.255.255.248, 1 successors, FD is 2560515840
         via 192.150.42.9 (2560515840/2560513280), Fddi0
P 192.150.42.16 255.255.255.248, 1 successors, FD is 281600
         via Connected, Ethernet0
P 192.150.42.40 255.255.255.248, 1 successors, FD is 2560026880
         via 192.150.42.9 (2560026880/2560001280), Fddi0
P 192.150.42.32 255.255.255.248, 1 successors, FD is 2560026880
         via 192.150.42.9 (2560026880/2560001280), Fddi0
```

Adding Enhanced IGRP to Multiple IGRP Networks

In Figure 4-3, Routers A, B, and C are connected to each other through several different networks. Routers A, B, and C are configured to run IGRP only within IGRP autonomous system (AS) 68. Router A redistributes static routes for subnetworks of network 9.0.0.0 (not shown). Assume that the IGRP AS continues at network 10.0.0.0.

Figure 4-3 Adding Enhanced IGRP to multiple IGRP networks.



The configuration for Router A is as follows:

```
router igrp 68
network 10.0.0.0
network 11.0.0.0
default-metric 1000 100 1 1 1500
redistribute static
ip route 9.1.0.0 255.255.0.0 e0
ip route 9.2.0.0 255.255.0.0 e1
```

The configuration for Router B is as follows:

```
router igrp 68
network 11.0.0.0
```

The configuration for Router C is as follows:

```
router igrp 68
network 11.0.0.0
network 12.0.0.0
```

This example takes you through the steps to add Enhanced IGRP to the internetwork one router at a time:

Step 1 Configure Enhanced IGRP for Router C as follows:

```
router eigrp 68
network 11.0.0.0
network 12.0.0.0
```

Because they are directly connected networks, Router C automatically summarizes networks 11.0.0.0 and 12.0.0.0 in its routing updates. Router C learns about networks 9.0.0.0 and 10.0.0.0 through IGRP. Networks 9.0.0.0 and 10.0.0.0 are already IGRP-summarized by Router A before they reach Router C.

Step 2 Configure Router A to run Enhanced IGRP as follows:

```
router eigrp 68
network 10.0.0.0
network 11.0.0.0
default-metric 1000 100 1 1 1500
redistribute static
```

Router A now automatically summarizes networks 10.0.0.0 and 11.0.0.0 in its Enhanced IGRP routing updates. It also continues to summarize these networks in its IGRP routing updates. However, automatic summarization of network 9.0.0.0 through Enhanced IGRP is not performed.

Router C now learns Enhanced IGRP routes for specific subnetworks of network 9.0.0.0 from Router A. At the same time, Router C continues to receive a summary route for network 9.0.0.0 though IGRP from Router A. The summary route for network 10.0.0.0, which Router C had previously learned through IGRP from Router A, is replaced with an Enhanced IGRP route in Router C's routing table.

Step 3 Configure Router A to ensure that Router C does not unnecessarily learn about specific subnetworks of network 9.0.0.0. The following commands enable summarization of network 9.0.0.0 at Router A:

```
interface serial 1
ip summary-address eigrp 68 9.0.0.0 255.0.0.0
```

With this configuration on Router A, Router C's IGRP summary route for network 9.0.0.0 is replaced with an Enhanced IGRP summary route, and the more specific subnetworks of network 9.0.0.0 are no longer known by Router C.

Step 4 Enable Enhanced IGRP on Router B as follows:

```
router eigrp 68
network 11.0.0.0
```

Step 5 Ensure that Router B does not unnecessarily learn about specific subnetworks of network 9.0.0.0. Therefore, configure summarization of network 9.0.0.0 at Router A as follows:

```
interface serial 0
ip summary-address eigrp 68 9.0.0.0 255.0.0.0
```

With this configuration on Router A, Router B learns a summary route for network 12.0.0.0 through Enhanced IGRP from Router C. Router B learns summary routes for networks 9.0.0.0 and 10.0.0.0 through Enhanced IGRP from Router A.

Step 6 Now that both of the next hop routers (Routers B and C) are running Enhanced IGRP, it is no longer necessary for these routers to run IGRP. Disable IGRP on Routers B and C with the following command:

```
no router igrp 68
```

Router A continues to run both IGRP and Enhanced IGRP and redistribute static routes.

If there were more routers on the network, you could continue deployment of Enhanced IGRP throughout network 10.0.0.0 one router at a time.

Route Selection

Enhanced IGRP uses three kinds of routes: internal, external, and summary. Internal routes are routes that are learned from Enhanced IGRP. External routes are routes that are learned from another protocol and then redistributed into Enhanced IGRP. Summary routes are routes that Enhanced IGRP may dynamically create due to auto summarization, or due to an explicit summary route configuration. Route selection is based on administrative distance. The default administrative distance for Enhanced IGRP is 90 (internal), 170 (external), or 5 (summary). For IGRP, the default administrative distance is 100 because internal Enhanced IGRP routes take precedence over IGRP routes, and IGRP routes are preferred to external Enhanced IGRP routes.

Metric Handling

The metric calculation and default metric value for IGRP and Enhanced IGRP are the same. By default, the composite metric is the sum of the segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. Although you can adjust the default value with the **metric weights** command, the defaults were carefully selected to provide excellent operation in most networks.

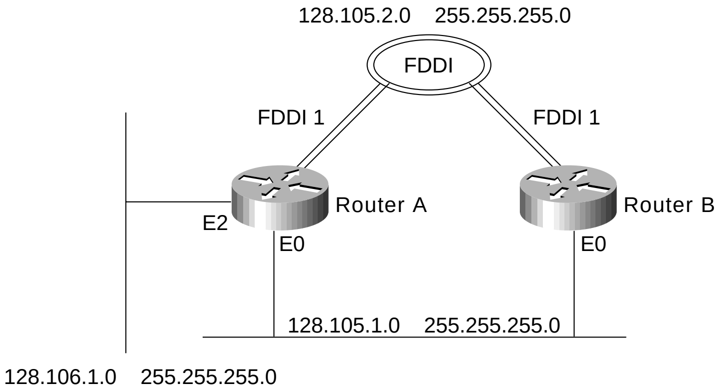
Redistribution

Enhanced IGRP can be added to an IGRP network in two ways: using the same IGRP AS number or using a new AS number. If Enhanced IGRP uses the same AS number as IGRP, redistribution of IGRP into Enhanced IGRP and redistribution of Enhanced IGRP into IGRP occurs. If Enhanced IGRP uses a different AS number, the network administrator needs to configure redistribution manually with the **redistribute** command. For redistributing information from Enhanced IGRP into other dynamic routing protocols besides IGRP and vice versa, the designer must use the **redistribute** and **default-metric** commands. IGRP routes redistributed into Enhanced IGRP are marked as external.

Route Summarization

With IGRP, routing information advertised out an interface is often automatically summarized at major network number boundaries. Specifically, this automatic summarization occurs for those routes whose major network number differs from the major network number of the interface to which the advertisement is being sent. The remaining routes, which are part of the major network number of the interface, are advertised without summarization. For the following example, refer to Figure 4-4.

Figure 4-4 Route summarization.



In this example, Router A is directly connected to two different major networks and configured as follows:

```
interface ethernet 0
ip address 128.105.1.1 255.255.255.0
interface fddi 1
ip address 128.105.2.1 255.255.255.0
interface ethernet 2
ip address 128.106.1.1 255.255.255.0
router igrp 5
network 128.105.0.0
network 128.106.0.0
```

When advertising routing information out Ethernet interface 0, IGRP will summarize network 128.106.0.0 and will not summarize network 128.105.0.0. Therefore, IGRP will advertise routes for 128.106.0.0 with a network mask of 255.255.0.0 and routes for 128.105.2.1 with a network mask of 255.255.255.0.

Because it provides automatic route summarization, Enhanced IGRP will advertise the same routing information in the previous IGRP example. However, in the Enhanced IGRP example that follows, the previous configuration is modified so that it allows redistribution of routing information that is not summarized:

```
ip route 128.107.1.0 255.255.255.0 128.106.1.2
router eigrp 5
redistribute static
network 128.105.0.0
network 128.106.0.0
router igrp 5
redistribute static
```

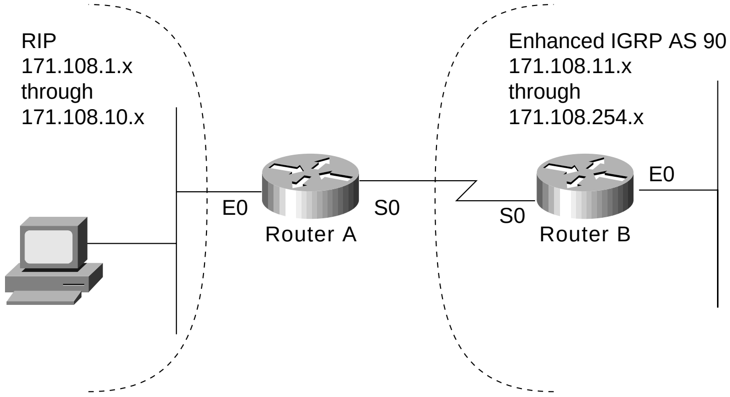
At this point, there is a third subnetted major network in the IP routing table. When advertising out Ethernet interface 0, IGRP will summarize the route for 128.107.1.0 as 128.107.0.0 with a network mask of 255.255.0.0. However, Enhanced IGRP will not summarize network 128.107.0.0. It will advertise 128.107.1.0 with network mask 255.255.255.0. Enhanced IGRP's automatic summarization only applies to networks that are directly connected, not redistributed. For Enhanced IGRP, you can explicitly cause network 128.107.0.0 to be summarized out all three interfaces as shown in the following example:

```
interface ethernet 0
ip summary-address eigrp 5 128.107.0.0 255.255.0.0
interface fddi 1
ip summary-address eigrp 5 128.107.0.0 255.255.0.0
interface ethernet 2
ip summary-address eigrp 5 128.107.0.0 255.255.0.0
```

Redistribution between Enhanced IGRP and RIP

Figure 4-5 shows a router that connects two networks; one network uses RIP and the other network uses Enhanced IGRP. The goal for the router is to advertise RIP routes in the Enhanced IGRP network and to advertise Enhanced IGRP routes in the RIP network, while preventing the occurrence of route feedback. (That is, the router must be configured so that Enhanced IGRP does not send routes learned from RIP back into the RIP network and so that RIP does not send routes learned from Enhanced IGRP back into the Enhanced IGRP network.)

Figure 4-5 Redistributing RIP routes into Enhanced IGRP.



The RIP portion of the configuration for Router A is as follows:

```
router rip
network 171.108.0.0
redistribute eigrp 90
default-metric 2
passive-interface serial 0
```

The **router rip** global configuration command starts a RIP process.

The **network** router configuration command specifies that the RIP process is to send RIP updates out on the interfaces that are directly connected to network number 171.108.0.0. In this case, the RIP process will send updates out on Ethernet interface 0 and not on serial interface 0 because of the **passive-interface** command applied to serial interface 0.

The **redistribute eigrp** router configuration command specifies that routing information derived from Enhanced IGRP be advertised in RIP routing updates.

The **default-metric** router configuration command causes RIP to use the same metric value (in this case, a hop count of 2) for all routes obtained from Enhanced IGRP. A default metric helps solve the problem of redistributing routes that have incompatible metrics. Whenever metrics do not convert, using a default metric provides a reasonable substitute and enables the redistribution to proceed.

The **passive-interface** router configuration command disables the sending of routing updates on serial interface 0. In this case, the **passive-interface** command is used with RIP, which means the router does not send out any updates on a passive interface, but the router still processes updates that it receives on that interface.The result is that the router still learns of networks that are behind a passive interface. (The same is true when the **passive-interface** command is used with IGRP.)

The Enhanced IGRP portion of the configuration for Router A is as follows:

```
router eigrp 90
network 171.108.0.0
redistribute rip
default-metric 1544 100 255 1 1500
distribute-list 1 in
passive interface ethernet 0
access-list 1 permit ip 171.108.1.0 255.255.255.0
access-list 1 permit ip 171.108.2.0 255.255.255.0
access-list 1 permit ip 171.108.3.0 255.255.255.0
access-list 1 permit ip 171.108.4.0 255.255.255.0
access-list 1 permit ip 171.108.5.0 255.255.255.0
access-list 1 permit ip 171.108.6.0 255.255.255.0
access-list 1 permit ip 171.108.7.0 255.255.255.0
access-list 1 permit ip 171.108.8.0 255.255.255.0
access-list 1 permit ip 171.108.9.0 255.255.255.0
access-list 1 permit ip 171.108.10.0 255.255.255.0
access-list 1 deny ip
```

The **router eigrp** global configuration command starts an Enhanced IGRP process and assigns to it autonomous system number 90.

The **network** router configuration command specifies that the Enhanced IGRP process is to send Enhanced IGRP updates to the interfaces that are directly connected to network number 171.108.0.0. In this case, the Enhanced IGRP process will send updates out on serial interface 0 and not on Ethernet interface 0 because of the **passive-interface** command applied to Ethernet interface 0.

The **redistribute eigrp** router configuration command specifies that routing information derived from RIP be advertised in Enhanced IGRP routing updates.

The **default-metric** router configuration command assigns an Enhanced IGRP metric to all RIP-derived routes. The first value (1544) specifies a minimum bandwidth of 1544 kilobits per second. The second value (100) specifies a route delay in tens of microseconds. The third value (255) specifies the connection is guaranteed to be 100 percent reliable. The fourth value (1) specifies the effective bandwidth of the route. The fifth value (1500) specifies in bytes the maximum transmission unit (MTU) of the route.

The **distribute-list in** router configuration command causes the router to use access list 1 to filter networks learned from RIP and allows only those networks that match the list to be redistributed into Enhanced IGRP. This prevents route feedback loops from occurring.

When used with Enhanced IGRP, the **passive-interface** router configuration command has a different effect than it has when used with RIP or IGRP. When the **passive-interface** command is used with Enhanced IGRP, the router does not send out any updates—including hello messages—on the interface. Because hello messages are not sent, the router cannot discover any neighbors on that interface, which means that the router does not learn about networks that are behind a passive interface.

Access list 1 permits subnetworks 1 through 10 and denies all other networks. Although ten statements have been used, this particular access list could be written with four **access-list** commands if the address space had been divided efficiently. This example illustrates the need to think carefully about how to divide an address space. For example, if the RIP AS had been subnets 0 through 7, a single access list statement would have covered all of the subnetworks. The implication is that, when using a protocol that can summarize, summarization can be achieved much more efficiently when the IP address space is divided optimally. For information about dividing an IP address space optimally, see Appendix A, “Subnetting an IP Address Space.”

Novell IPX Network

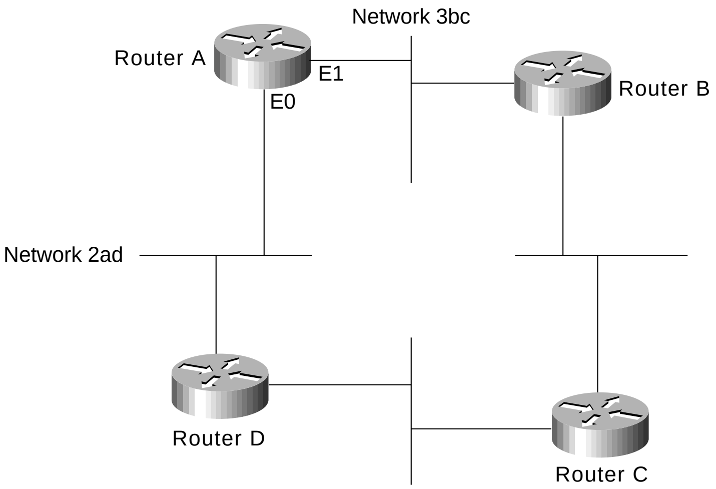
This case study illustrates the integration of Enhanced IGRP into a Novell IPX internetwork in two phases: configuring an IPX network and adding Enhanced IGRP to the IPX network. The key considerations for integrating Enhanced IGRP into an IPX network running RIP and SAP are as follows:

- Route selection
- Redistribution metric handling
- Redistribution from IPX RIP to Enhanced IGRP and vice versa
- Reducing SAP traffic

Configuring a Novell IPX Network

Cisco’s implementation of Novell’s IPX protocol provides all the functions of a Novell router. In this case study, routers are configured to run Novell IPX. (See Figure 4-6.)

Figure 4-6 Configuring a Novell IPX network.



The configuration commands to enable IPX routing for Router A are as follows:

```
ipx routing
interface ethernet 0
ipx network 2ad
interface ethernet 1
ipx network 3bc
```

Note In Software Release 9.21 and later, the command to enable Novell IPX routing is **ipx** rather than **novell**.

Adding Enhanced IGRP to a Novell IPX Network

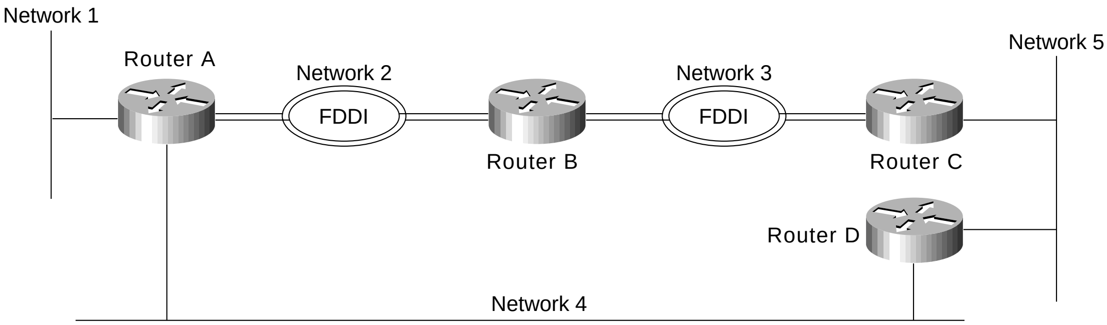
Enhanced IGRP for a Novell IPX network has the same fast rerouting and partial update capabilities as Enhanced IGRP for IP. In addition, Enhanced IGRP has several capabilities that are designed to facilitate the building of large, robust Novell IPX networks.

The first capability is support for incremental SAP updates. Novell IPX RIP routers send out large RIP and SAP updates every 60 seconds. This can consume substantial amounts of bandwidth. Enhanced IGRP for IPX sends out SAP updates only when changes occur and sends only changed information.

The second capability that Enhanced IGRP adds to IPX networks is the ability to build large networks. IPX RIP networks have a diameter limit of 15 hops. Enhanced IGRP networks can have a diameter of 224 hops.

The third capability that Enhanced IGRP for Novell IPX provides is optimal path selection. The RIP metric for route determination is based on ticks with hop count used as a tie-breaker. If more than one route has the same value for the tick metric, the route with the least number of hops is preferred. Instead of ticks and hop count, IPX Enhanced IGRP uses a combination of these metrics: delay, bandwidth, reliability, and load. For an illustration of how IPX Enhanced IGRP provides optimal path selection, see Figure 4-7.

Figure 4-7 Enhanced IGRP Novell IPX optimal path utilization.

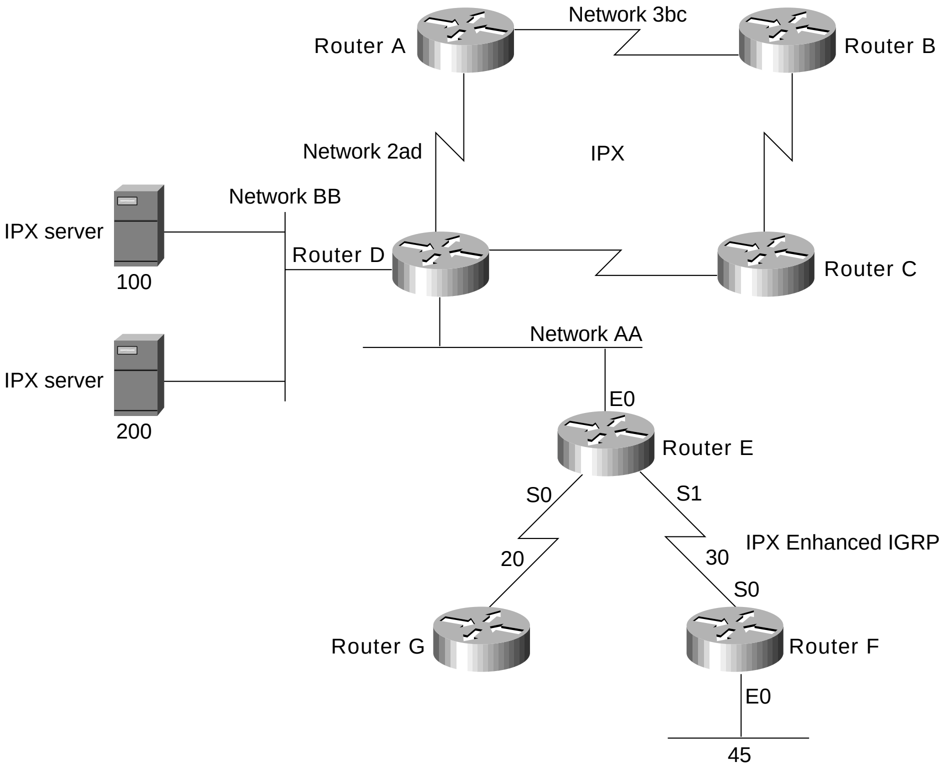


Both Ethernet and FDDI interfaces have a tick value of 1. If configured for Novell RIP, Router A will choose the Ethernet connection via network 4 to reach network 5 because Router D is only one hop away from Router A. However, the fastest path to network 5 is two hops away, via the FDDI rings. With IPX Enhanced IGRP configured, Router A will automatically take the optimal path through Routers B and C to reach network 5.

To add Enhanced IGRP to a Novell RIP and SAP network, configure Enhanced IGRP on the Cisco router interfaces that connect to other Cisco routers also running Enhanced IGRP. Configure RIP and SAP on the interfaces that connect to Novell hosts and or Novell routers that do not support Enhanced IGRP.

In Figure 4-8, Routers E, F, and G are running IPX Enhanced IGRP. Router E redistributes Enhanced IGRP route information via Network AA to Router D.

Figure 4-8 Adding Enhanced IGRP to a Novell IPX network.



The configuration for Router E is as follows:

```
ipx routing
interface ethernet 0
ipx network AA
interface serial 0
ipx network 20
interface serial 1
ipx network 30
ipx router eigrp 10
network 20
network 30
ipx router rip
no network 20
```

With Enhanced IGRP configured, periodic SAP updates are replaced with Enhanced IGRP incremental updates when an Enhanced IGRP peer is found. Unless RIP is explicitly disabled for an IPX network number, as shown for network 20, both RIP and Enhanced IGRP will be active on the interface associated with that network number. Based on the above configuration, and assuming an Enhanced IGRP peer on each Enhanced IGRP configured interface, RIP updates are sent on networks AA and 30, while Enhanced IGRP routing updates are sent on networks 20 and 30. Incremental SAP updates are sent on network 20 and network 30, and periodic SAP updates are sent on network AA.

The configuration for Router F is as follows:

```
ipx routing
interface ethernet 0
ipx network 45
interface serial 0
ipx network 30
ipx router eigrp 10
network 30
network 45
```

Partial output for a **show ipx route** command on Router E indicates that network 45 was discovered using Enhanced IGRP (E), whereas network BB was discovered via a RIP (R) update:

```
R  Net 3bc
R  Net 2ad
C  Net 20 (HDLC), is directly connected, 66 uses, Serial0
C  Net 30 (HDLC), is directly connected, 73 uses, Serial1
E  Net 45 [2195456/0] via 30.0000.0c00.c47e, age 0:01:23, 1 uses, Serial1
C  Net AA (NOVELL-ETHER), is directly connected, 3 uses, Ethernet0
R  Net BB [1/1] via AA.0000.0c03.8b25,  48 sec, 87 uses, Ethernet0
```

Partial output for a **show ipx route** command on Router F indicates that networks 20, AA, and BB were discovered using Enhanced IGRP (E):

```
E  Net 20 [2681856/0] via 30.0000.0c01.f0ed, age 0:02:57, 1 uses, Serial0
C  Net 30 (HDLC), is directly connected, 47 uses, Serial0
C  Net 45 (NOVELL-ETHER), is directly connected, 45 uses, Ethernet0
E  Net AA [267008000/0] via 30.0000.0c01.f0ed, age 0:02:57, 1 uses, Serial0
E  Net BB [268416000/2] via 30.0000.0c01.f0ed, age 0:02:57, 11 uses, Serial0
```

A **show ipx servers** command on Router E shows that server information was learned via periodic (P) SAP updates:

```
Codes: S - Static, I - Incremental, P - Periodic, H - Holddown
5 Total IPX Servers
Table ordering is based on routing and server info
Type Name              Net Address       Port    RouteHopsItf
P     4 Networkers        100.0000.0000.0001:0666    2/022Et1
P     5 Chicago           100.0000.0000.0001:0234    2/022Et1
P     7 Michigan          100.0000.0000.0001:0123    2/022Et1
P     8 NetTest1          200.0000.0000.0001:0345    2/022Et1
P     8 NetTest           200.0000.0000.0001:0456    2/022Et1
```

A **show ipx servers** command on Router F shows that server information was learned via incremental SAP (I) updates allowed with Enhanced IGRP:

```
Codes: S - Static, I - Incremental, P - Periodic, H - Holddown
5 Total IPX Servers
Table ordering is based on routing and server info
Type Name              Net Address       Port    RouteHopsItf
I     4 Networkers        100.0000.0000.0001:0666 268416000/033Se0
I     5 Chicago           100.0000.0000.0001:0234 268416000/033Se0
I     7 Michigan          100.0000.0000.0001:0123 268416000/033Se0
I     8 NetTest1          200.0000.0000.0001:0345 268416000/033Se0
I     8 NetTest           200.0000.0000.0001:0456 268416000/033Se0
```

A **show ipx eigrp topology** command on Router E shows that the state of the networks is passive (P) and that each network provides one successor, and it lists the feasible distance (FD) of each successor via a neighbor to the destination. For example, for network 45, the neighbor is located at address 0000.0c00.c47e and the computed/advertised cost metric for that neighbor to the destination is 2195456/281600:

```
IPX EIGRP Topology Table for process 10
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status
P 20, 1 successors, FD is 1
          via Connected, Serial0
P 30, 1 successors, FD is 1
          via Connected, Serial1
P 45, 1 successors, FD is 2195456
          via 30.0000.0c00.c47e (2195456/281600), Serial1
P AA, 1 successors, FD is 266496000
          via Redistributed (266496000/0),
P BB, 1 successors, FD is 267904000
          via Redistributed (267904000/0),
```

The output for a **show ipx eigrp topology** command on Router F lists the following information:

```
IPX EIGRP Topology Table for process 10
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status
P 20, 1 successors, FD is 2681856
          via 30.0000.0c01.f0ed (2681856/2169856), Serial0
P 30, 1 successors, FD is 1
          via Connected, Serial0
P 45, 1 successors, FD is 1
          via Connected, Ethernet0
P AA, 1 successors, FD is 267008000
          via 30.0000.0c01.f0ed (267008000/266496000), Serial0
P BB, 1 successors, FD is 268416000
          via 30.0000.0c01.f0ed (268416000/267904000), Serial0
```

Route Selection

IPX Enhanced IGRP routes are automatically preferred over RIP routes regardless of metrics unless a RIP route has a hop count less than the external hop count carried in the Enhanced IGRP update, for example, a server advertising its own internal network.

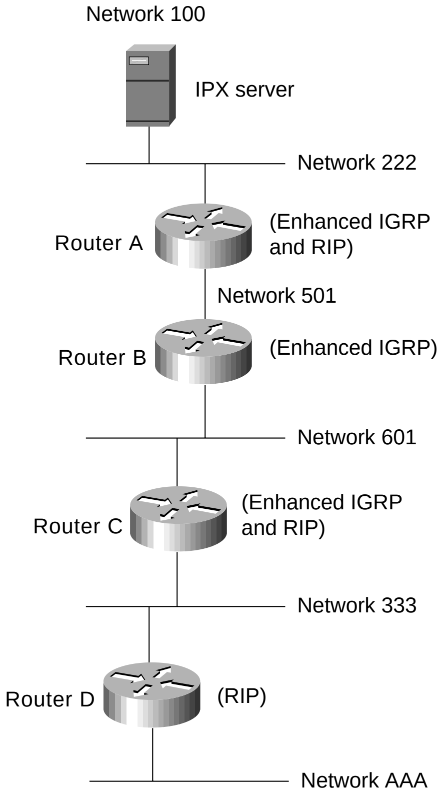
Redistribution and Metric Handling

Redistribution is automatic between RIP and Enhanced IGRP, and vice versa. Automatic redistribution can be turned off using the **no redistribute** command. Redistribution is not automatic between different Enhanced IGRP autonomous systems.

The metric handling for integrating RIP into Enhanced IGRP is bandwidth plus delay, left shifted by 8 bits. The metric handling for Enhanced IGRP to RIP is the external metric plus 1. An IPX Enhanced IGRP router that is redistributing RIP into Enhanced IGRP takes the RIP metric associated with each RIP route, increments it, and stores that metric in the Enhanced IGRP routing table as the external metric.

In Figure 4-9, a Novell IPX server with an internal network number of 100 advertises this network number using RIP on network 222. Router A hears this advertisement and installs it in its routing table as being 1 hop and 1 tick away. Router A then announces this network to Router B on network 501 using Enhanced IGRP.

Figure 4-9 IPX metric handling example.



The configuration for Router A is as follows:

```
ipx routing
!
interface ethernet 0
ipx network 222
!
interface serial 0
ipx network 501
!
ipx router eigrp 9000
network 222
network 501
!
!The following commands turn off IPX RIP on the serial interface:
!
ipx router rip
no network 501
```

The configuration for Router B is as follows:

```
ipx routing
!
interface ethernet 0
ipx network 601
!
interface serial 0
ipx network 501


ipx router eigrp 9000
network 501
network 601
!
!The following command turns off IPX RIP on this router:
!
no ipx router rip
```

The configuration for Router C is as follows:

```
ipx routing
!
interface ethernet 0
ipx network 333
!
interface ethernet 1
ipx network 601
!
ipx router eigrp 9000
network 333
network 601
!
!The following commands turn off IPX RIP on ethernet 1:
!
ipx router rip
no network 601
```

The configuration for Router D is as follows:

```
ipx routing
!
interface ethernet 0
ipx network 333
!
interface ethernet 1
ipx network AAA
```

The output from a **show ipx route** command on Router A is as follows:

```
R  Net 100 [1/1] via 222.0260.8c4c.4f22,  59 sec, 1 uses, Ethernet0
C  Net 222 (ARPA), is directly connected, 1252 uses, Ethernet0
E  Net 333 [46277376/0] via 501.0000.0c05.84bc, age 0:04:07, 1 uses, Serial0
C  Net 501 (HDLC), is directly connected, 3908 uses, Serial0
E  Net 601 [46251776/0] via 501.0000.0c05.84bc, age 5:21:38, 1 uses, Serial0
E  Net AAA [268441600/2] via 501.0000.0c05.84bc, age 0:16:23, 1 uses, Serial0
```

The output from a **show ipx route** command on Router B is as follows:

```
E  Net 100 [268416000/2] via 501.0000.0c05.84b4, age 0:07:30, 2 uses, Serial0
E  Net 222 [267008000/0] via 501.0000.0c05.84b4, age 0:07:30, 1 uses, Serial0
E  Net 333 [307200/0] via 601.0000.0c05.84d3, age 0:07:30, 1 uses, Ethernet0
C  Net 501 (HDLC), is directly connected, 4934 uses, Serial0
C  Net 601 (NOVELL-ETHER), is directly connected, 16304 uses, Ethernet0
E  Net AAA [267929600/2] via 601.0000.0c05.84d3, age 0:14:40, 1 uses, Ethernet0
```

The output from a **show ipx route** command on Router C is as follows:

```
E  Net 100 [268441600/2] via 601.0000.0c05.84bf, age 0:07:33, 1 uses, Ethernet1
E  Net 222 [267033600/0] via 601.0000.0c05.84bf, age 0:07:34, 1 uses, Ethernet1
C  Net 333 (NOVELL-ETHER), is directly connected, 15121 uses, Ethernet0
E  Net 501 [46251776/0] via 601.0000.0c05.84bf, age 0:07:32, 9 uses, Ethernet1
C  Net 601 (NOVELL-ETHER), is directly connected, 1346 uses, Ethernet1
R  Net AAA [1/1] via 333.0000.0c05.8b25,  35 sec, 1 uses, Ethernet0
```

The output from a **show ipx route** command on Router D is as follows:

```
R  Net 100 [8/2] via 333.0000.0c05.84d1,  18 sec, 1 uses, Ethernet0
R  Net 222 [6/1] via 333.0000.0c05.84d1,  18 sec, 1 uses, Ethernet0
R  Net 333 [1/1] via 333.0000.0c05.84d1,  18 sec, 1 uses, Ethernet0
R  Net 501 [3/1] via 333.0000.0c05.84d1,  17 sec, 3 uses, Ethernet0
R  Net 601 [1/1] via 333.0000.0c05.84d1,  18 sec, 1 uses, Ethernet0
C  Net AAA (SNAP), is directly connected, 20 uses, Ethernet1
```

The Enhanced IGRP metric is created using the RIP ticks for the delay vector. The hop count is incremented and stored as the external metric. The external delay is also stored. Router B computes the metric to network 100 given the information received from Router A and installs this in its routing table. In this case, the tick value for network 100 is 8.

The “2” after the slash in the routing entry for network 100 is the external metric. This number does not increase again while the route is in the Enhanced IGRP autonomous system. Router C computes the metric to network 100 through Router B and stores it in its routing table. Finally, Router C redistributes this information back into RIP with a hop count of 2 (the external metric) and a tick value derived from the original tick value of the RIP route (1) plus the Enhanced IGRP delay through the autonomous system converted to ticks.

Reducing SAP Traffic

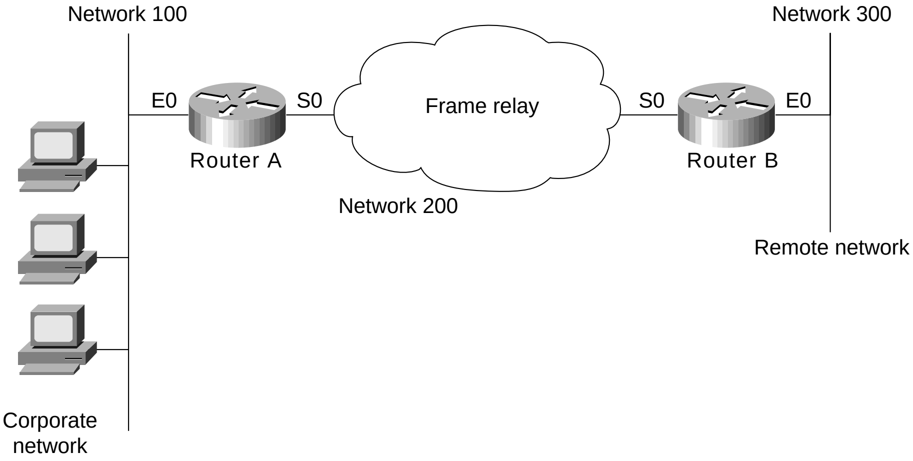
Novell IPX RIP routers send out large RIP and SAP updates every 60 seconds regardless of whether a change has occurred. These updates can consume a substantial amount of bandwidth. You can reduce SAP update traffic by configuring Enhanced IGRP to do incremental SAP updates. When Enhanced IGRP is configured for incremental SAP updates, the updates consist only of information that has changed and the updates are sent out only when a change occurs, thus saving bandwidth.

When you configure Enhanced IGRP for incremental SAP updates, you can do the following:

- Retain RIP, in which case only the reliable transport of Enhanced IGRP is used for sending incremental SAP updates. (This is the preferred configuration over bandwidth-sensitive connections.)
- Turn off RIP, in which case Enhanced IGRP replaces RIP as the routing protocol.

Figure 4-10 shows a bandwidth-sensitive topology in which configuring incremental SAP updates is especially useful. The topology consists of a corporate network that uses a 56-Kbps Frame Relay connection to communicate with a remote branch office. The corporate network has several Novell servers, each of which advertises many services. Depending on the number of servers and the number of advertised services, a large portion of the available bandwidth could easily be consumed by SAP updates.

Figure 4-10 Example of incremental SAP updates.



Router A is configured as follows:

```
ipx routing
!
interface ethernet 0
ipx network 100
!
interface serial 0
encapsulation frame-relay
!
interface serial 0.1 point-to-point
ipx network 200
ipx sap-incremental eigrp 90 rsup-only
frame-relay interface-dlci 101
!
ipx router eigrp 90
network 200
```

The **ipx routing** global configuration command enables IPX routing on the router.

The **ipx network** interface configuration command enables IPX routing on Ethernet interface 0 for network 100.

For serial interface 0, the **encapsulation frame-relay** interface configuration command establishes Frame Relay encapsulation using Cisco’s own encapsulation, which is a 4-byte header, with 2 bytes to identify the DLCI and 2 bytes to identify the packet type.

The **interface serial** global configuration command establishes a point-to-point subinterface (**0.1**). Subinterfaces are logical interfaces associated with a physical interface. Using subinterfaces allows Router A to receive multiple simultaneous connections over a single Frame Relay interface.

The **ipx network** interface configuration command enables IPX routing on subinterface serial interface 0.1 for network 200.

The **ipx sap-incremental** interface configuration command enables the incremental SAP feature. The required **eigrp** keyword enables Enhanced IGRP and its transport mechanism and, in this case, specifies an autonomous system number of 90. Because this command uses the **rsup-only** keyword, the router sends incremental SAP updates on this link.

The **frame-relay interface-dlci** interface configuration command associates data link connection identifier (DLCI) 101 with subinterface serial interface 0.1.

The **ipx router eigrp** global configuration command starts an Enhanced IGRP process and assigns to it autonomous system number 90.

The **network** IPX-router configuration command enables Enhanced IGRP for network 200.

Router B is configured as follows:

```
ipx routing
!
interface ethernet 0
ipx network 300
!
interface serial 0
encapsulation frame-relay
ipx network 200
ipx sap-incremental eigrp 90 rsup-only
!
ipx router eigrp 90
network 200
```

The **ipx routing** global configuration command enables IPX routing on the router.

The **ipx network** interface configuration command enables IPX routing on Ethernet interface 0 for network 300.

On serial interface 0, the **encapsulation frame-relay** interface configuration command establishes Frame Relay encapsulation using Cisco's own encapsulation, which is a 4-byte header, with 2 bytes to identify the DLCI and 2 bytes to identify the packet type.

The **ipx network** interface configuration command enables IPX routing on subinterface serial 0 for network 200.

The **ipx sap-incremental** interface configuration command enables the incremental SAP feature. The required **eigrp** keyword enables Enhanced IGRP and its transport mechanism and, in this case, specifies an autonomous system number of 90. Because this command uses the **rsup-only** keyword, the router sends incremental SAP updates on this link.

The **ipx router eigrp** global configuration command starts an Enhanced IGRP process and assigns to it autonomous system number 90.

The **network** IPX-router configuration command enables Enhanced IGRP for network 200.

Note The absence of the **ipx router rip** command means the IPX RIP is still being used for IPX routing, and the use of the **rsup-only** keyword means that the router is sending incremental SAP updates over the Frame Relay link.

AppleTalk Network

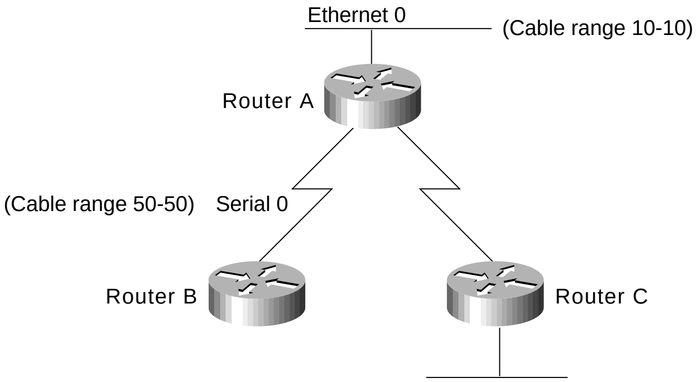
This case study illustrates the integration of Enhanced IGRP into an existing AppleTalk internetwork in two phases: configuring an AppleTalk network and adding Enhanced IGRP to an AppleTalk network. The key considerations for integrating Enhanced IGRP into an AppleTalk network are as follows:

- Route selection
- Metric handling
- Redistribution from AppleTalk to Enhanced IGRP and vice versa

Configuring an AppleTalk Network

Cisco routers support AppleTalk Phase 1 and AppleTalk Phase 2. For AppleTalk Phase 2, Cisco routers support both extended and nonextended networks. In this case study, Routers A, B, and C are running AppleTalk, as illustrated in Figure 4-11.

Figure 4-11 Configuring an AppleTalk network.



The configuration for Router A is as follows:

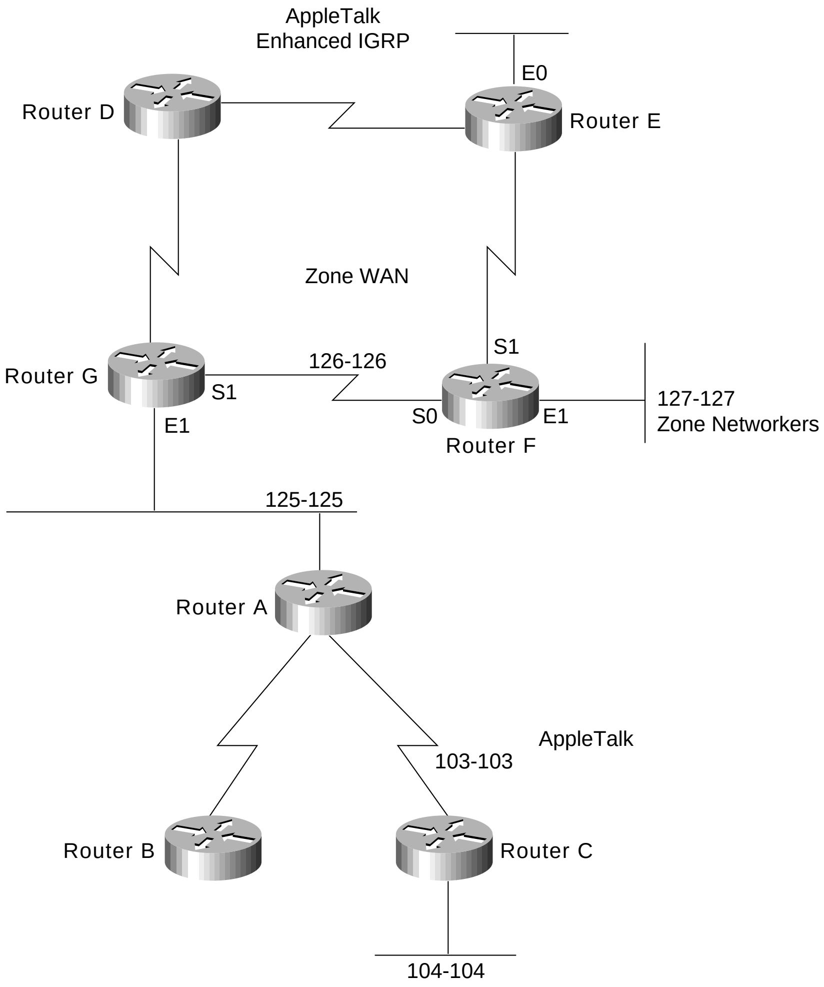
```
appletalk routing
interface ethernet 0
appletalk cable-range 10-10
appletalk zone casestudy
interface serial 0
appletalk cable-range 50-50
appletalk zone casestudy
```

Adding Enhanced IGRP to an AppleTalk Network

To add Enhanced IGRP to an AppleTalk network, configure Enhanced IGRP on the interface that connects to the routers. Do not disable RTMP on the interfaces that connect to AppleTalk hosts or that connect to AppleTalk routers that do not support Enhanced IGRP. RTMP is the enabled by default when AppleTalk routing is enabled and when an interface is assigned an AppleTalk cable range.

In this case study, Routers D and E are running AppleTalk Enhanced IGRP. Routers F and G run both AppleTalk and AppleTalk Enhanced IGRP. Router G redistributes the routes from the AppleTalk network to the AppleTalk Enhanced IGRP network, and vice versa. (See Figure 4-12.)

Figure 4-12 Example of adding Enhanced IGRP to an AppleTalk network.



The configuration for Router G is as follows:

```
appletalk routing eigrp 1
interface ethernet 1
appletalk cable-range 125-125
appletalk zone Marketing Lab
appletalk protocol eigrp
interface serial 1
appletalk cable-range 126-126
appletalk zone WAN
appletalk protocol eigrp
no appletalk protocol rtmp
```

The configuration for Router F is as follows:

```
appletalk routing eigrp 2
interface serial 0
appletalk cable-range 126-126
appletalk zone WAN
appletalk protocol eigrp
no appletalk protocol rtmp
```

A **show appletalk route** command on Router G shows that the first set of routes is learned from an RTMP update, that the second set of routes is directly connected, and that the last route is learned by AppleTalk Enhanced IGRP via serial interface 1:

```
R Net 103-103 [1/G] via 125.220, 0 sec, Ethernet1, zone Marketing Lab
R Net 104-104 [1/G] via 125.220, 1 sec, Ethernet1, zone Marketing Lab
R Net 105-105 [1/G] via 125.220, 1 sec, Ethernet1, zone Marketing Lab
R Net 108-108 [1/G] via 125.220, 1 sec, Ethernet1, zone Marketing Lab
C Net 125-125 directly connected, Ethernet1, zone Marketing Lab
C Net 126-126 directly connected, Serial1, zone Wan
E Net 127-127 [1/G] via 126.201, 114 sec, Serial1, zone Networkers
```

A **show appletalk route** command on Router F shows that routes are learned from AppleTalk Enhanced IGRP:

```
E Net 103-103 [2/G] via 126.220, 519 sec, Serial0, zone Marketing Lab
E Net 104-104 [2/G] via 126.220, 520 sec, Serial0, zone Marketing Lab
E Net 105-105 [2/G] via 126.220, 520 sec, Serial0, zone Marketing Lab
E Net 108-108 [2/G] via 126.220, 520 sec, Serial0, zone Marketing Lab
E Net 125-125 [1/G] via 126.220, 520 sec, Serial0, zone Marketing Lab
C Net 126-126 directly connected, Serial0, zone Wan
C Net 127-127 directly connected, Ethernet1, zone Networkers
```

Route Selection

AppleTalk Enhanced IGRP routes are automatically preferred over Routing Table Maintenance Protocol (RTMP) routes. Whereas the AppleTalk metric for route determination is based on hop count only, AppleTalk Enhanced IGRP uses a combination of these configurable metrics: delay, bandwidth, reliability, and load.

Metric Handling

The formula for converting RTMP metrics to AppleTalk Enhanced IGRP metrics is hop count multiplied by 252524800. This is a constant based on the bandwidth for a 9.6-Kbps serial line and includes an RTMP factor. An RTMP hop distributed into Enhanced IGRP appears as a slightly worse path than an Enhanced IGRP-native, 9.6-Kbps serial link. The formula for converting Enhanced IGRP to RTMP is the value of the Enhanced IGRP external metric plus 1.

Redistribution

Redistribution between AppleTalk and Enhanced IGRP and vice versa is automatic by default. Redistribution involves converting the Enhanced IGRP metric back into an RTMP hop count metric. In reality, there is no conversion of an Enhanced IGRP composite metric into a RTMP metric. Because a hop count is carried in an Enhanced IGRP metric tuple as the Enhanced IGRP route spreads through the network, 1 is added to the hop-count carried in the Enhanced IGRP metric blocks through the network and put into any RTMP routing tuple generated.

There is no conversion of an Enhanced IGRP metric back into an RTMP metric because, in reality, what RTMP uses as a metric (the hop count) is carried along the Enhanced IGRP metric all the way through the network. This is true of Enhanced IGRP-derived routes and routes propagated through the network that were originally derived from an RTMP route.

Summary

This case study illustrates the integration of Enhanced IGRP in graduated steps, starting at the periphery of the network before adding Enhanced IGRP into the core network. With Enhanced IGRP for IP networks, route summarization and redistribution of routing updates are key considerations. To add Enhanced IGRP to IPX networks, it is critical to configure RIP and SAP on interfaces connecting to Novell hosts or routers that do not support Enhanced IGRP. When adding Enhanced IGRP to AppleTalk networks, turn off RTMP on the interfaces that are configured to support Enhanced IGRP.

CHAPTER 5

Reducing SAP Traffic in Novell IPX Networks

One of the limiting factors in the operation of large Novell Internetwork Packet Exchange (IPX) internetworks is the amount of bandwidth consumed by the large, periodic Service Advertisement Protocol (SAP) updates. Novell servers periodically send clients information about the services they provide by broadcasting this information onto their connected local-area network (LAN) or wide-area network (WAN) interfaces. Routers are required to propagate SAP updates through an IPX network so that all clients can see the service messages. It is possible to reduce SAP traffic on Novell IPX networks by the following means:

- *Filtering SAP updates through access lists*. SAP updates can be filtered by prohibiting routers from advertising services from specified Novell servers.
- *Configuring Cisco routers on Novell IPX networks to run Enhanced IGRP.* Although filters provide a means of *eliminating* the advertisements of specified services, Enhanced IGRP provides *incremental* SAP updates for a finer granularity of control. Complete SAP updates are sent periodically on each interface only until an IPX Enhanced IGRP neighbor is found. Thereafter, SAP updates are sent only when there are *changes* to the SAP table. In this way, bandwidth is conserved, and the advertisement of services is reduced without being eliminated.

  Incremental SAP updates are automatic on serial interfaces and can be configured on LAN media. Enhanced IGRP also provides partial routing updates and fast convergence for IPX networks. Administrators may choose to run only the partial SAP updates or to run both the reliable SAP protocol and the partial routing update portion of Enhanced IGRP.
- *Configuring Cisco routers on Novell IPX networks to send incremental SAP updates*. With Software Release 10.0, the incremental SAP updates just described can be configured for Cisco routers on Novell IPX networks, *without* the requirement of running the routing update feature of Enhanced IGRP (only the partial SAP updates are enabled). This feature is supported on all interface types. Again, SAP updates are sent only when changes occur on a network. Only the changes to SAP tables are sent as updates.

To illustrate how to reduce SAP traffic, this case study is organized into two parts:

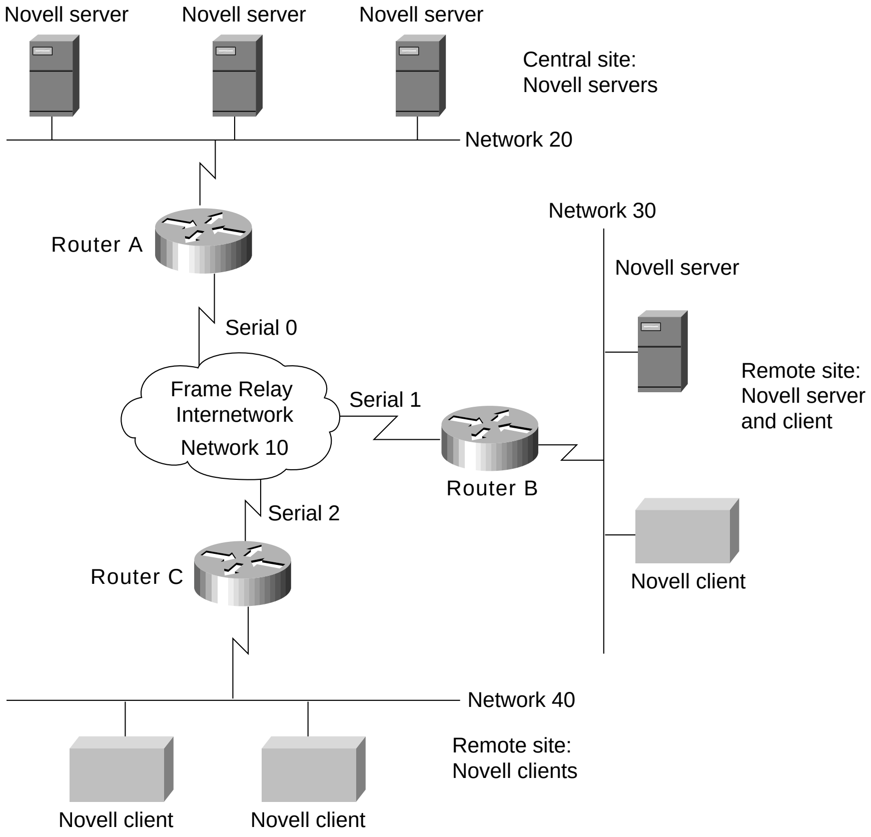
- Configuring Access Lists to Filter SAP Updates
- Configuring Incremental SAP Updates

Note For a detailed case study on configuring Novell IPX Enhanced IGRP, see the “Integrating Enhanced IGRP into Existing Networks” section in Chapter 4, “Integrating Enhanced IGRP into Existing Networks.”

The internetwork for this case study is illustrated in Figure 5-1. The following portions of a large-scale Novell IPX network spanning across a Frame Relay WAN are examined:

- Router A connects from the Frame Relay internetwork to the central site with three Novell servers.
- Router B connects from the Frame Relay internetwork to a remote site with one Novell client and one Novell server.
- Router C connects from the Frame Relay internetwork to a remote site with two Novell clients.

Figure 5-1 Large-scale Novell IPX internetwork.



Configuring Access Lists to Filter SAP Updates

Access lists can control which routers send or receive SAP updates and which routers do not send or receive SAP updates. SAP access lists can be defined to filter SAP updates based on the source network address of a SAP entry, the type of SAP entry (file server, print server, and so forth), and the name of the SAP server. A SAP access list is made up of entries in the following format:

```
access-list n [deny|permit] network[.node] [service-type[server-name]]
```

where *n* is between 1000-1099. A network number of -1 indicates any network, and a service type of 0 indicates any service. For example, the following access list accepts print server SAP entries from server PRINTER_1, all file servers, and any other SAP entries from network 123 except those from a server called UNTRUSTED; all other SAP entries are to be ignored:

```
access-list 1000 permit -1 47 PRINTER_1
access-list 1000 permit -1 4
access-list 1000 deny 123 0 UNTRUSTED
access-list 1000 permit 123
```

When checking the entries in a SAP update, each statement in the access list is processed in order, and if there is no match for a SAP entry, it is not accepted. Thus, to block server UNTRUSTED, the **deny** statement must be placed before the **permit** for all other devices on network 123.

Two techniques can be used with filtering. Either the SAP entries that are required can be permitted and the rest denied, or the unwanted SAP entries can be denied and the rest permitted. In general, the first method is preferred because it avoids new and unexpected services being propagated throughout the network.

The most common form of SAP filtering is to limit which services are available across a WAN. For example, it does not, in general, make sense for clients in one location to be able to access print servers in another location because printing is a local operation. In this case study, only file servers are permitted to be visible across the WAN.

Central Site

Router A connects to the central site. The following access lists configured on Router A permit everything except print servers from being announced out the serial interface:

```
access-list 1000 deny -1 47
access-list 1000 permit -1
!
interface serial 0
ipx network 10
ipx output-sap-filter 1000
```

To permit only IPX file servers and to deny all other IPX servers, use the following configuration:

```
access-list 1000 permit -1 4
!
interface serial 0
ipx network 10
ipx out-sap-filter 1000
```

Remote Sites

This section provides information on the configuration of the routers at the remote sites:

- Router B connected to an IPX server and client
- Router C connected to two IPX clients

IPX Server and Client

For Router B, the following access lists permit everything except print servers from being announced out the serial interface.

```
access-list 1000 deny -1 47
access-list 1000 permit -1
!
interface serial 1
ipx network 10
ipx output-sap-filter 1000
```

To permit only IPX file servers and to deny all other IPX servers, use the following configuration:

```
access-list 1000 permit -1 4
!
interface serial 1
ipx network 10
ipx out-sap-filter 1000
```

IPX Clients

Router C does not require an access list configuration because the remote site does not have any servers. Only Novell servers generate SAP updates.

Configuring Incremental SAP Updates

Incremental SAP updates allow any-to-any connectivity with reduced network SAP overhead. Instead of eliminating the receipt of SAP updates entirely, all necessary IPX services can be broadcast to remote sites only as changes to the SAP tables occur.

Central Site

To configure Enhanced IGRP encapsulated SAP updates to be sent only on a incremental basis, use the following configuration. Although the defined Enhanced IGRP autonomous system number is 999, Enhanced IGRP routing (and routing updates) are not performed because of the **rsup-only** keyword used with the **ipx sap-incremental** command. The **rsup-only** keyword indicates a reliable SAP update.

```
interface ethernet 0
ipx network 20
!
interface serial 0
ipx network 10
ipx sap-incremental eigrp 999 rsup-only
!
ipx router eigrp 999
network 10
```

To configure both incremental SAP and Enhanced IGRP routing, simply configure Enhanced IGRP with the following commands:

```
interface ethernet 0
ipx network 20
!
interface serial 0
ipx network 10
!
ipx router eigrp 999
network 10
```

Remote Sites

This section provides information on the configuration of the routers at the remote sites:

- Router B connected to an IPX server and client
- Router C connected to two IPX clients

IPX Server and Client

To configure Enhanced IGRP encapsulated SAP updates to be sent only on a incremental basis, use the following configuration for Router B. Although the defined Enhanced IGRP autonomous system number is 999, Enhanced IGRP routing is not performed because of the **rsup-only** keyword used with the **ipx sap-incremental** command.

```
interface ethernet 1
ipx network 30
!
interface serial 1
ipx network 10
ipx sap-incremental eigrp 999 rsup-only
!
ipx router eigrp 999
network 10
```

To configure both incremental SAP and Enhanced IGRP routing, simply configure Enhanced IGRP with the following commands:

```
interface ethernet 1
ipx network 30
!
interface serial 1
ipx network 10
!
ipx router eigrp 999
network 10
```

IPX Clients

To configure Enhanced IGRP encapsulated SAP updates to be sent only on a incremental basis, use the following configuration for Router C:

```
interface ethernet 2
ipx network 40
!
interface serial 2
ipx network 10
ipx sap-incremental eigrp 999 rsup-only
!
ipx router eigrp 999
network 10
```

Even though there are no servers, these configuration commands are required to support the incremental SAP updates being advertised from the central site and other remote sites to Router C.

Summary

This case study illustrates two methods of reducing SAP traffic on Novell IPX networks: the use of access lists to eliminate the advertisements of specified services, and the use of the incremental SAP feature to exchange SAP changes as they occur. This technique eliminates periodic SAP updates.

UDP Broadcast Flooding

A *broadcast* is a data packet that is destined for multiple hosts. Broadcasts can occur at the data link layer and the network layer. Data-link broadcasts are sent to all hosts attached to a particular physical network. Network layer broadcasts are sent to all hosts attached to a particular logical network. The Transmission Control Protocol/Internet Protocol (TCP/IP) supports the following types of broadcast packets:

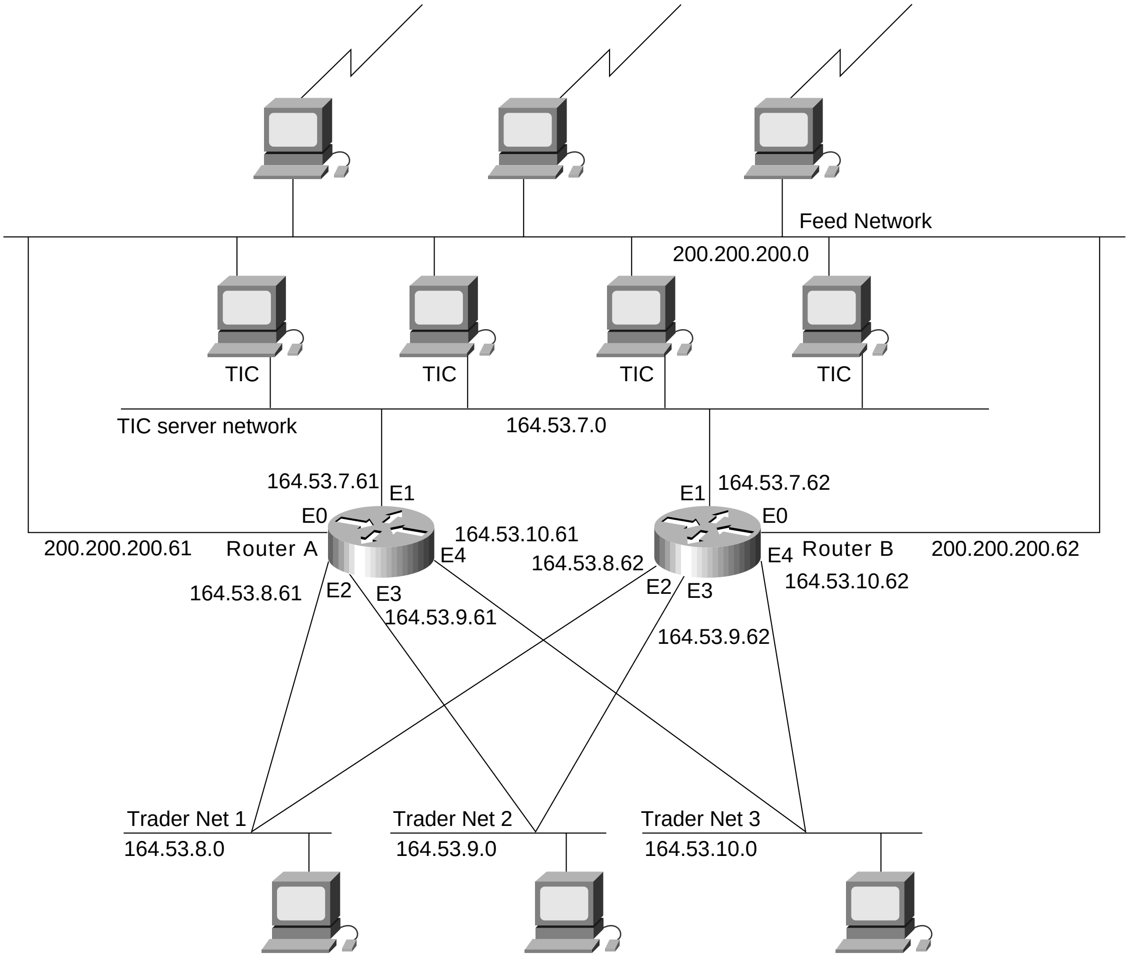
- *All ones*—By setting the broadcast address to all ones (255.255.255.255), all hosts on the network receive the broadcast.
- *Network*—By setting the broadcast address to a specific network number in the network portion of the IP address and setting all ones in the host portion of the broadcast address, all hosts on the specified network receive the broadcast. For example, when a broadcast packet is sent with the broadcast address of 131.108.255.255, all hosts on network number 131.108 receive the broadcast.
- *Subnet*—By setting the broadcast address to a specific network number and a specific subnet number, all hosts on the specified subnet receive the broadcast. For example, when a broadcast packet is set with the broadcast address of 131.108.4.255, all hosts on subnet 4 of network 131.108 receive the broadcast.

Because broadcasts are recognized by all hosts, a significant goal of router configuration is to control unnecessary proliferation of broadcast packets. Cisco routers support two kinds of broadcasts: *directed* and *flooded*. A directed broadcast is a packet sent to a specific network or series of networks, whereas a flooded broadcast is a packet sent to every network. In IP internetworks, most broadcasts take the form of User Datagram Protocol (UDP) broadcasts.

Although current IP implementations use a broadcast address of all ones, the first IP implementations used a broadcast address of all zeros. Many of the early implementations do not recognize broadcast addresses of all ones and fail to respond to the broadcast correctly. Other early implementations forward broadcasts of all ones, which causes a serious network overload known as a *broadcast storm*. Implementations that exhibit these problems include systems based on versions of BSD UNIX prior to Version 4.3.

In the brokerage community, applications use UDP broadcasts to transport market data to the desktops of traders on the trading floor. This case study gives examples of how brokerages have implemented both directed and flooding broadcast schemes in an environment that consists of Cisco routers and Sun workstations. Figure 6-1 illustrates a typical topology. Note that the addresses in this network use a 10-bit netmask of 255.255.255.192.

Figure 6-1 Topology that requires UDP broadcast forwarding.



In Figure 6-1, UDP broadcasts must be forwarded from a source segment (Feed network) to many destination segments that are connected redundantly. Financial market data, provided, for example, by Reuters, enters the network through the Sun workstations connected to the Feed network and is disseminated to the TIC servers. The TIC servers are Sun workstations running Teknekron Information Cluster software. The Sun workstations on the trader networks subscribe to the TIC servers for the delivery of certain market data, which the TIC servers deliver by means of UDP broadcasts. The two routers in this network provide redundancy so that if one router becomes unavailable, the other router can assume the load of the failed router without intervention from an operator. The connection between each router and the Feed network is for network administration purposes only and does not carry user traffic.

Two different approaches can be used to configure Cisco routers for forwarding UDP broadcast traffic: IP helper addressing and UDP flooding. This case study analyzes the advantages and disadvantages of each approach.

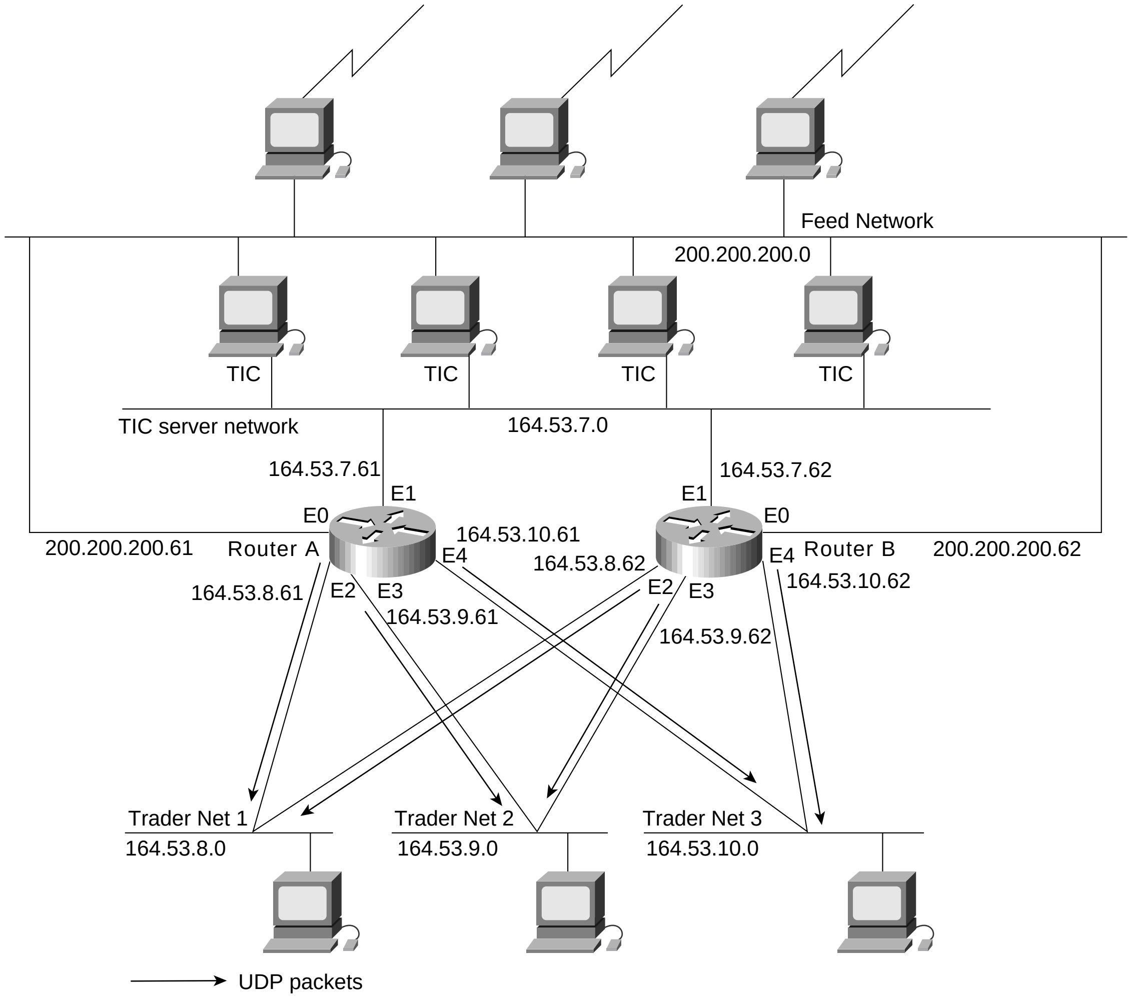
Note Regardless of whether you implement IP helper addressing or UDP flooding, you must use the **ip forward-protocol udp** global configuration command to enable the UDP forwarding. By default, the **ip forward-protocol udp** command enables forwarding for ports associated with the following protocols: Trivial File Transfer Protocol, Domain Name System, Time service, NetBIOS Name Server, NetBIOS Datagram Server, Boot Protocol, and Terminal Access Controller Access Control System. To enable forwarding for other ports, you must specify them as arguments to the **ip forward-protocol udp** command.

Implementing IP Helper Addressing

IP helper addressing is a form of static addressing that uses directed broadcasts to forward local and all-nets broadcasts to desired destinations within the internetwork.

To configure helper addressing, you must specify the **ip helper-address** command on every interface on every router that receives a broadcast that needs to be forwarded. On Router A and Router B, IP helper addresses can be configured to move data from the TIC server network to the trader networks. IP helper addressing in not the optimal solution for this type of topology because each router receives unnecessary broadcasts from the other router, as shown in Figure 6-2.

Figure 6-2 Flow of UDP packets from routers to trader networks using IP helper addressing.



In this case, Router A receives each broadcast sent by Router B *three times*, one for each segment, and Router B receives each broadcast sent by Router A three times, one for each segment. When each broadcast is received, the router must analyze it and determine that the broadcast does not need to be forwarded. As more segments are added to the network, the routers become overloaded with unnecessary traffic, which must be analyzed and discarded.

When IP helper addressing is used in this type of topology, no more than one router can be configured to forward UDP broadcasts (unless the receiving applications can handle duplicate broadcasts). This is because duplicate packets arrive on the trader network. This restriction limits redundancy in the design and can be undesirable in some implementations.

To send UDP broadcasts bidirectionally in this type of topology, a second **ip helper address** command must be applied to every router interface that receives UDP broadcasts. As more segments and devices are added to the network, more **ip helper address** commands are required to reach them, so the administration of these routers becomes more complex over time. Note, too, that bidirectional traffic in this topology significantly impacts router performance.

Although IP helper addressing is well-suited to nonredundant, nonparallel topologies that do not require a mechanism for controlling broadcast loops, in view of these drawbacks, IP helper addressing does not work well in this topology. To improve performance, network designers considered several other alternatives:

- *Setting the broadcast address on the TIC servers to all ones (255.255.255.255)*—This alternative was dismissed because the TIC servers have more than one interface, causing TIC broadcasts to be sent back onto the Feed network. In addition, some workstation implementations do not allow all ones broadcasts when multiple interfaces are present.
- *Setting the broadcast address of the TIC servers to the major net broadcast (164.53.0.0)*—This alternative was dismissed because the Sun TCP/IP implementation does not allow the use of major net broadcast addresses when the network is subnetted.
- *Eliminating the subnets and letting the workstations use Address Resolution Protocol (ARP) to learn addresses*—This alternative was dismissed because the TIC servers cannot quickly learn an alternative route in the event of a primary router failure.

With alternatives eliminated, the network designers turned to a simpler implementation that supports redundancy without duplicating packets and that ensures fast convergence and minimal loss of data when a router fails: UDP flooding.

Implementing UDP Flooding

UDP flooding uses the spanning tree algorithm to forward packets in a controlled manner. Bridging is enabled on each router interface for the sole purpose of building the spanning tree. The spanning tree prevents loops by stopping a broadcast from being forwarded out an interface on which the broadcast was received. The spanning tree also prevents packet duplication by placing certain interfaces in the blocked state (so that no packets are forwarded) and other interfaces in the forwarding state (so that packets that need to be forwarded are forwarded).

To enable UDP flooding, the router must be running software that supports transparent bridging and bridging must be configured on each interface that is to participate in the flooding. If bridging is not configured for an interface, the interface will receive broadcasts, but the router will not forward those broadcasts and will not use that interface as a destination for sending broadcasts received on a different interface.

Note Releases prior to Cisco Internetwork Operating System (Cisco IOS) Software Release 10.2 do not support flooding subnet broadcasts.

When configured for UPD flooding, the router uses the destination address specified by the **ip broadcast-address** command on the output interface to assign a destination address to a flooded UDP datagram. Thus, the destination address might change as the datagram propagates through the network. The source address, however, does not change.

With UDP flooding, both routers shown in Figure 19-1 use a spanning tree to control the network topology for the purpose of forwarding broadcasts. The key commands for enabling UDP flooding are as follows:

```
bridge group protocol protocol
ip forward-protocol spanning tree
bridge-group group input-type-list access-list-number
```

The **bridge protocol** command can specify either the **dec** keyword (for the DEC spanning-tree protocol) or the **ieee** keyword (for the IEEE Ethernet protocol). All routers in the network must enable the same spanning tree protocol. The **ip forward-protocol spanning tree** command uses the database created by the **bridge protocol** command. Only one broadcast packet arrives at each segment, and UDP broadcasts can traverse the network in both directions.

Note Because bridging is enabled only to build the spanning tree database, use access lists to prevent the spanning tree from forwarding non-UDP traffic. The configuration examples later in this chapter configure an access list that blocks all bridged packets.

To determine which interface forwards or blocks packets, the router configuration specifies a path cost for each interface. The default path cost for Ethernet is 100. Setting the path cost for each interface on Router B to 50 causes the spanning tree algorithm to place the interfaces in Router B in forwarding state. Given the higher path cost (100) for the interfaces in Router A, the interfaces in Router A are in the blocked state and do not forward the broadcasts. With these interface states, broadcast traffic flows through Router B. If Router B fails, the spanning tree algorithm will place the interfaces in Router A in the forwarding state, and Router A will forward broadcast traffic.

With one router forwarding broadcast traffic from the TIC server network to the trader networks, it is desirable to have the other forward unicast traffic. For that reason, each router enables the ICMP Router Discovery Protocol (IRDP), and each workstation on the trader networks runs the **irdp** daemon. On Router A, the **preference** keyword sets a higher IRDP preference than does the configuration for Router B, which causes each **irdp** daemon to use Router A as its preferred default gateway for unicast traffic forwarding. Users of those workstations can use **netstat -rn** to see how the routers are being used.

On the routers, the **holdtime**, **maxadvertinterval**, and **minadvertinterval** keywords reduce the advertising interval from the default so that the **irdp** daemons running on the hosts expect to see advertisements more frequently. With the advertising interval reduced, the workstations will adopt Router B more quickly if Router A becomes unavailable. With this configuration, when a router becomes unavailable, IRDP offers a convergence time of less than one minute.

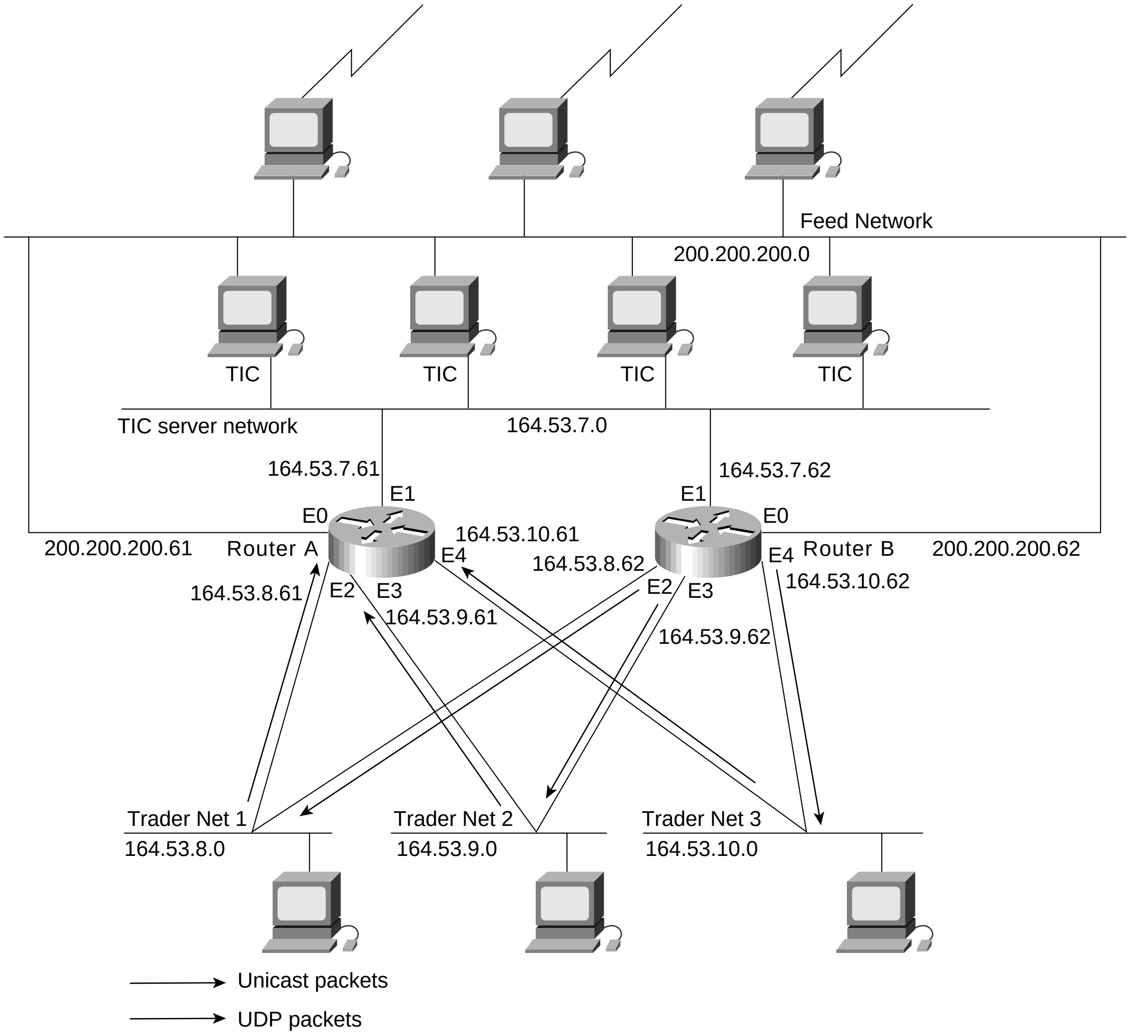
IRDP is preferred over the Routing Information Protocol (RIP) and default gateways for the following reasons:

- RIP takes longer to converge, typically from one to two minutes.
- Configuration of Router A as the default gateway on each Sun workstation on the trader networks would allow those Sun workstations to send unicast traffic to Router A, but would not provide an alternative route if Router A becomes unavailable.

Note Some workstation vendors include an **irdp** daemon with their operating systems. Source code for an **irdp** daemon is available by anonymous FTP at *ftp.cisco.com*.

Figure 6-3 shows how data flows when the network is configured for UDP flooding.

Figure 6-3 Data flow with UDP flooding and IRDP.



Note This topology is broadcast intensive—broadcasts sometimes consume 20 percent of the Ethernet bandwidth. However, this is a favorable percentage when compared to the configuration of IP helper addressing, which, in the same network, causes broadcasts to consume up to 50 percent of the Ethernet bandwidth.

If the hosts on the trader networks do not support IRDP, the Hot Standby Routing Protocol (HSRP) can be used to select which router will handle unicast traffic. HSRP allows the standby router to take over quickly if the primary router becomes unavailable. For information about configuring HSRP, see Chapter 9, "Using HSRP for Fault-Tolerant IP Routing."

By default, the router performs UDP flooding by process switching the UDP packets. To increase performance on AGS+ and Cisco 7000 series routers, enable fast switching of UDP packets by using the following command:

```
ip forward-protocol turbo-flood
```

Note Turbo flooding increases the amount of processing that is done at interrupt level, which increases the CPU load on the router. Turbo flooding may not be appropriate on routers that are already under high CPU load or that must also perform other CPU-intensive activities.

The following commands configure UDP flooding on Router A. Because this configuration does not specify a lower path cost than the default and because the configuration of Router B specifies a lower cost than the default with regard to UDP flooding, Router A acts as a backup to Router B. Because this configuration specifies an IRDP preference of 100 and because Router B specifies a IRDP preference of 90 (**ip irdp preference 90**), Router A forwards unicast traffic from the trader networks, and Router B is the backup for unicast traffic forwarding.

```
!Router A:
ip forward-protocol spanning-tree
ip forward-protocol udp 111
ip forward-protocol udp 3001
ip forward-protocol udp 3002
ip forward-protocol udp 3003
ip forward-protocol udp 3004
ip forward-protocol udp 3005
ip forward-protocol udp 3006
ip forward-protocol udp 5020
ip forward-protocol udp 5021
ip forward-protocol udp 5030
ip forward-protocol udp 5002
ip forward-protocol udp 1027
ip forward-protocol udp 657
!
interface ethernet 0
ip address 200.200.200.61 255.255.255.0
ip broadcast-address 200.200.200.255
no mop enabled
!
interface ethernet 1
ip address 164.53.7.61 255.255.255.192
ip broadcast-address 164.53.7.63
ip irdp
ip irdp maxadvertinterval 60
ip irdp minadvertinterval 45
ip irdp holdtime 60
ip irdp preference 100
bridge-group 1
bridge-group 1 input-type-list 201
no mop enabled
!
interface ethernet 2
ip address 164.53.8.61 255.255.255.192
ip broadcast-address 164.53.8.63
ip irdp
ip irdp maxadvertinterval 60
ip irdp minadvertinterval 45
ip irdp holdtime 60
ip irdp preference 100
bridge-group 1
bridge-group 1 input-type-list 201
no mop enabled
!
interface ethernet 3
ip address 164.53.9.61 255.255.255.192
ip broadcast-address 164.53.9.63
ip irdp
ip irdp maxadvertinterval 60
ip irdp minadvertinterval 45
ip irdp holdtime 60
ip irdp preference 100
bridge-group 1
bridge-group 1 input-type-list 201
no mop enabled
!
interface ethernet 4
ip address 164.53.10.61 255.255.255.192
ip broadcast-address 164.53.10.63
ip irdp
ip irdp maxadvertinterval 60
ip irdp minadvertinterval 45
ip irdp holdtime 60
ip irdp preference 100
bridge-group 1
bridge-group 1 input-type-list 201
no mop enabled
!
router igrp 1
network 164.53.0.0
!
ip name-server 255.255.255.255
snmp-server community public RW
snmp-server host 164.53.7.15 public
bridge 1 protocol dec
bridge 1 priority 255
access-list 201 deny   0xFFFF 0x0000
```

The following commands configure UDP flooding on Router B. Because this configuration specifies a lower path cost than the default (**bridge-group 1 path-cost 50**) and because the configuration of Router A accepts the default, Router B forwards UDP packets. Because this configuration specifies an IRDP preference of 90 (**ip irdp preference 90**) and because Router A specifies a IRDP preference of 100, Router B acts as the backup for Router A for forwarding unicast traffic from the trader networks.

```
!Router B
ip forward-protocol spanning-tree
ip forward-protocol udp 111
ip forward-protocol udp 3001
ip forward-protocol udp 3002
ip forward-protocol udp 3003
ip forward-protocol udp 3004
ip forward-protocol udp 3005
ip forward-protocol udp 3006
ip forward-protocol udp 5020
ip forward-protocol udp 5021
ip forward-protocol udp 5030
ip forward-protocol udp 5002
ip forward-protocol udp 1027
ip forward-protocol udp 657
!
interface ethernet 0
ip address 200.200.200.62 255.255.255.0
ip broadcast-address 200.200.200.255
no mop enabled
!
interface ethernet 1
ip address 164.53.7.62 255.255.255.192
ip broadcast-address 164.53.7.63
ip irdp
ip irdp maxadvertinterval 60
ip irdp minadvertinterval 45
ip irdp holdtime 60
ip irdp preference 90
bridge-group 1
bridge-group 1 path-cost 50
bridge-group 1 input-type-list 201
no mop enabled
!
interface ethernet 2
ip address 164.53.8.62 255.255.255.192
ip broadcast-address 164.53.8.63
ip irdp
ip irdp maxadvertinterval 60
ip irdp minadvertinterval 45
ip irdp holdtime 60
ip irdp preference 90
bridge-group 1
bridge-group 1 path-cost 50
bridge-group 1 input-type-list 201
no mop enabled
!
interface ethernet 3
ip address 164.53.9.62 255.255.255.192
ip broadcast-address 164.53.9.63
ip irdp
ip irdp maxadvertinterval 60
ip irdp minadvertinterval 45
ip irdp holdtime 60
ip irdp preference 90
bridge-group 1
bridge-group 1 path-cost 50
bridge-group 1 input-type-list 201
no mop enabled
!
interface ethernet 4
ip address 164.53.10.62 255.255.255.192
ip broadcast-address 164.53.10.63
ip irdp
ip irdp maxadvertinterval 60
ip irdp minadvertinterval 45
ip irdp holdtime 60
ip irdp preference 90
bridge-group 1
bridge-group 1 path-cost 50
bridge-group 1 input-type-list 201
no mop enabled
!
router igrp 1
network 164.53.0.0
!
ip name-server 255.255.255.255
snmp-server community public RW
snmp-server host 164.53.7.15 public
bridge 1 protocol dec
bridge 1 priority 255
access-list 201 deny 0xFFFF 0x0000
```

Note In releases prior to Cisco IOS Software Release 10.2, the spanning tree algorithm prevented the forwarding of local broadcast addresses, but allowed the forwarding of local secondary addresses. For that reason, when running a release prior to Cisco IOS Software Release 10.2, a secondary address must be specified for each Ethernet interface that will forward local broadcast address packets. The secondary address is used to forward packets, whereas the primary address is never used. In such configurations, the secondary addresses are assigned to an Interior Gateway Routing Protocol (IGRP) group instead of the primary address.

Summary

Although IP helper addressing is useful in networks that do not require redundancy, when configured in networks that feature redundancy, IP helper addressing results in packet duplication that severely reduces router and network performance.

By configuring UDP flooding, one router forwards UDP traffic without burdening the second router with duplicate packets. By dedicating one router to the task of forwarding UDP traffic, the second router becomes available for forwarding unicast traffic. At the same time, because each router is configured as the backup for the other router, redundancy is maintained; if either router fails, the other router can assume the work of the failed router without intervention from an operator. When compared with IP helper addressing, UDP flooding makes the most efficient use of router resources.

STUN for Front-End Processors

Serial tunneling (STUN) enables the integration of traditional *systems network architecture* (SNA) networks with multiprotocol networks. STUN also lowers operating costs by reducing the need for redundant remote wide-area links. This case study explores three implementations of STUN between Cisco routers and front-end processors (FEPs):

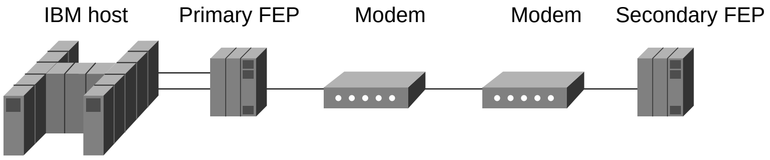
- *Basic STUN*—Presents a STUN implementation that is simple and quick to configure because it does not require the specification of addresses. This implementation is recommended for networks that do not require synchronous data link control (SDLC) address checking or local acknowledgment.
- *SDLC STUN*—Presents a STUN implementation that includes the configuration of addresses. This implementation is recommended for networks that require SDLC address checking.
- *SDLC-Transmission Group STUN*—Presents a STUN implementation that supports enhanced FEP-to-FEP communications features, such as transmission groups, as well as advanced router features. This implementation is recommended for networks that require local acknowledgment.

Note This case study introduces basic SNA concepts, but does not discuss them in detail. For additional information, see Chapters 2–4.

Understanding FEP Configuration

In a traditional SNA environment, serial lines connect FEPs in a master-slave topology, as shown in Figure 7-1. The primary FEP is connected to the IBM host, which is typically an IBM 3090 mainframe. Synchronous modems connect the FEPs.

Figure 7-1 Map of a traditional SNA network.



The software running on the FEP is called the *Network Control Program* (NCP). This section describes NCP configuration parameters and optional NCP features that network administrators must consider when they introduce routers into an FEP environment.

Serial Connections

Typically, a serial port on a line interface card in the FEP connects the FEP to a synchronous modem. Depending on the type of line interface card, the serial port may be EIA/TIA-232 or V.35. The modem acts as data communications equipment (DCE) and provides clocking and synchronization. The FEP acts as data terminal equipment (DTE). The NCP statement that configures the FEP for DTE is CLOCKNG=EXT.

Primary and Secondary Roles

The FEPs dynamically determine their primary and secondary roles. Typically, the FEP with the higher subarea address becomes the primary FEP. In some versions of NCP, the role parameter is configurable. Typically, the local FEP (the closest to the mainframe) is the primary FEP, whereas the remote FEP is the secondary FEP.

NRZ and NRZI Encoding

The NRZI parameter specifies whether the FEP should operate in nonreturn-to-zero inverted (NRZI) mode or in nonreturn-to-zero (NRZ) mode. Both techniques encode binary data on a synchronous serial line. The specification depends on the way the modem operates. Old modems and satellite links that are not sensitive to a pattern of repeated binary ones and zeros (that is, 101010...) operate in NRZI mode. Modems that are sensitive to repeated patterns operate in NRZ mode.

The NCP statement that configures the FEP for NRZI is NRZI=YES, which is the default and is correct for most IBM modems. The NCP statement that configures the FEP for NRZ is NRZI=NO, which is correct for most non-IBM modems.

Note All devices on the same SDLC link must use the same encoding scheme.

MODULO and MAXOUT Parameters

The MODULO parameter specifies the number of information frames (I-frames) that NCP can send to the remote device before receiving an acknowledgment. The statement MODULO=8 allows NCP to send seven unacknowledged I-frames, whereas the statement MODULO=128 and the statement MAXOUT=127 allows NCP to send 127 unacknowledged I-frames. (Note that when the MODULO parameter is set to 128, the NCP MAXOUT parameter specifies the number of I-frames that can be sent before receiving an acknowledgment. MAXOUT can range from 8 [the default] to 127.)

Typically, network administrators configure NCP to allow a high number of outstanding I-frames (that is, MODULO=128 and MAXOUT=127) for slow links or for satellite links. Allowing a high number of outstanding I-frames uses the link more efficiently by reducing the number of acknowledgments and by preventing session timeouts. When the MODULO parameter is 128, make sure the TCP output queue on the router is greater than 128.

The SDLC STUN implementation supports setting the MODULO parameter to 8 as well as 128. Note, however, that setting the MODULO parameter to any legal value other than 8 causes the router to use additional buffers to store unacknowledged I-frames.

When local acknowledgment is configured to reduce supervisory frame traffic and to prevent session timeouts, 8 is the only supported value of the MODULO parameter. When the MODULO value is 8, the router does not use additional buffers unnecessarily.

ADDRESS Parameter

The ADDRESS parameter has the following format: ADDRESS=(*line-number*, *mode*).

If *mode* is FULL, the FEP can send and receive data at the same time. When mode is HALF, the FEP is limited to sending data and then receiving data. The default value of *mode* is FULL. The value of mode affects the operation of the DUPLEX parameter. For more information, see the "DUPLEX Parameter" section later. The value of *line-number* specifies the channel adapter position or the relative line number of all the telecommunication links defined for this NCP.

When implementing SDLC STUN or SDLC-Transmission Group STUN, the network administrator must specify SDLC addresses in the configuration of the router. The addresses specified in the router configuration are based on the order in which the ADDRESS parameter appears in the NCP configuration. Consider the following NCP configuration:

```
*********************************************************************
*        LOCAL NCP LINKS -- PRIMARY FEP                             *
*********************************************************************

LINK04   GROUP LNCTL=SDLC,                GROUP LEVEL               X
               NPACOLL=YES,               <== 3745 Dallas           X
               DUPLEX=FULL,                                         X
               NEWSYNC=NO,                                          X
               NRZI=NO,                                             X
               SDLCST=(CPRI4,CSEC4),                                X
               RETRIES=(10,5,10),         PU LEVEL                  X
               IRETRY=YES,                                          X
               MAXOUT=7,                                            X
               PASSLIM=254,                                         X
               SERVLIM=254,                                         X
               ISTATUS=ACTIVE,            VTAM-ONLY LEVEL           X
               OWNER=CMC
*
*--------------------------------------------------------------------
*
X1010442 LINE  ADDRESS=(005,FULL)         <== 3745 Chicago (01)
S1010442 PU    PUTYPE=4
*
X1030442 LINE  ADDRESS=(132,FULL)         <== 3745 Raleigh (02)
S1030442 PU    PUTYPE=4
*
X1010446 LINE  ADDRESS=(068,FULL),MODULO=128,ISTATUS=ACTIVE, <== 3745 Houston (03) X
               SPEED=56000,SDLCST=(S04PRI,S04SEC)
S1010446 PU    PUTYPE=4,MAXOUT=63
*
X1020412 LINE  ADDRESS=(100,FULL),MODULO=128,ISTATUS=ACTIVE, <== 3745 Lafayette (04) X
               SPEED=56000,SDLCST=(S04PRI,S04SEC)
S1020412 PU    PUTYPE=4,MAXOUT=63
*
X1010412 LINE  ADDRESS=(112,FULL),SPEED=56000,ISTATUS=ACTIVE <== 3745 Atlanta (05) X
S1010412 PU    PUTYPE=4
*
X1010462 LINE  ADDRESS=(080,FULL),       <== 3745 San Francisco (06) X
               NRZI=NO,                                              X
               NEWSYNC=NO,                                           X
               DUPLEX=FULL,                                          X
               ISTATUS=ACTIVE,                                       X
               SERVLIM=254,                                          X
               SDLCST=(S04PRI,S04SEC),                               X
               MODULO=128,                                           X
               SPEED=56000
S1010462 PU    PUTYPE=4,                                             X
               MAXOUT=63
*
**********************************************************************
```

Given this configuration, the router configuration uses address 01 for Chicago, address 02 for Raleigh, address 03 for Houston, address 04 for Lafayette, address 05 for Atlanta, and address 06 for San Francisco.

DUPLEX Parameter

The DUPLEX parameter specifies whether the communication line and the modem operate in full- or half-duplex mode, and controls the Request To Send (RTS) signal. If the ADDRESS parameter specifies that the mode is FULL, the value of the DUPLEX parameter has no effect, and RTS is always high (that is, permanent RTS). If the ADDRESS parameter specifies that the mode is HALF, the following applies:

- The statement DUPLEX=FULL causes RTS to be permanently high regardless of whether the FEP is sending or receiving data.
- The statement DUPLEX=HALF causes RTS to be high only when the FEP is sending data.

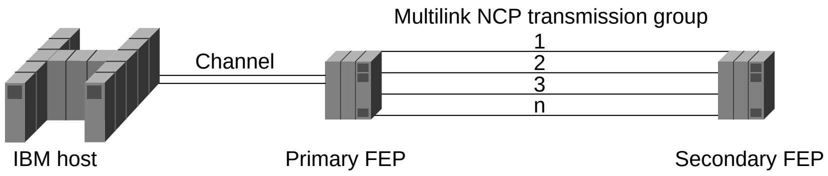
Enhanced NCP Features

This section describes the following enhanced NCP features that are supported by Cisco routers: transmission groups, echo addressing, and remote NCP loading. Note, however, that the basic STUN and SDLC STUN implementations do not support transmission groups.

Transmission Groups

A *transmission group* is one or more parallel SDLC links that connect FEPs. Transmission groups increase the reliability of the logical link connection between FEPs and provide additional bandwidth. When one link fails or congests, NCP uses one of the other links in the group to send data (see Figure 7-2).

Figure 7-2 Map of a network that uses transmission groups.



NCP uses virtual routes to provide more than one route between two FEPs. This multiple active routing mechanism increases the probability that an SDLC route is available when a session needs to be established.

Note SDLC-TG STUN is the only implementation that supports transmission groups.

Echo Addressing

Later versions of NCP use echo addressing. With echo addressing, the secondary FEP sets the high-order bit of the SDLC address when sending a response to the primary FEP. For example, the primary FEP sends frames with address 0x01, and the secondary FEP sends frames with address 0x81. This addressing scheme limits the range of SDLC addresses from 0x01 to 0x7F. Although echo addressing is a violation of the SDLC standard, it is supported because it occurs only between FEPs.

Note Echo addressing is implicitly supported by the basic STUN implementation because that implementation does not perform any address checking. The **echo** keyword of the **sdlc address** interface configuration command configures support for echo addressing in the SDLC STUN and SDLC-TG STUN implementations.

Remote NCP Loading

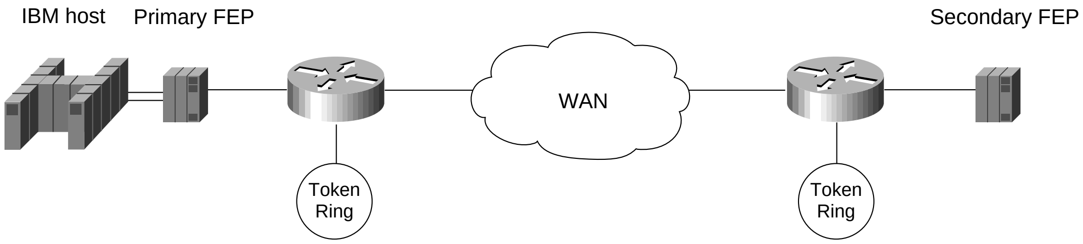
When a local FEP is loading a remote FEP with a new NCP configuration, the local FEP uses a nonstandard form of SDLC to complete the remote NCP load. This violation of the SDLC standard is supported because it occurs only between FEPs.

Note The basic STUN implementation implicitly supports remote NCP loading. When used with the **stun protocol-group** command, the **sdlc-tg** keyword automatically includes support for remote NCP loading in the SDLC-TG STUN implementation.

Understanding FEP-to-FEP Communications with Routers

Figure 7-3 illustrates the topology of an FEP-based network that includes routers. In this multiprotocol topology, the routers already handle traffic between Token Rings and the IBM host. When used to handle traffic between the FEPs, the routers replace the modems and lines that formerly connected the FEPs.

Figure 7-3 Map showing the addition of routers.



An EIA/TIA-232 (formerly RS-232) cable or a V.35 cable connects each router to its FEP, and a serial T1 line connects each router to the wide-area network (WAN). The FEPs continue to act as DTE devices, and, by providing clocking and synchronization, the serial interfaces on the routers act as DCE devices.

Advanced Router Features

When configured for STUN, Cisco routers can take advantage of the following advanced router features: priority queuing, custom queuing, and local acknowledgment.

Note When priority queuing or custom queuing is enabled, the router takes longer to switch packets because the processor card has to classify each packet.

Priority Queuing

Priority queuing allows the network administrator to set priorities on the traffic that passes through the network. Packets are classified according to various criteria, including protocol and subprotocol type, and then queued on one of four output queues: high, medium, normal, or low.

A FEP-to-FEP STUN implementation can use priority queuing to prioritize SNA traffic over other protocols that share the same link. The following commands distribute transmission control protocol (TCP) traffic among the four queues and assign STUN traffic encapsulated in TCP to the high queue:

```
priority-list 1 ip high tcp 1994
priority-list 1 ip medium tcp 1990
priority-list 1 ip normal tcp 1991
priority-list 1 ip low tcp 1992
priority-list 1 stun high
!
interface serial 0
encapsulation stun
stun group 1
sdlc address 01
stun route address 01 tcp 1.1.1.2 local-ack
priority-group 1
```

Note Configure the **priority-group** interface configuration command on the STUN input interface.

Custom Queuing

Custom queuing, available in Software Release 9.21 and subsequent software releases, is a queuing strategy that imparts a measure of fairness not provided by priority queuing. The network administrator can control on each interface the minimum percentage of bandwidth allocated to a particular kind of traffic.

When custom queuing is enabled on an interface, the router maintains for that interface eleven output queues (numbered 0 to 10). The router reserves queue number 0 for its own use. The router cycles sequentially through queue numbers 1 to 10, delivering packets in the current queue before moving to the next queue.

Each output queue has an associated configurable byte count that specifies how many bytes of data the router should deliver from the current queue before it moves to the next queue. When the router processes a particular queue, it sends packets until the number of bytes sent exceeds the queue byte count or until the queue is empty.

Custom queuing can be used instead of, but not in addition to, the **priority-group** interface configuration command in a single interface. The following configuration commands place STUN traffic on queue 1 with a byte-count limit of 4000 bytes and a maximum of 40 queues:

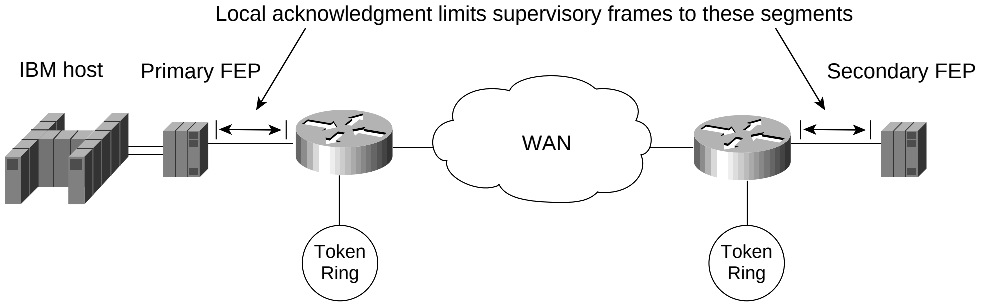
```
stun peer-name 1.1.1.1
stun protocol-group 1 sdlc-tg
!
interface serial 0
encapsulation stun
stun route address 01 tcp 1.1.1.2 local-ack
!
interface serial 1
encapsulation hdlc
custom-queue-list 1
!
queue-list 1 protocol stun 1
queue-list 1 protocol novell 2
queue-list 1 default 3
queue-list 1 queue 1 byte-count 4000
queue-list 1 queue 1 limit 40
```

Note Configure the **custom-queue-list** interface configuration command on the output interface that connects to the WAN.

Local Acknowledgment

Local acknowledgment is a router feature that prohibits supervisory frames from traversing the WAN, as shown in Figure 7-4.

Figure 7-4 Local acknowledgment limits the range of supervisory frames.


Local acknowledgment limits supervisory frames to these segments
IBM host
Primary FEP
Secondary FEP
WAN
Token Ring
Token Ring

Cisco recommends the use of local acknowledgment when one or both of the following conditions exist:

- *WAN link use is high*—When local acknowledgment is configured, supervisory frames, such as Receiver Ready (RR), Receiver Not Ready (RNR), and Reject (REJ), do not traverse the WAN link. Instead, supervisory frames are locally acknowledged by the router, which reduces the amount of traffic on the WAN link.
- *Network delay causes NCP timers to expire*—Link congestion, busy local-area networks, or high end-station use can cause excessive network delays, which can result in delayed acknowledgment of I-frames. When configured for local acknowledgment, the router acknowledges I-frames locally, which helps to prevent NCP timers from timing out and closing existing sessions.

Basic STUN

Basic STUN is easy to configure because it does not require the router configuration to match line addresses that may be configured on the FEPs. The mainframe sends data to the primary FEP, which passes the data to its router. The router for the primary FEP passes the data over an arbitrary medium (serial, fiber distributed data interface [FDDI], Token Ring, or Ethernet) to the router for the secondary FEP, and the router for the secondary FEP sends the data to the secondary FEP. Data from the secondary FEP flows to the mainframe by the reverse path. Network administrators use basic STUN for three purposes:

- *To accommodate existing addressing schemes*—Some NCP configurations use nonstandard addresses. For example, some configurations use address 0x00 or 0xFF for broadcasts and address 0xC1 for communication. By configuring the router for basic STUN, the network administrator does not have to configure the router to match existing addressing schemes.
- *To test connectivity*—When network administrators plan to implement SDLC STUN or SDLC-TG STUN, they often implement basic STUN first to verify physical connections.
- *To improve performance*—Because basic STUN requires minimal processing, it passes frames faster than SDLC STUN and SDLC-TG STUN.

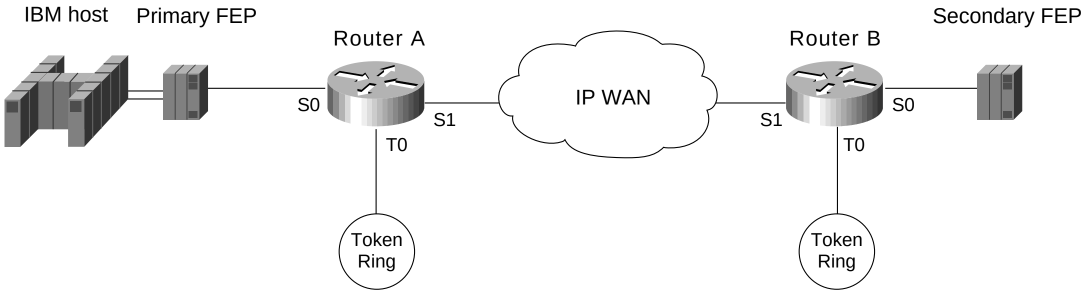
The basic STUN implementation has the following limitations:

- Lack of support for transmission groups, as well as lack of support for advanced router features. For information about advanced router features, see the “SDLC-Transmission Group STUN” section later in this chapter.
- Limited output from the router debugging commands.
- Lack of support for multidrop environments. (However, multidrop support is usually a requirement for cluster controller environments rather than FEP environments.)

Basic STUN Configuration: Example 1

In Figure 7-5, the routers pass data over an IP WAN. The FEPs are configured for DTE, full-duplex mode, and NRZ encoding. The serial interfaces on the routers are configured for DCE.

Figure 7-5 Topology for basic STUN: example 1.



The following commands configure basic STUN (example 1) for Router A:

```
stun peer-name 1.1.1.1
stun protocol-group 1 basic
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun route all tcp 1.1.2.1
clockrate 19200
!
interface tokenring 0
ip address 1.1.4.1 255.255.255.0
!
interface serial 1
ip address 1.1.3.1 255.255.255.0
!
interface loopback 0
ip address 1.1.1.1 255.255.255.0
!
router igrp 1
network 1.0.0.0
```

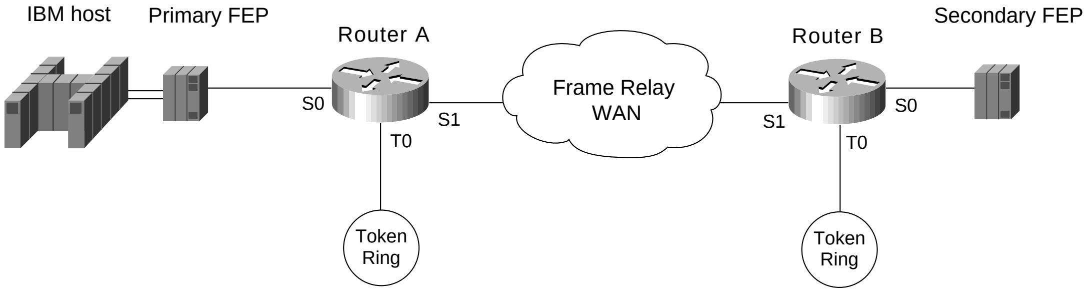
The following commands configure basic STUN (example 1) for Router B:

```
stun peer-name 1.1.2.1
stun protocol-group 1 basic
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun route all tcp 1.1.1.1
clockrate 19200

interface tokenring 0
ip address 1.1.5.1 255.255.255.0
!
interface serial 1
ip address 1.1.3.2 255.255.255.0
!
interface loopback 0
ip address 1.1.2.1 255.255.255.0
!
router igrp 1
network 1.0.0.0
```

Basic STUN Configuration: Example 2

In Figure 7-6, the routers transmit data over a Frame Relay WAN. The FEPs are configured for DTE, full-duplex mode, and NRZI encoding. The serial interfaces on the routers are configured for DCE.

Figure 7-6 Topology for basic STUN: example 2.



The following commands configure basic STUN (example 2) for Router A:

```
stun peer-name 1.1.1.1
stun protocol-group 1 basic
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun route all tcp 1.1.2.1
nrzi-encoding
clockrate 56000
!
interface tokenring 0
ip address 1.1.4.1 255.255.255.0
!
interface serial 1
ip address 1.1.3.1 255.255.255.0
encapsulation frame-relay
frame-relay map ip 1.1.3.2 40 broadcast
!
interface loopback 0
ip address 1.1.1.1 255.255.255.0
!
router igrp 1
network 1.0.0.0
```

The following commands configure basic STUN (example 2) for Router B:

```
stun peer-name 1.1.2.1
stun protocol-group 1 basic
!
interface serial 0
no ip address
encapsulation stun
stun group 1
stun route all tcp 1.1.1.1
nrzi-encoding
clockrate 56000
!
interface tokenring 0
ip address 1.1.5.1 255.255.255.0


interface serial 1
ip address 1.1.3.2 255.255.255.0
encapsulation frame-relay
frame-relay map ip 1.1.3.1 40 broadcast
!
interface loopback 0
ip address 1.1.2.1 255.255.255.0
!
router igrp 1
network 1.0.0.0
```

Note NRZ encoding is the default for all Cisco routers. NRZI encoding is software configurable for Cisco 250*x*, Cisco 3*x*04, Cisco 4000 4T, and Cisco 7000 routers. NRZI encoding is hardware configurable for Cisco 4000 2T and AGS+ routers. Full-duplex mode is the default for all router serial cards. Half-duplex mode is software configurable for the Cisco 4000 4T and Cisco 250*x* routers and is hardware configurable on the EIA/TIA-232/H applique for the AGS+.

SDLC STUN

SDLC STUN is the most commonly used tunneling configuration in Cisco multiprotocol networks. It is frequently implemented for gateways and cluster controllers. SDLC STUN uses the standard SDLC protocol. In most cases, IBM FEPs and compatible FEPs comply with that standard. If an FEP uses a nonstandard form of SDLC, the router must be configured for basic STUN.

The SDLC STUN implementation requires coordination of SDLC addresses between the router and the NCP configuration. To configure the router for SDLC STUN, the network administrator must know the relative position of the ADDRESS parameters in the NCP configuration. For details, see the earlier “ADDRESS Parameter” section. Network administrators use SDLC STUN for two purposes:

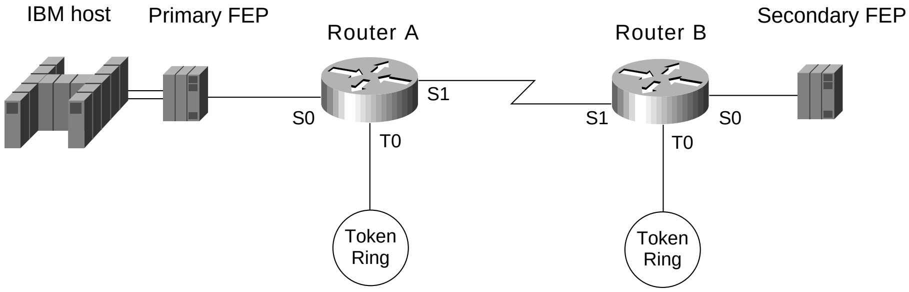
- *To support specific addressing schemes*—SDLC STUN allows the network administrator to configure specific line addresses. SDLC STUN is required in certain environments, such as multidrop, that depend on specific addresses.
- *To support network tuning and monitoring*—Occasionally, the network administrator needs to tune and monitor the SNA SDLC and multiprotocol network.

SDLC STUN is limited by its lack of support for transmission groups.

Configuring SDLC STUN

In Figure 7-7, the routers transmit data over a serial line. The FEPs are configured for DTE, full-duplex mode, and NRZ encoding. The router serial interfaces are configured as DCE devices.

Figure 7-7 SDLC STUN topology.



The following commands configure SDLC STUN for Router A:

```
stun peer-name 1.1.1.1
stun protocol-group 1 sdlc
!
interface serial 0
no ip address
encapsulation stun
sdlc address 04
stun route address 04 interface s1
stun route address ff interface s1
clockrate 19200
!
interface tokenring 0
ip address 1.1.4.1 255.255.255.0
!
interface serial 1
ip address 1.1.3.1 255.255.255.0
!
interface loopback 0
ip address 1.1.1.1 255.255.255.0
!
router igrp 1
network 1.0.0.0
```

The following commands configure SDLC STUN for Router B:

```
stun peer-name 1.1.2.1
stun protocol-group 1 sdlc
!
interface serial 0
no ip address
encapsulation stun
sdlc address 04
stun route address 04 interface s1
stun route address ff interface s1
clockrate 19200
!
interface tokenring 0
ip address 1.1.5.1 255.255.255.0
!
interface serial 1
ip address 1.1.3.2 255.255.255.0
!
interface loopback 0
ip address 1.1.2.1 255.255.255.0
!
router igrp 1
network 1.0.0.0
```

SDLC-Transmission Group STUN

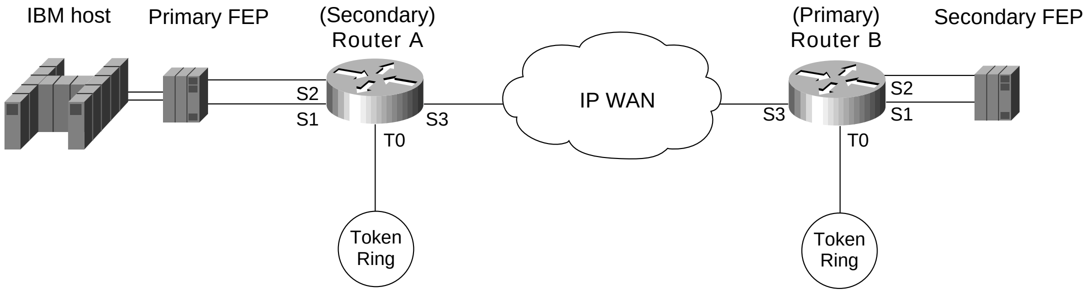
SDLC-Transmission Group (TG) STUN is a complex implementation that supports enhanced NCP features.When configuring STUN-TG, many network administrators also configure the routers to take advantage of the advanced features described in the “Advanced Router Features” section earlier in this chapter. Because these features increase memory and processor use, they should be used only when necessary to support the existing network or to relieve congestion. SDLC-TG STUN forces local acknowledgment. If you do not want to configure local acknowledgment, use the basic STUN or the SDLC STUN implementation.

The SDLC-TG implementation requires coordination of SDLC addresses between the router and the NCP configuration. To configure the router for SDLC-TG, the network administrator must know the relative position of the ADDRESS parameters in the NCP configuration. For details, see the “ADDRESS Parameter” section earlier in this chapter.

Configuring SDLC-TG STUN

Figure 7-8 illustrates a network that implements SDLC-TG STUN. The routers transmit data over an IP WAN. The FEPs are configured for DTE, full-duplex mode, and NRZ encoding. The serial interfaces on the routers are configured as DCE devices.

Figure 7-8 The SDLC-TG STUN topology.



To the primary FEP, Router A looks like a secondary FEP. To the secondary FEP, Router B looks like a primary FEP. The following commands configure SDLC-TG STUN for Router A:

```
stun peer-name 1.1.1.1
stun remote-peer-keepalive
stun protocol-group 1 sdlc-tg
!
interface tokenring 0
ip address 1.1.4.1 255.255.255.0
!
interface serial 1
mtu 4400
hold-queue 150 in
no ip address
encapsulation stun
stun group 1
stun sdlc-role secondary
sdlc n1 35200
sdlc address 01 echo
stun route address 1 tcp 1.1.2.1 local-ack tcp-queue-max 120
clockrate 56000


interface serial 2
mtu 4400
hold-queue 150 in
no ip address
encapsulation stun
stun group 1
stun sdlc-role secondary
sdlc n1 35200
sdlc address 02 echo
stun route address 2 tcp 1.1.2.1 local-ack tcp-queue-max 120
clockrate 56000
!
interface serial 3
ip address 1.1.3.1
interface loopback 0
ip address 1.1.1.1 255.255.255.0
!
router igrp 1
network 1.0.0.0
```

The following commands configure SDLC-TG STUN for Router B:

```
stun peer-name 1.1.2.1
stun remote-peer-keepalive
stun protocol-group 1 sdlc-tg
!
interface tokenring 0
ip address 1.1.5.1 255.255.255.0
!
interface serial 1
mtu 4400
hold-queue 150 in
no ip address
encapsulation stun
stun group 1
stun sdlc-role primary
sdlc line-speed 56000
sdlc n1 35200
sdlc address 01 echo
stun route address 1 tcp 1.1.1.1 local-ack tcp-queue-max 120
clockrate 56000
!
interface serial 2
mtu 4400
hold-queue 150 in
no ip address
encapsulation stun
stun group 1
stun sdlc-role primary
sdlc line-speed 56000
sdlc n1 35200
sdlc address 02 echo
stun route address 2 tcp 1.1.1.1 local-ack tcp-queue-max 120
clockrate 56000
!
interface serial 3
ip address 1.1.3.2
!
interface loopback 0
ip address 1.1.2.1 255.255.255.0
!
router igrp 1
network 1.0.0.0
```

The **stun peer-name** global configuration command identifies this router as a peer to its peer group.

The **stun remote-peer-keepalive** global configuration command causes Router A and Router B to exchange keepalive messages on each idle line. (An idle line is a line over which no I-frames are flowing.) Keepalive messages allow a router to detect when its peer router is not longer available. A peer router might become unavailable if it goes down or if the line goes down. The routers do not send keepalive traffic to the FEPs.

Routers send keepalive messages over an idle line at a default interval of 30 seconds and waits three times that interval for a response. If the router does not receive a response, it closes the STUN session.

The **stun protocol-group** global configuration command establishes a protocol group that is part of an SNA transmission group. The **sdlc-tg** keyword can be used only when the **stun route address tcp** interface configuration command is used to configure local acknowledgment and TCP encapsulation. The SDLC broadcast address 0xFF is routed automatically for interfaces on which the **sdlc-tg** keyword is configured. The **stun protocol-group** global configuration command also alerts the router that it should support transmission group features, such as the following:

- Echo addressing
- Transmission group rerouting if a single link in a multilink transmission group goes down
- Remote NCP load
- Broadcast addressing

The **mtu** interface configuration command specifies a maximum transmission unit (MTU) of 4400 bytes, which is the highest recommended value, for the interface. The value of the NCP MAXDATA parameter should be no more than the MTU on the router interface. The recommended value of MAXDATA is 4096 bytes.

Note Depending on the version of NCP, the MAXDATA parameter may or may not take into account the number of bytes in the frame header (which, for example, includes the source and destination address of the frame), so the MTU on the router interface should be at least 100 bytes larger than the value of MAXDATA in the NCP configuration.

The **hold-queue** interface configuration command increases the size of the input hold queue from 75 packets (the default) to 150 packets. The specified value should be greater than the depth of the TCP output queue (as specified by the **tcp-queue-max** keyword of the **stun route address tcp** interface configuration command). Increasing the size of the input hold queue allows flow control to activate when the TCP output queue reaches a threshold of 90 percent, which occurs before the input interface throttling mechanism can activate.

The **stun sdlc-role primary** interface configuration command is used when the router is connected to a secondary FEP. The **stun sdlc-role secondary** interface configuration command is used when the router is connected to a primary FEP.

On the primary router, the **sdlc line-speed** interface configuration command adjusts the SDLC poll timer based on the line speed. The line speed argument should be equal to the speed of the line connected to the interface, regardless of whether the interface is configured as a DCE or a DTE.

The **sdlc n1** interface configuration command specifies the maximum size (in bits) of an incoming frame on the SDLC link and is required when the MTU is not 1500 bytes (the default). The **sdlc n1** command must be eight times larger than the value specified by the **stun** command.

The **sdlc address** interface configuration command specifies an SDLC address. The specified address must be the same as the relative line number at which the ADDRESS parameter is specified in the NCP configuration of the FEP to which the router is connected. (For more information, see the “ADDRESS Parameter” section earlier in this chapter.) The **echo** keyword causes the router to treat nonecho (for example, 0x01) and echo (for example, 0x81) SDLC addresses as the same address. The **sdlc address** interface configuration command is valid only for interfaces on which the **stun protocol-group** command with the **sdlc-tg** keyword is configured. Only one **sdlc address** interface configuration command with **echo** keyword is required per interface.

The **stun route address tcp** interface configuration command specifies TCP encapsulation. The value of *address* specifies the SDLC address, which must be specified with the echo bit turned off. The **local-ack** keyword causes the router to perform local acknowledgment and is required when the **sdlc-tg** keyword appears with the **stun protocol-group** command. The **tcp-queue-max** keyword sets the maximum size of the TCP output queue for a serial line. The default is 100 packets. The recommended minimum is 70, and the recommended maximum is 500. The **clockrate** interface configuration command specifies the clocking speed when the serial interface is in DCE mode.

Summary

This case study presents three types of STUN implementations in SNA environments: basic STUN, SDLC STUN, and SDLC-TG STUN. Although basic STUN is the easiest to configure because it does not require the configuration of line addresses on the router, it does not support local acknowledgment. Compared to basic STUN, the SDLC STUN implementation is the most flexible because it supports, but does not require, local acknowledgment. However, the use of SDLC STUN is limited because it does not support transmission groups. SDLC-TG STUN is not as flexible as SDLC STUN because it enforces local acknowledgment. At the same time, SDLC-TG STUN is the only STUN implementation that supports transmission groups.

CHAPTER 8

Using ISDN Effectively in Multiprotocol Networks

As telephone companies make Integrated Services Digital Network (ISDN) services available, ISDN is becoming an increasingly popular way of connecting remote sites. This case study covers the following ISDN scenarios:

- *Configuring DDR over ISDN*—This telecommuting scenario describes the configuration of home sites that use ISDN to connect to a central company network and shows you how to use calling line identification numbers to prevent unauthorized access to the central network.
- *Configuring Snapshot Routing over ISDN*—Snapshot routing provides cost-effective access to a central company network from branch or home offices. Snapshot routing is used to upgrade the telecommuting network and control routing updates in Novell IPX networks.
- *Configuring AppleTalk over ISDN*—This scenario shows you how to control AppleTalk packets that might otherwise trigger unnecessary ISDN connections.

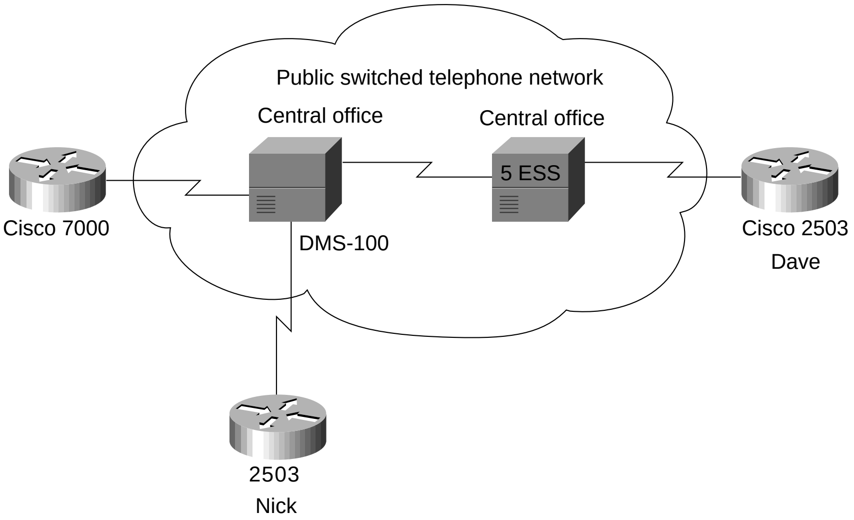
Configuring DDR over ISDN

In the United States, many companies today regard telecommuting as a way to solve space problems, conform to the Clean Air Act, and make employees more productive. In Europe, companies are looking for solutions that allow central offices to connect to remote sites. In the past, analog modems provided the necessary connectivity over serial lines, but they are not fast enough for LAN-to-LAN connections or for remote use of graphical programs, such as computer-aided design (CAD) tools. ISDN provides the needed additional bandwidth without requiring a leased line.

An ISDN Basic Rate Interface (BRI) provides two 64-kilobits-per-second (Kbps) B channels for voice or data and one 16-Kbps D channel for signaling. Voice and data information is carried over the B channels digitally. In the United States, an ISDN Primary Rate Interface (PRI) provides 23 64-Kbps B channels for voice and data over a T1 connection, and one 64-Kbps D channel for signaling. In Europe, a PRI provides 30 B channels for voice and data and one D channel for signaling over an E1 connection.

Figure 8-1 shows the network that will be discussed in this case study. The ISDN network uses multiple central office ISDN switches.

Figure 8-1 ISDN network example.



In this case study, the remote sites (homes) use Cisco 2503 routers, which provide one BRI, an Ethernet interface, and two high-speed serial interfaces. At the central company site, a Cisco 7000 series router equipped with a channelized T1 card answers the calls. The channelized T1 card provides a PRI.

Currently in many parts of the United States, telephone companies have not deployed Signaling System 7, which means that calls between certain central offices must be placed at 56 Kbps. This restriction does not apply to all parts of the United States or to other countries, but it does apply to some of the sample ISDN networks described in this chapter.

Native ISDN Interfaces

If you are using an external ISDN terminal adapter, also known as an *ISDN modem*, you can use the configuration examples provided in Chapter 2, "Dial-on-Demand Routing." Although an ISDN modem provides ISDN connectivity and allows you to use existing serial interfaces, it is not always the optimal solution because of the investment in an external unit and in additional cabling. Also, using V.25bis does not give the router full access to certain information that is available in an ISDN network, such as the speed of the call or the number of the calling party.

The native ISDN interface on the Cisco 2503 router allows the router to be directly connected to an ISDN NT1 device. In many countries, the NT1 is provided by the telephone company. In the United States, however, the NT1 is customer-owned equipment. By directly connecting to the ISDN network, the router has more direct control over ISDN parameters and has access to ISDN information.

Configuring an ISDN Interface

Configuring a native ISDN interface is similar to configuring a serial interface using DDR routing as described in Chapter 2, "Dial-on-Demand Routing." There are two major differences:

- The **dialer in-band** interface configuration command is not required with ISDN. PRI and BRI interfaces are assumed by the router to be a DDR interface.

- The individual B channels cannot be configured separately. The B channels of a BRI appear to be a dialer rotary group with two members. In the United States, the B channels of a PRI appear to be a dialer rotary group with 23 members, and in Europe, the B channels of a PRI appear to be a dialer rotary group with 30 members. Because the PRI or BRI is a dialer rotary group, all configuration commands associated with a PRI or BRI apply to all B channels.

The following sections describe the configurations of the central site and the home site routers. In this case study, both the central site and the home sites can place calls. The central site uses a Cisco 7000 router that connects to a NorTel DMS-100 central office ISDN switch. One remote site router (nick-isdn) connects to the same central office switch that the central site router uses. Connections from the other remote site router (dave-isdn) pass through two central office switches to reach the central site router.

Central Site

Two remote site users, Dave and Nick, dial from their homes into the central site router that is configured as follows. Part of the configuration of the central site router is specific to the DMS-100 switch, whereas other commands apply to any type of ISDN central office switch.

```
hostname central-isdn
!
username dave-isdn password 7 130318111D
username nick-isdn password 7 08274D02A02
isdn switch-type primary-dms100
!
interface ethernet 0
ip address 11.108.40.53 255.255.255.0
no mop enabled
!
controller t1 1/0
framing esf
linecode b8zs
pri-group timeslots 2-6
!
interface serial 1/0:23
ip address 11.108.90.53 255.255.255.0
encapsulation ppp
dialer idle-timeout 300
dialer map ip 11.108.90.1 name dave-isdn speed 56 914085553680
dialer map ip 11.108.90.7 name nick-isdn 8376
dialer-group 1
ppp authentication chap
!
router igrp 10
network 11.108.0.0
redistribute static
!
! route to nick-isdn
ip route 11.108.137.0 255.255.255.0 11.108.90.7
! route to dave-isdn
ip route 11.108.147.0 255.255.255.0 11.108.90.1
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!NTP
access-list 101 deny udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 123
!SNMP
access-list 101 deny udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 161
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
dialer-list 1 list 101
```

The configuration begins by establishing the host name of the router. The **username** global configuration commands establish the names of the routers that are allowed to dial up this router. The names correspond to the host names of Dave's router and Nick's router. The **isdn switch-type** command global configuration command specifies that the central site router connects to a NorTel DMS-100 switch. The host name, usernames, and ISDN switch type vary from router to router.

Controller Configuration

The **controller** global configuration command uses **T1** to specify a T1 controller interface. The "1" indicates that the controller card is located in backplane slot number 1. The "0" indicates port 0.

The **framing** controller configuration command selects the frame type for the T1 data line. In this case, the **framing** command uses the **esf** keyword to indicate the extended super frame (ESF) frame type. The service provider determines which framing type, either sf, esf, or crc4, is required for your T1/E1 circuit.

The **linecode** controller configuration command defines the line-code type for the T1 data line. In this case, the **linecode** command uses the **b8zs** keyword to indicate that the line-code type is bipolar 8 zero substitution (B8ZS). The service provider determines which line-code type, either alternate mark inversion (AMI) or B8ZS, is required for your T1/E1 circuit.

The **pri-group** controller configuration command specifies an ISDN PRI on a channelized T1 card in a Cisco 7000 series router. The **timeslots** keyword establishes the B channels. In this example, only five B channels (channels 2 through 6) are in use on this controller.

Interface Configuration

The **ip address** interface configuration command establishes the IP address of the interface, and the **encapsulation ppp** command establishes the Point-to-Point protocol (PPP) as the encapsulation method. PPP supports Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) as authentication mechanisms for identifying the caller and providing a level of security. The **dialer idle-timeout** interface configuration command sets the idle timeout to five minutes.

The **dialer map** interface configuration commands establish the remote sites that the router can call. Because Dave's router connects to a central office switch that does not use Signaling System 7, the **dialer map** command for calling Dave's router uses the **speed** keyword, which is valid for native ISDN interfaces only. The native ISDN interface on the Cisco 2503 operates at either 64 or 56 Kbps. If the calling party and the called party use the same ISDN switch, they can communicate at 64 Kbps. Otherwise, they must communicate at 56 Kbps.

Because Nick's ISDN line connects to the same central office as the line that the central site router uses, the telephone number in the **dialer map** command for connecting to Nick's router does not have to include the three-digit prefix. Note that because the central site router uses lines that are part of a Centrex, the outgoing telephone numbers start with 9 if they are not four-digit numbers.

The **dialer-group** interface configuration command associates the BRI with dialer access group 1. The **ppp authentication chap** interface configuration command enables CHAP authentication.

Routing Configuration

In the routing section of the configuration, the **router igrp** global configuration command enables the Interior Gateway Routing Protocol (IGRP) and sets the autonomous system number to 10. The **network** router configuration command assigns the network number. The **redistribute** router configuration command sends the static route information (defined with the **ip route** global configuration commands) to other routers in the same IGRP area. Without this command, other routers connected to the central site would not have routes to the remote routers.

DDR tends to use static routes extensively because routing updates are not received when the dial-up connection is not active. The first two **ip route** commands create the static routes that define the subnets that Dave and Nick use.

Note The IGRP commands are the same on all central site routers, except that the static routes correspond to the home sites calling into each central site router.

Access List Configuration

DDR uses access lists to determine whether a packet is *interesting* or *uninteresting*. Interesting packets cause a call to be placed if a call is not active or cause a call that has already been placed to be maintained as active. The first extended **access-list** global configuration command states that IGRP updates are uninteresting. The second extended **access-list** command states that Network Time Protocol (NTP) packets are uninteresting. The third extended **access-list** command specifies that Simple Network Management Protocol (SNMP) packets are uninteresting, and the final extended **access-list** command states that all other IP packets are interesting. The **dialer-list list** global configuration command assigns the set of access lists to dialer access group 1.

Home Site

The configurations of the home site routers are similar, but Nick’s configuration is simpler because his router connects to the same central office switch as the central site router.

Nick

The configuration for the router at Nick's home is as follows:

```
hostname nick-isdn
!
username central-isdn password 7 050D130C2A5
isdn switch-type basic-dms100
!
interface ethernet 0
ip address 11.108.137.1 255.255.255.0
no mop enabled
!
interface bri 0
ip address 11.108.90.7 255.255.255.0
encapsulation ppp
no ip route-cache
isdn spid1 415555837601 5558376
isdn spid2 415555837802 5558378
dialer idle-timeout 300
dialer map ip 11.108.90.53 name central-isdn 8362
dialer map ip 11.108.90.53 name central-isdn 8370
dialer-group 1
ppp authentication chap
!
ip route 11.108.0.0 255.255.0.0 11.108.90.53
!
access-list 101 deny udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 177
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
dialer-list 1 list 101
```

As with the central site router, the **isdn switch-type** global configuration command specifies that the switch is an NT DMS-100 switch. Because Nick's router connects to the DMS-100, SPIDs are required for the BRI. PPP and CHAP are configured, along with a **username** command for the central site router. The configuration for Nick's router differs from that of the central site with regard to the **dialer map** commands and the routing section. Two **dialer map** commands point to the same next-hop address. If the attempt to call the first number fails, the second number will be used to connect to the next-hop address.

The **isdn spid1** and **isdn spid2** interface configuration commands represent service profile identifiers (SPIDs). SPIDs are used when a BRI connects to a NorTel DMS-100 switch or a National ISDN-1 switch. SPIDs are assigned by the service provider to associate a SPID number with a telephone number. Other switch types do not require SPIDs. Your service provider can tell you if SPIDs are required for your switch. In this example, SPID 1 identifies 415 as the area code, 555 as the exchange, 8376 as the station ID, and 01 as the terminal identifier. The SPID format required by your service provider may differ from the examples shown in this case study.

Dave

The configuration for Dave's router is similar to the configuration for Nick's router, except that Dave's router is not in the same Centrex as the central company site. The configuration for Dave's router is as follows:

```
hostname dave-isdn
!
username central-isdn password 7 08274341
isdn switch-type basic-5ess
!
interface ethernet 0
ip address 11.108.147.1 255.255.255.0
no mop enabled
!
interface bri 0
ip address 11.108.90.1 255.255.255.0
encapsulation ppp
no ip route-cache
bandwidth 56
dialer map ip 11.108.90.53 name central-isdn speed 56 14155558370
dialer-group 1
ppp authentication chap
!
ip route 11.108.0.0 255.255.0.0 11.108.90.53
!
dialer-list 1 list 101
```

Dave's configuration is different from Nick's configuration because Dave's router connects to an AT&T 5ESS central office ISDN switch that does not run Signaling System 7. The **isdn switch-type** global configuration command specifies a basic rate AT&T switch, which does not require Dave's router configuration to use the **isdn spid1** and **isdn spid2** interface configuration commands that the DMS-100 switch requires. The **bandwidth** interface configuration command tells routing protocols that the line operates at 56 Kbps. The **dialer map** interface configuration command uses the **speed** keyword so that when Dave's router dials up the central site router, it sets the line speed to 56 Kbps. This setting is necessary when the connection traverses a switch that does not run Signaling System 7.

Configuring Calling Line Identification Numbers

Because Nick is in the same Centrex as the central company routers, the central router can use the Calling Line Identification (CLID) number received from the ISDN switch to identify Nick. With CLID, the configuration for Nick does not require CHAP or PAP; however, Nick needs to modify his configuration to include CLID. Nick's new configuration and a sample of the central site changed configuration are shown in the following sections.

Note CLID is not available in all parts of the United States and other countries. Some countries do not require Centrex for CLID.

Central Site

Here is the central site PRI interface configuration modified for CLID:

```
controller t1 1/0
framing esf
linecode b8zs
pri-group timeslots 2-6
!
interface serial 1/0:23
ip address 11.108.90.53 255.255.255.0
dialer idle-timeout 300
dialer map ip 11.108.90.7 name 5558376 8376
dialer-group 1
```

The **name** keyword in the **dialer map** interface configuration command specifies the actual string that calling line identification returns. This string differs from the number called: the number called is a four-digit Centrex number, and the number returned is the full seven digits.

Home Site

As with the central site, the major difference in Nick's configuration is the use of the **name** keyword with the **dialer map** command that specifies the actual number being returned as the calling line number.

```
interface bri 0
ip address 11.108.90.7 255.255.255.0
no ip route-cache
isdn spid1 415555837601 5558376
isdn spid2 415555837802 5558378
dialer idle-timeout 300
dialer map ip 11.108.90.53 name 5558362 8362
dialer map ip 11.108.90.53 name 5558370 8370
dialer-group 1
```

Note If the **debug isdn-q931** EXEC command is enabled, the decode for an incoming call setup can be seen and the CLID number will be shown.

Configuring Callback

Because Dave is located several miles from the central office, calls to the central office router are metered and billed to Dave's telephone number. The callback feature (introduced in Cisco IOS 11.0) allows Dave's router to place a call to the central site router requesting that the central site router call Dave's router. Then the central site router disconnects the call and places a return call to Dave's router.With callback configured, Dave's telephone bill is reduced because actual data transfers occur when the central office router calls back. The following commands configure callback on Dave's router:

```
interface bri 0
ppp callback request
dialer hold-queue 100 timeout 20
```

The **ppp callback** interface configuration command with the **request** keyword specifies that when the interface places a call, it is to request callback. The **dialer hold-queue** interface configuration command specifies that up to 100 packets can be held in a queue until the central site router returns the call. If the central site router does not return the call within 20 seconds plus the length of the enable timeout configured on the central site router, the packets are dropped. The following commands configure callback on the central office router:

```
map-class dialer class1
dialer callback-server username
interface serial 1/0:23
dialer map ip 11.108.90.1 name dave-isdn speed 56 class class1 914085553680
ppp callback accept
dialer callback-secure
dialer enable-timeout 1
dialer hold-queue
```

The **map-class** global configuration command establishes a quality of service (QoS) parameter that is to be associated with a static map. The **dialer** keyword specifies that the map is a dialer map. The **class1** parameter is a user-defined value that creates a map class to which subsequent encapsulation specific commands apply.

The **dialer map** interface configuration command has been modified to include the **class** keyword and the name of the class, as specified in the **map-class** command. The **name** keyword is required so that, when Dave's router dials in, the interface can locate this dialer map statement and obtain the dial string for calling back Dave's router.

The **ppp callback** interface configuration command with the **accept** keyword allows the interface to accept and honor callback requests that come into the interface. (Callback depends on PPP authentication, using PAP or CHAP.)

The **dialer callback-server** map class configuration command allows the interface to return calls when callback is successfully negotiated. The **username** keyword specifies that the interface is to locate the dial string for making the return call by looking up the authenticated host name in a **dialer map** command.

The **dialer callback-secure** interface configuration command specifies that the router is to disconnect the initial call, and call back only if it has a **dialer map** command with a defined class for the remote router. If the **dialer callback-secure** command is not present, the central router will not drop the connection if it does not have a **dialer map** command with a defined class. The **dialer enable-timeout** interface configuration command specifies that the interface is to wait one second after disconnecting the initial call before making the return call.

Configuring Snapshot Routing over ISDN

Snapshot routing is an easy way to reduce connection time in ISDN networks by suppressing the transfer of routing updates for a configurable period of time. Snapshot routing is best suited for networks whose data-transfer connections typically last longer than five minutes and that are running the following distance-vector protocols:

- Routing Information Protocol (RIP) and Integrated Gateway Routing Protocol (IGRP) for IP
- Routing Table Maintenance Protocol (RTMP) for AppleTalk
- Routing Information Protocol (RIP) and Service Advertisement Protocol (SAP) for Novell Internet Packet Exchange (IPX)
- Routing Table Protocol (RTP) for Banyan VINES

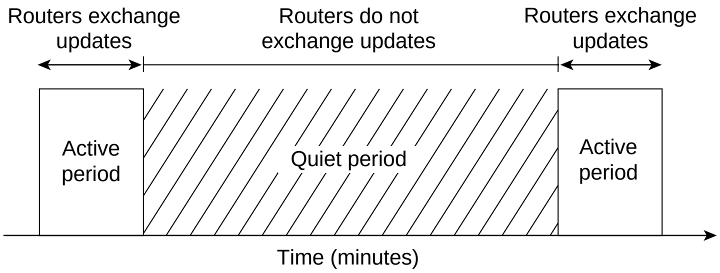
The goal of snapshot routing is to allow routing protocols to exchange updates as they normally would. Because Enhanced IGRP and link-state routing protocols, such as Novell Link Services Protocol (NLSP), Open Shortest Path First (OSPF), and Intermediate System-to-Intermediate System (IS-IS) depend on the frequent sending of hello messages to neighboring routers in order to discover and maintain routes, they are incompatible with snapshot routing.

Note This case study applies snapshot routing to an ISDN network, but other similar media, such as dedicated leased lines, can benefit from the reduction of periodic updates that snapshot routing provides.

Before snapshot routing became available in Cisco Internetwork Operating System (IOS) Software Release 10.2, ISDN interfaces were configured using static routes. Static routes, such as the routes defined by the **ip route** commands in the "Central Site" section earlier in this chapter, prevent bandwidth from being consumed by routing updates, but they are difficult to maintain as the network grows.

Snapshot routing supports dynamic routes by allowing routing updates to occur during an active period and reduces connection cost by suppressing routing updates during a quiet period, which can be up to 65 days long. During the quiet period, the routing tables on the routers at both ends of a link are frozen. Figure 8-2 shows the relationship of active and quiet periods over time.

Figure 8-2 Active periods and frozen periods over time.



During the active period, the routers at each end of the connection exchange the routing updates that are normal for their configured routing protocols. They continue to exchange routing updates until the active period ends. When the active period ends, each router freezes its routing tables, stops sending routing updates, and enters the quiet period. Each router remains in the quiet period until a configurable timer expires, at which time one of the routers initiates a connection to send and receive routing updates.

To ensure that routing tables are updated, the active period must be long enough for several routing updates to come through the link. An active period that is too short might allow only one routing update to cross the link. If that update is lost due to noise on the line, the router on the other end would age out a valid route or would not learn about a new valid route. To make sure that updates occur, the active period must be at least five minutes long (that is, three times longer than the routing protocols' update interval). Because the routing protocols update their routing tables during the active period as they normally would, there is no need to adjust any routing protocol timers.

If the line is not available when the router transitions from the quiet period to the active period, it enters a retry period. During the retry period, the router continually attempts to connect until it enters an active period, as shown in Figure 8-3.

Figure 8-3 The router continually attempts to connect during the retry period.

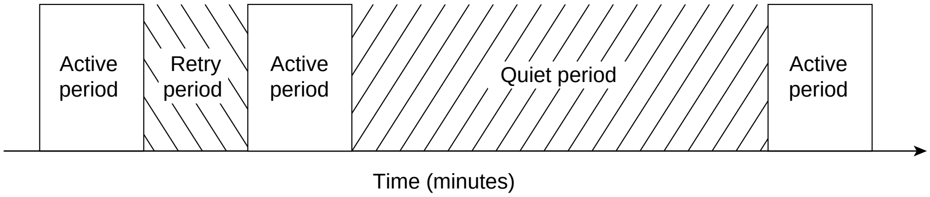


Table 8-1 shows the minimum and maximum lengths of each period.

Table 8-1 Snapshot Routing Periods

Period	Configurable	Minimum Length	Maximum Length
Active	Yes	5 minutes	100 minutes
Quiet	Yes	5 minutes	65 days
Retry	No	8 minutes	8 minutes

By default, snapshot routing allows routing updates to be exchanged over connections that are established to transfer user data. This means that, if necessary, snapshot routing forces the connection to last as long as the active period. If you do not want the routers to exchange updates during connections that are established to transfer user data, use the **suppress-statechange-updates** keyword.

Upgrading the Telecommuting Network

Snapshot routing is well-suited to the hub-and-spoke topology of the telecommuting network described in the "Configuring DDR over ISDN" section at the beginning of this chapter. Snapshot routing is designed for a client-server relationship. The client routers, such as the home sites, determine the frequency at which the routers exchange updates by setting the length of the quiet period, and the server router accepts incoming snapshot connections from several client routers.

Note Snapshot routing is not recommended for meshed topologies. In meshed topologies, configuring static routes is more efficient than configuring snapshot routing.

Central Site Modified for Snapshot Routing

The following is the configuration of the central site router after modification for snapshot routing:

```
hostname central-isdn
!
username dave-isdn password 7 130318111D
username nick-isdn password 7 08274D02A02
isdn switch-type primary-dms100
!
interface ethernet 0
ip address 11.108.40.53 255.255.255.0
no mop enabled
!
controller t1 1/0
framing esf
linecode b8zs
pri-group timeslots 2-6
ip address 11.108.90.53 255.255.255.0
encapsulation ppp
dialer idle-timeout 300
dialer map ip 11.108.90.1 name dave-isdn speed 56 914085553680
dialer map ip 11.108.90.7 name nick-isdn 8376
dialer-group 1
isdn spid1 415555836201 5558362
isdn spid2 415555837002 5558370
snapshot server 5
ppp authentication chap
!
```

```
router igrp 10
network 11.108.0.0
redistribute static
!
! route to nick-isdn
ip route 11.108.137.0 255.255.255.0 11.108.90.7
! route to dave-isdn
ip route 11.108.147.0 255.255.255.0 11.108.90.1
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!NTP
access-list 101 deny udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 123
!SNMP
access-list 101 deny udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 161
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
dialer-list 1 list 101
```

The **ip route** global configuration commands that configured static routes for the home sites have been removed from the configuration. The **snapshot server** interface configuration command enables snapshot routing. The “5” sets the length of the active period to five minutes.

Note Snapshot routing must be configured on rotary interfaces, which are established by the **dialer rotary-group** interface configuration command. ISDN interfaces are rotary interfaces by definition, so you do not need to use the **dialer rotary-group** command in ISDN configurations.

Home Site Modified for Snapshot Routing

The following is the configuration of Dave’s home site router after modification for snapshot routing:

```
hostname dave-isdn
!
username central-isdn password 7 08274341
isdn switch-type basic-5ess
!
interface ethernet 0
ip address 11.108.147.1 255.255.255.0
no mop enabled
!
interface bri 0
ip address 11.108.90.1 255.255.255.0
encapsulation ppp
no ip route-cache
bandwidth 56
dialer map snapshot 1 name central-isdn 14155558370
dialer map ip 11.108.90.53 name central-isdn speed 56 14155558370
dialer-group 1
snapshot client 5 43200 suppress-statechange-updates dialer
ppp authentication chap
!
dialer-list 1 list 101
```

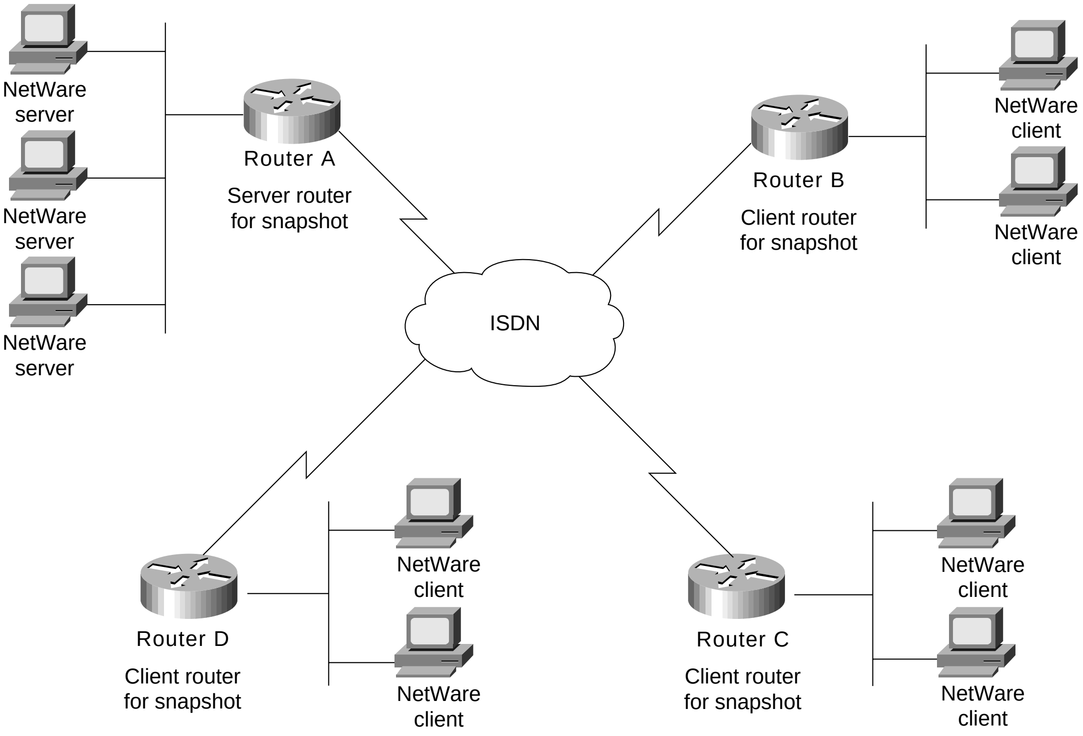
The **ip route** commands that configured static routes for the home sites have been removed from the configuration. The **dialer map snapshot** interface configuration command establishes a map (whose sequence number is 1) that the router uses to connect to the central site router for the exchange of routing updates. The **name** keyword specifies the name of the remote router that is associated with the dial string. Because the **ppp authentication** interface configuration command enables CHAP authentication, when this router dials the central router, it receives the host name of the central router and compares it with the name specified by the **name** keyword.

The **snapshot client** interface configuration command sets the length of the active period to five minutes (a value that must match the value set in the snapshot server's configuration) and sets the length of the quiet period to 43,200 seconds (12 hours). The **suppress-statechange-updates** keyword prevents the routers from exchanging updates during connections that are established to transfer user data. The **dialer** keyword allows the client router to dial up the server router in the absence of regular traffic and is required when you use the **suppress-statechange-update** keyword.

Snapshot and Novell IPX Networks

This section describes a Novell IPX network for which snapshot routing has been configured. Client routers at branch offices use DDR to connect to a central router over ISDN. At the central office, NetWare servers use the Novell IPX protocol to provide services to NetWare clients on each branch office network. Some client-to-server connections are required during a limited period of the day. Figure 8-4 illustrates the network.

Figure 8-4 Topology of the Novell IPX network.



In this topology, the client routers are responsible for updating their routing tables by connecting to the server router when the quiet period expires. The client routers also retrieve update information if a reload occurs.

Note Snapshot routing works with Novell 3.*x* and 4.*x* networks. However, Novell 4.*x* includes a time synchronization protocol that causes Novell 4.*x* time servers to send an update every 10 minutes. To prevent the time server from generating update packets that would cause unwanted connections, you should load a NetWare Loadable Module (NLM) named TIMESYNC.NLM that allows you to increase the update interval for these packets to several days. A similar problem is caused by Novell's efforts to synchronize NDS replicas. NetWare 4.1 includes two NLMs, DSFILTER.NLM and PINGFILT.NLM, that work together to control NDS synchronization updates. You should use these two modules to make sure that NDS synchronization traffic is sent to specified servers only at the specified times.

Server Router Configuration

The following is the complete configuration for the server router:

```
hostname RouterA
!
username RouterB password 7 120DOA031D
username RouterC password 7 111D161118
username RouterD password 7 43E7528384
isdn switch-type vn3
!
ipx routing

interface Ethernet 0
ip address 192.104.155.99 255.255.255.0
ipx network 300
!
interface bri 0
ip address 1.0.0.1 255.0.0.0
encapsulation ppp
ipx network 10
no ipx route-cache
ipx update-time 20
ipx watchdog-spoof
dialer idle-timeout 60
dialer wait-for-carrier-time 12
dialer map ipx 10.0000.0000.0002 name RouterB broadcast 041389082
dialer map ipx 10.0000.0000.0003 name RouterC broadcast 041389081
dialer map ipx 10.0000.0000.0004 name RouterD broadcast 041389083
!
dialer-group 1
snapshot server 10
ppp authentication chap
!
access-list 901 deny 0 FFFFFFF 0 FFFFFFFF 457
access-list 901 deny 1 10.0000.0000.0001 0 10.ffff.ffff.ffff 453
access-list 901 deny 4 10.0000.0000.0001 0 l0.ffff.ffff.ffff 452
access-list 901 deny 4 FFFFFFFF 0 FFFFFFFF 456
access-list 901 permit -1
!
dialer-list 1 list 901
```

The configuration begins with the host name used for CHAP authentication. The usernames correspond to the host names of Router B, Router C, and Router D. The **isdn switch-type** global configuration command specifies that the router connects to a French VN3 ISDN BRI switch.

Interface Configuration

The **dialer idle-timeout** interface configuration command specifies 60 seconds as the amount of idle time that must elapse before the router disconnects the line. The **dialer wait-for-carrier-time** interface configuration command sets the wait-for-carrier time to 60 seconds.

The first **dialer map** interface configuration command sets the next-hop address of Router B to 10.0000.0000.0002. When Router B dials up the server router (Router A), the server router uses the next hop address to transmit packets to Router B. The **broadcast** keyword sets 041389082 as the address to which IPX broadcasts are to be forwarded. The second and third **dialer map** commands set similar values for Router C and Router D.

The **snapshot server** interface configuration command sets the length of the active period to 10 minutes. The **ppp authentication** interface configuration command sets CHAP as the authentication protocol.

Access List Configuration

Access lists are used to determine whether an outgoing packet is interesting or uninteresting. Packets that are not interesting are dropped, and packets that are interesting cause a call to be placed if a call is not active or cause a call that has already been placed to be maintained as active. The access lists defined by this configuration are extended Novell IPX access lists. The first **access-list** global configuration command defines any packets intended for the Novell serialization socket as uninteresting. The second **access-list** command defines RIP packets as uninteresting. The third **access-list** command defines SAP packets as uninteresting. The fourth **access-list** command defines Novell diagnostic packets generated by the Autodiscovery feature as uninteresting, and the final **access-list** command states that all other packets are interesting. The **dialer-list global configuration** command assigns access list 901 to dialer access group 1, which is associated with BRI 0 by the **dialer-group** interface configuration command.

Client Router Configuration

The configurations for the client routers are the same except for the commands that configure the router’s host name, the username that it uses when it dials up Router A, and the router’s network numbers. The following is the configuration for Router B:

```
hostname RouterB
!
username RouterA password 7 105A060D0A
ipx routing
isdn switch-type vn3
isdn tei first-call
!
interface ethernet 0
ip address 192.104.155.100 255.255.255.0
ipx network 301
!
interface bri 0
no ip address
encapsulation ppp
ipx network 10
no ipx route-cache
ipx update-time 20
ipx watchdog-spoof
dialer idle-timeout 60
dialer wait-for-carrier-time 12
dialer map snapshot 1 name RouterA 46148412
dialer map ipx 10.0000.0000.0001 name RouterA broadcast 46148412
dialer-group 1
```

```
snapshot client 10 86400 dialer
ppp authentication chap
!
access-list 901 deny 0 FFFFFFFF 0 FFFFFFFF 457
access-list 901 deny 1 10.0000.0000.0002 0 10.ffff.ffff.ffff 453
access-list 901 deny 4 10.0000.0000.0002 0 10.ffff.ffff.ffff 452
access-list 901 deny 4 FFFFFFFF 0 FFFFFFFF 456
access-list 901 permit 0
!
dialer-list 1 list 901
```

The configuration begins with the host name used for CHAP authentication. The usernames correspond to the host names of Router B, Router C, and Router D. The **isdn switch-type** global configuration command specifies that the router connects to a French VN3 ISDN BRI switch.

The **isdn tei** global configuration command uses the **first-call** keyword to specify that ISDN terminal *endpoint identifier* (TEI) negotiation is to occur when Router A places or receives its first ISDN call. (The default is for TEI negotiation to occur when the router is powered on.)

Interface Configuration

The **dialer wait-for-carrier** interface configuration command specifies 12 seconds as the number of seconds that the interface will wait for the carrier to come up when it places a call.

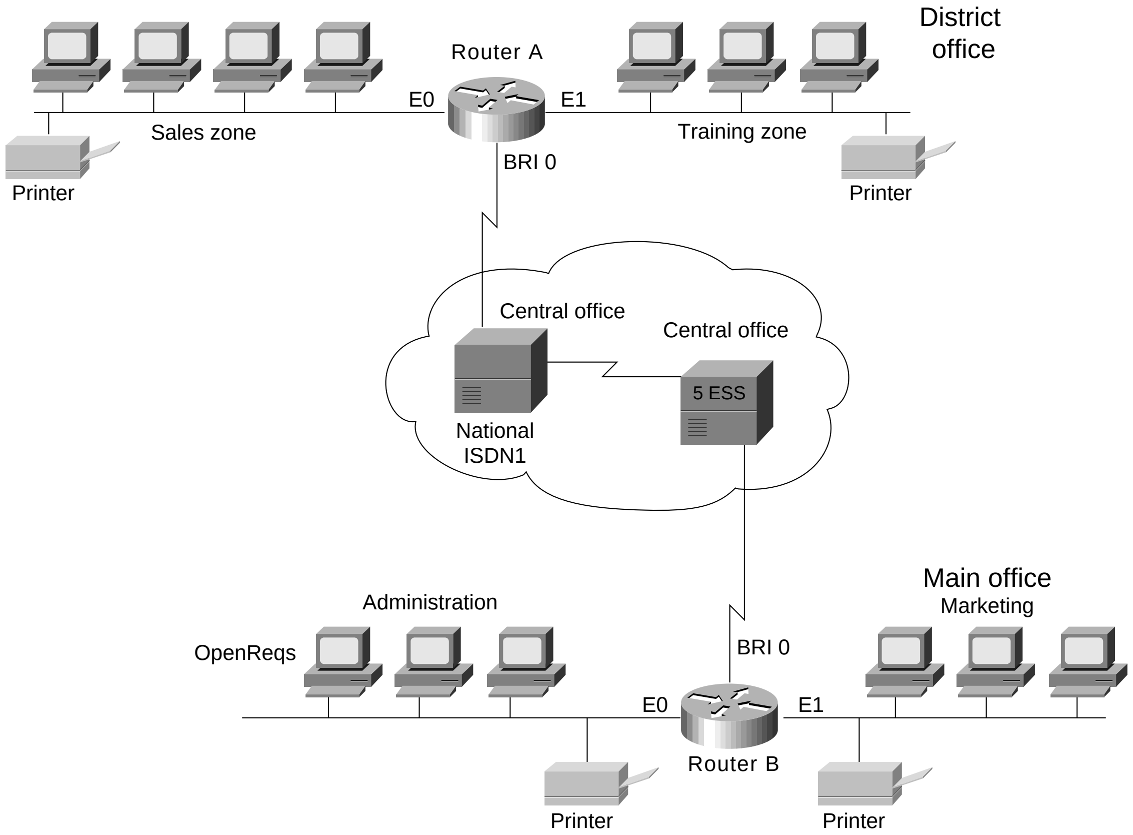
The **snapshot client** interface configuration command sets the length of the active period to 10 minutes (a value that must match the value set in the snapshot server's configuration) and sets the length of the quiet period to 86,400 seconds (24 hours). Because the **suppress-statechange-updates** keyword is not used, the routers can exchange updates during connections that are established to transfer user data. The **dialer** keyword allows the client router to dial up the server router in the absence of regular traffic.

Configuring AppleTalk over ISDN

To run AppleTalk over an ISDN network effectively, you need to prevent Name Binding Protocol (NBP) packets and RTMP updates from triggering unnecessary connections over ISDN connections.

Figure 8-5 shows a sample AppleTalk network that uses ISDN to connect two networks located in different cities. Users on the district office network occasionally need access to servers located on the main office network and vice versa. In this scenario, both routers dial up each other when user data from one part of the network needs to reach the other part of the network.

Figure 8-5 An AppleTalk network over ISDN.



Users of hosts connected to the main office network do not need to access the Training zone, so when configuring Router A, one goal is to prevent NBP packets generated by the Training zone from triggering an ISDN connection with the main office network. Another configuration goal for both routers is to prevent NBP packets generated by the printers on each network from triggering an ISDN connection.

To control the forwarding of NBP packets, use AppleTalk-style access lists. AppleTalk-style access lists allow you to control the flow of NBP packets based on the type of the entity that originated the packet, the name of the entity that originated the packet, and the zone of the entity that originated the packet.

Note The capability to control the forwarding of NBP packets was introduced in Cisco IOS Software Release 11.0.

Both routers also need to control RTMP packets. To control RTMP packets, configure static AppleTalk cable ranges and node numbers and use the **no appletalk send rtmps** command on the ISDN BRI or PRI interface that connects two AppleTalk networks.

Router A Configuration

As shown in Figure 8-5, Router A is located in the district office. The district office network consists of two zones: Sales and Training. On Router A, an AppleTalk-style access list is assigned to BRI 0 to prevent the forwarding of NBP packets that come from printers and NBP packets that come from the Training zone. If the router were to allow the forwarding of these packets, they would trigger an unnecessary ISDN connection to the main office network.

```
hostname RouterA
!
username RouterB password 7 125D063D2E
appletalk routing
appletalk static cable-range 20-20 to 15.43 zone Administration
appletalk static cable-range 25-25 to 15.43 zone Marketing
isdn switch-type basic-ni1
!
interface ethernet 0
appletalk cable-range 5-5 5.128
appletalk zone Sales
!
interface ethernet 1
appletalk cable-range 10-10 10.26
appletalk zone Service
!
interface bri 0
appletalk static cable-range 15-15 15.42
appletalk zone PhoneZone
no appletalk send-rtmps
encapsulation ppp
ppp authentication chap
dialer idle-timeout 240
bandwidth 56
dialer map appletalk 15.43 name RouterA speed 56 912065553240
dialer-group 1
isdn spid1 602555463101 5554631
!
access-list 601 deny nbp 1 type LaserWriter
access-list 601 deny nbp 2 zone Training
access-list 601 permit nbp 3 zone Sales
access-list 601 deny other-nbps
access-list 601 permit other-access
!
dialer-list 1 list 601
```

The **hostname** global configuration command establishes the host name of Router A. The **username** global configuration command establishes the name of the router that is allowed to dial up Router A. The name corresponds to the host name of Router B. The **password** keyword indicates that the **username** command specifies a password. The “7” indicates that the password is encrypted using a Cisco-defined encryption algorithm. The **appletalk routing** global configuration command enables AppleTalk routing.

The **appletalk static cable-range** global configuration commands create static AppleTalk routes to the zones in the main office network. Static AppleTalk routes are required because the **no appletalk send-rtmps** interface configuration command prevents the exchange of RTMP updates between the two networks. Without static routes, zones for the main office would not appear when users open the Chooser on hosts connected to the district office network. The **isdn switch-type** global configuration command specifies that Router A connects to a National ISDN-1 switch.

Interface Configuration

The **appletalk cable-range** interface configuration commands for each Ethernet interface establish the network number for the cable segment to which the interface connects and the node number of the interface. For each interface, the **appletalk zone** interface configuration command establishes the zone name for the network that is connected to the interface. None of the interface configurations specifies an AppleTalk routing protocol, so the interfaces use the default routing protocol, RTMP.

The **no appletalk send-rtmps** interface configuration command prevents Router A from sending RTMP updates out on interface BRI 0. To compensate for the lack of RTMP exchange, you must configure static AppleTalk routes (using the **appletalk static cable-range** global configuration command).

The **encapsulation ppp** interface configuration command specifies PPP encapsulation, and the **ppp authentication chap** command enables CHAP authentication. The **dialer idle-timeout** interface configuration command sets the idle timeout to 240 seconds (four minutes). The **bandwidth** interface configuration command tells routing protocols that the line operates at 56 Kbps.

The **dialer map** interface configuration command establishes the remote site that Router A is to call. In this case, the **dialer map** command establishes 15.43 as the next hop address. The **name** keyword specifies the name of the remote router that is associated with the dial string. The **speed** keyword specifies that Router A is to set the line’s rate to 56 Kbps, which is required when the connection traverses a switch that does not support Signaling System 7. The **dialer-group** interface configuration command associates the interface BRI 0 with dialer access group 1.

The **isdn spid1** interface configuration commands represent service profile identifiers (SPIDs) and are required by National ISDN-1 switches. Service providers assign SPIDs to associate a SPID number with a telephone number. Your service provider can tell you if SPIDs are required for your switch. In this example, SPID 1 identifies 602 as the area code, 555 as the exchange, 4631 as the station ID, and 01 as the terminal identifier.

Access List Configuration

The first **access-list nbp** global configuration command defines access list 601 and prevents the forwarding of NBP packets generated by any LaserWriter printer on the district office network. The second **access-list nbp** command prevents the forwarding of NBP packets generated by the Training zone. The third **access-list nbp** command allows the forwarding of NBP packets generated by the Sales zone.

The **access-list other-nbps** global configuration command prevents the forwarding of all other NBP packets that have not been explicitly permitted or denied by previous **access-list nbp** global configuration commands.

The **access-list other-access** global configuration command permits all other access checks that would otherwise be denied because they are not explicitly permitted by an **access-list** command. The **dialer-list** global configuration command assigns the access list 601 to dialer access group 1, which is associated with BRI 0.

Router B Configuration

As shown in Figure 8-5, Router B is located in the main office. The main office network consists of two zones: Marketing and Administration. With the exception of the OpenReqs server in the Administration zone, users of hosts connected to the district office network do not need to access servers located in the Administration zone. Like the district office network, each zone in the main office network has its own printer, so there is no need for Router B to forward NBP packets that the printers originate. The access list for Router B prevents NBP packets that come from printers and NBP packets that come from all servers in the Administration zone (except OpenReqs) from triggering an ISDN connection to the district office network.

```
hostname RouterB
!
username RouterA password 7 343E821D4A
appletalk routing
appletalk static cable-range 5-5 to 15.42 zone Sales
appletalk static cable-range 10-10 to 15.42 zone Training
isdn switch-type basic-5ess
!
interface ethernet 0
appletalk cable-range 20-20 20.5
appletalk zone Administration
!
interface ethernet 1
appletalk cable-range 25-25 25.36
appletalk zone Marketing
!
interface bri 0
appletalk static cable-range 15-15 15.43
appletalk zone PhoneZone
no appletalk send-rtmps
encapsulation ppp
ppp authentication chap
dialer idle-timeout 240
bandwidth 56
dialer map appletalk 15.42 name RouterB speed 56 917075553287
dialer-group 1
!
access-list 601 deny nbp 1 type LaserWriter
access-list 601 permit nbp 2 object OpenReqs
access-list 601 permit nbp 3 zone Marketing
access-list 601 deny other-nbps
access-list 601 permit other-access
dialer-list 1 list 601
```

The configuration for Router B is similar to the configuration for Router A, with the follwing differences:

- The **isdn switch-type** global configuration command specifies that Router B connects to an AT&T 5ESS central office ISDN switch. This type of switch does not use SPID numbers, so the **isdn spid1** command is not used.
- The first **access-list nbp** global configuration command defines access list 601 and prevents the forwarding of NBP packets generated by the LaserWriter printers connected to the main office network. The second **access-list nbp** command allows the forwarding of packets generated by the server OpenReqs. The third **access-list nbp** command allows the forwarding of packets generated by the Marketing zone.

Summary

When you configure ISDN, controlling packets that trigger unnecessary connections is a major concern. In the past, one way of controlling routing update packets was to configure static routes. Snapshot routing and NBP-packet filtering provide new ways to control routing updates. Snapshot routing allows you to configure the network so that routed protocols update their routing tables dynamically without triggering frequent and costly ISDN connections. Snapshot routing is ideally suited for relatively stable networks in which a single router is a central point through which routing updates flow.

CHAPTER 9

Using HSRP for Fault-Tolerant IP Routing

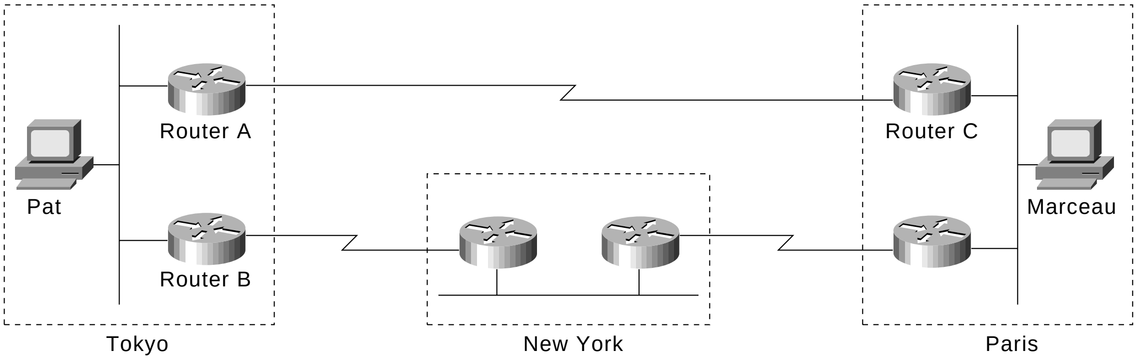
This case study examines Cisco's Hot Standby Routing Protocol (HSRP), which provides automatic router backup when you configure it on Cisco routers that run the Internet Protocol (IP) over Ethernet, Fiber Distributed Date Interface (FDDI), and Token Ring local-area networks (LANs). HSRP is compatible with Novell's Internetwork Packet Exchange (IPX), AppleTalk, and Banyan VINES, and it is compatible with DECnet and Xerox Network Systems (XNS) in certain configurations.

Note Banyan VINES serverless clients do not respond well to topology changes (regardless of whether HSRP is configured). This case study describes the effect of topology changes in networks that include Banyan VINES serverless clients.

For IP, HSRP allows one router to automatically assume the function of the second router if the second router fails. HSRP is particularly useful when the users on one subnet require continuous access to resources in the network.

Consider the network shown in Figure 9-1. Router A is responsible for handling packets between the Tokyo segment and the Paris segment, and Router B is responsible for handling packets between the Tokyo segment and the New York segment. If the connection between Routers A and C goes down or if either router becomes unavailable, fast converging routing protocols, such as the Enhanced Interior Gateway Routing Protocol (Enhanced IGRP) and Open Shortest Path First (OSPF) can respond within seconds so that Router B is prepared to transfer packets that would otherwise have gone through Router A.

Figure 9-1 A typical WAN.



However, in spite of fast convergence, if the connection between Router A and Router C goes down, or if either router becomes unavailable, the user Pat on the Tokyo segment might not be able to communicate with the user Marceau even after the routing protocol has converged. That's because IP hosts, such as Pat's workstation, usually do not participate in routing protocols. Instead, they are configured statically with the address of a single router, such as Router A. Until someone manually modifies the configuration of Pat's host to use the address of Router B instead of Router A, Pat cannot communicate with Marceau.

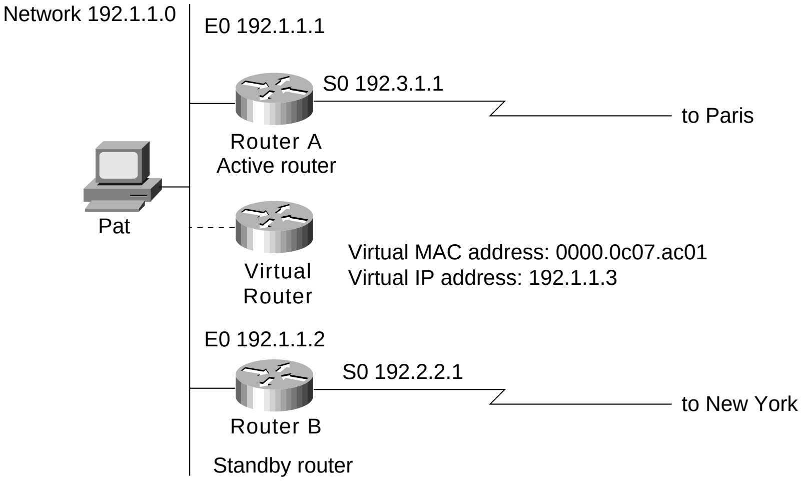
Some IP hosts use proxy Address Resolution Protocol (ARP) to select a router. If Pat's workstation were running proxy ARP, it would send an ARP request for the IP address of Marceau's workstation. Router A would reply on behalf of Marceau's workstation and would give to Pat's workstation its own media access control (MAC) address (instead of the IP address of Marceau's workstation). With proxy ARP, Pat's workstation behaves as if Marceau's workstation were connected to the same segment of the network as Pat's workstation. If Router A fails, Pat's workstation will continue to send packets destined for Marceau's workstation to the MAC address of Router A even though those packets have nowhere to go and are lost. Pat either waits for ARP to acquire the MAC address of Router B by sending another ARP request or reboots the workstation to force it to send an ARP request. In either case, for a significant period of time, Pat cannot communicate with Marceau—even though the routing protocol has converged, and Router B is prepared to transfer packets that would otherwise go through Router A.

Some IP hosts use the Routing Information Protocol (RIP) to discover routers. The drawback of using RIP is that it is slow to adapt to changes in the topology. If Pat's workstation is configured to use RIP, 3 to 10 minutes might elapse before RIP makes another router available.

Some newer IP hosts use the ICMP Router Discovery Protocol (IRDP) to find a new router when a route becomes unavailable. A host that runs IRDP listens for *hello* multicast messages from its configured router and uses an alternate router when it no longer receives those hello messages. If Pat's workstation were running IRDP, it would detect that Router A is no longer sending hello messages and would start sending its packets to Router B.

For IP hosts that do not support IRDP, Cisco's HSRP provides a way to keep communicating when a router becomes unavailable. HSRP allows two or more HSRP-configured routers to use the MAC address and IP network address of a virtual router. The virtual router does not physically exist; instead, it represents the common target for routers that are configured to provide backup to each other. Figure 9-2 shows the Tokyo segment of the WAN as it might be configured for HSRP. Each actual router is configured with the MAC address and the IP network address of the virtual router.

Figure 9-2 HSRP addressing on the Tokyo segment.



In Figure 9-2, the MAC address of the virtual router is 0000.0c07.ac01. When you configure HSRP, the router automatically selects one of the virtual MAC addresses from a range of addresses in the Cisco IOS software that is within the range of Cisco's MAC address block. Ethernet and FDDI LANs use one of the preassigned MAC addresses as a virtual MAC address. Token Ring LANs use a functional address as a virtual MAC address.

In Figure 9-2, instead of configuring the hosts on network 192.1.1.0 with the IP address of Router A, they are configured with the IP address of the virtual router as their default router. When Pat's workstation sends packets to Marceau's workstation on the Paris segment, it sends them to the MAC address of the virtual router.

In Figure 9-2, Router A is configured as the active router. It is configured with the IP address and MAC address of the virtual router and sends any packets addressed to the virtual router out interface 1 to the Paris segment. As the standby router, Router B is also configured with the IP address and MAC address of the virtual router. If for any reason Router A stops transferring packets, the routing protocol converges, and Router B assumes the duties of Router A and becomes the active router. That is, Router B now responds to the virtual IP address and the virtual MAC address. Pat's workstation continues to use the IP address of the virtual router to address packets destined for Marceau's workstation, which Router B receives and sends to the Paris segment via the New York segment. Until Router A resumes operation, HSRP allows Router B to provide uninterrupted service to the users on the Tokyo segment that need to communicate with users on the Paris segment. While it is the active router, Router B continues to perform its normal function: handling packets between the Tokyo segment and the New York segment.

HSRP also works when the hosts are configured for proxy ARP. When the active HSRP router receives an ARP request for a host that is not on the local LAN, the router replies with the MAC address of the virtual router. If the active router becomes unavailable or its connection to the remote LAN goes down, the router that becomes the active router receives packets addressed to the virtual router and transfers them accordingly.

Note You can configure HSRP on any Cisco router that is running Cisco Internetwork Operating System (Cisco IOS) Software Release 10.0 or later. If you configure HSRP for one Cisco router on a Token Ring LAN, all Cisco routers on that LAN must run Cisco IOS Software Release 10.0 or later. Cisco IOS Software Releases 10.2(9), 10.3(6), and 11.0(2) allow standby IP addresses to respond to ping requests. Cisco Software Release 11.0(3)(1) provides improved support for the use of secondary IP addresses with HSRP.

Understanding How HSRP Works

HSRP uses a priority scheme to determine which HSRP-configured router is to be the default active router. To configure a router as the active router, you assign it a priority that is higher than the priority of all the other HSRP-configured routers. The default priority is 100, so if you configure just one router to have a higher priority, that router will be the default active router.

HSRP works by the exchange of multicast messages that advertise priority among HSRP-configured routers. When the active router fails to send a hello message within a configurable period of time, the standby router with the highest priority becomes the active router. The transition of packet-forwarding functions between routers is completely transparent to all hosts on the network.

HSRP-configured routers exchange three types of multicast messages:

- *Hello*—The hello message conveys to other HSRP routers the router's HSRP priority and state information. By default, an HSRP router sends hello messages every three seconds.
- *Coup*—When a standby router assumes the function of the active router, it sends a coup message.

- *Resign*—A router that is the active router sends this message when it is about to shut down or when a router that has a higher priority sends a hello message.

At any time, HSRP-configured routers are in one of the following states:

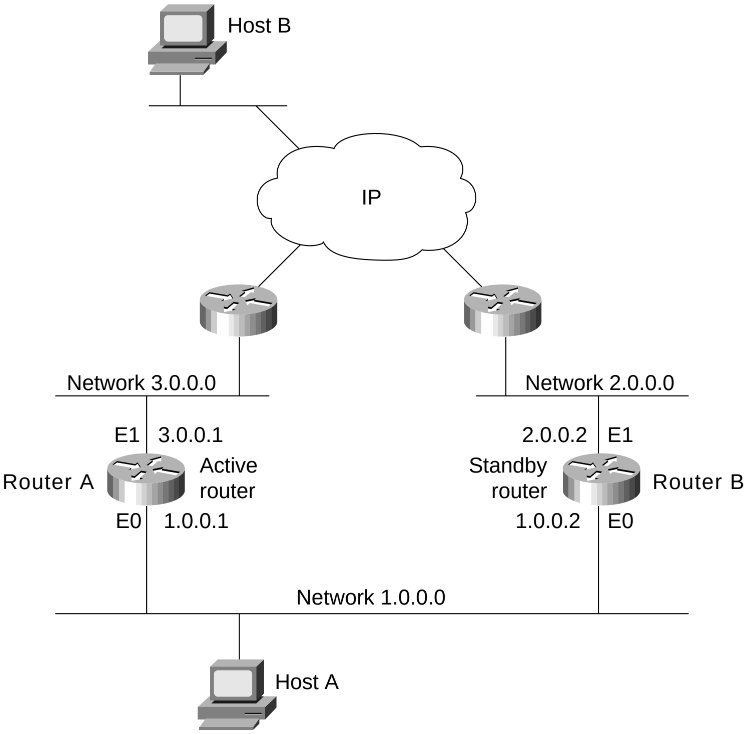
- *Active*—The router is performing packet-transfer functions.
- *Standby*—The router is prepared to assume packet-transfer functions if the active router fails.
- *Speaking and listening*—The router is sending and receiving hello messages.
- *Listening*—The router is receiving hello messages.

Note When configured on AGS, AGS+, and Cisco 7000 series routers, HSRP takes advantage of special hardware features that are not available on other Cisco routers. This means that HSRP operates in a slightly different way on these routers. For an example, see the "Using HSRP with Routed Protocols" section later in this chapter.

Configuring HSRP

Figure 9-3 shows the topology of an IP network in which two routers are configured for HSRP.

Figure 9-3 Example of a network configured for HSRP.



All hosts on the network are configured to use the IP address of the virtual router (in this case, 1.0.0.3) as the default gateway. The command for configuring the default gateway depends on the host's operating system, TCP/IP implementation, and configuration.

Note The configurations shown in this case study use the Enhanced IGRP routing protocol. HSRP can be used with any routing protocol supported by the Cisco IOS software. Some configurations that use HSRP still require a routing protocol to converge when a topology change occurs. The standby router becomes active, but connectivity does not occur until the protocol converges.

The following is the configuration for Router A:

```
hostname RouterA
!
interface ethernet 0
ip address 1.0.0.1 255.0.0.0
standby 1 ip 1.0.0.3
standby 1 preempt
standby 1 priority 110
standby 1 authentication denmark
standby 1 timers 5 15
!
interface ethernet 1
ip address 3.0.0.1 255.0.0.0
!
router eigrp 1
network 1.0.0.0
network 3.0.0.0
```

The following is the configuration for Router B:

```
hostname RouterB
!
interface ethernet 0
ip address 1.0.0.2 255.0.0.0
standby 1 ip 1.0.0.3
standby 1 preempt
standby 1 authentication denmark
standby 1 timers 5 15
!
interface ethernet 1
ip address 2.0.0.2 255.0.0.0
!
router eigrp 1
network 1.0.0.0
network 2.0.0.0
```

The **standby ip** interface configuration command enables HSRP and establishes 1.0.0.3 as the IP address of the virtual router. The configurations of both routers include this command so that both routers share the same virtual IP address. The 1 establishes Hot Standby group 1. (If you do not specify a group number, the default is group 0.) The configuration for at least one of the routers in the Hot Standby group must specify the IP address of the virtual router; specifying the IP address of the virtual router is optional for other routers in the same Hot Standby group.

The **standby preempt** interface configuration command allows the router to become the active router when its priority is higher than all other HSRP-configured routers in this Hot Standby group. The configurations of both routers include this command so that each router can be the standby router for the other router. The 1 indicates that this command applies to Hot Standby group 1. If you do not use the **standby preempt** command in the configuration for a router, that router cannot become the active router.

The **standby priority** interface configuration command sets the router's HSRP priority to 110, which is higher than the default priority of 100. Only the configuration of Router A includes this command, which makes Router A the default active router. The 1 indicates that this command applies to Hot Standby group 1.

The **standby authentication** interface configuration command establishes an authentication string whose value is an unencrypted eight-character string that is incorporated in each HSRP multicast message. This command is optional. If you choose to use it, each HSRP-configured router in the group should use the same string so that each router can authenticate the source of the HSRP messages that it receives. The "1" indicates that this command applies to Hot Standby group 1.

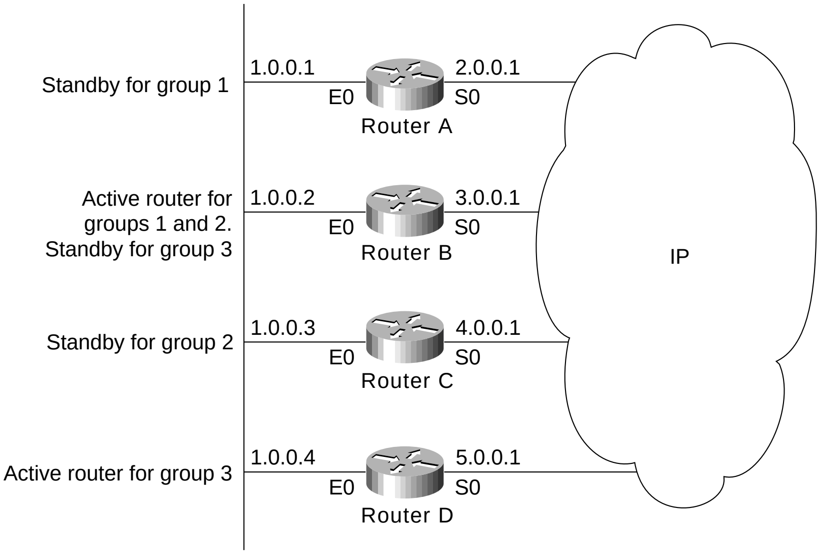
The **standby timers** interface configuration command sets the interval in seconds between hello messages (called the *hello time*) to five seconds and sets the duration in seconds that a router waits before it declares the active router to be down (called the *hold time*) to eight seconds. (The defaults are three and 10 seconds, respectively.) If you decide to modify the default values, you must configure each router to use the same hello time and hold time. The "1" indicates that this command applies to Hot Standby group 1.

Note There can be up to 255 Hot Standby groups on any Ethernet or FDDI LAN. There can be no more than three Hot Standby groups on any Token Ring LAN.

Configuring Multiple Hot Standby Groups

Multigroup HSRP (MHSRP) is an extension of HSRP that allows a single router interface to belong to more than one Hot Standby group. MHSRP requires the use of Cisco IOS Software Release 10.3 or later and is supported only on routers that have special hardware that allows them to associate an Ethernet interface with multiple unicast Media Access Control (MAC) addresses. These routers are the AGS and AGS+ routers and any router in the Cisco 7000 series. The special hardware allows you to configure a single interface in an AGS, AGS+, or Cisco 7000 series router so that the router is the backup router for more than one Hot Standby group, as shown in Figure 9-4.

Figure 9-4 **Example of hot standby groups.**



In Figure 9-4, the Ethernet interface 0 of Router A belongs to group 1. Ethernet interface 0 of Router B belongs to groups 1, 2, and 3. The Ethernet interface 0 of Router C belongs to group 2, and the Ethernet interface 0 of Router D belongs to group 3. When you establish groups, you might want to align them along departmental organizations. In this case, group 1 might support the Engineering Department, group 2 might support the Manufacturing Department, and group 3 might support the Finance Department.

Router B is configured as the active router for groups 1 and 2 and as the standby router for group 3. Router D is configured as the active router for group 3. If Router D fails for any reason, Router B will assume the packet-transfer functions of Router D and will maintain the ability of users in the Finance Department to access data on other subnets. The following is the configuration for Router A:

```
hostname RouterA
!
interface ethernet 0
ip address 1.0.0.1 255.0.0.0
standby 1 ip 1.0.0.5
standby authentication sclara
!
interface serial 0
ip address 2.0.0.1 255.0.0.0
!
router eigrp 1
network 1.0.0.0
network 2.0.0.0
```

The following is the configuration for Router B, which must be an AGS, AGS+, or Cisco 7000 series router:

```
hostname RouterB
!
interface ethernet 0
ip address 1.0.0.2 255.0 0.0
standby 1 ip 1.0.0.5
standby 1 priority 110
standby 1 preempt
standby 1 authentication sclara
standby 2 ip 1.0.0.6
standby 2 priority 110
standby 2 preempt
standby 2 authentication mtview
standby 3 ip 1.0.0.7
standby 3 preempt
standby 3 authentication svale
!
interface serial 0
ip address 3.0.0.1 255.0.0.0
!
router eigrp 1
network 1.0.0.0
network 3.0.0.0
```

The following is the configuration for Router C:

```
hostname RouterC
!
interface ethernet 0
ip address 1.0.0.3 255.0 0.0
standby 2 ip 1.0.0.6
standby 2 authentication mtview
!
interface serial 0
ip address 4.0.0.1 255.0.0.0
!
router eigrp 1
network 1.0.0.0
network 4.0.0.0
```

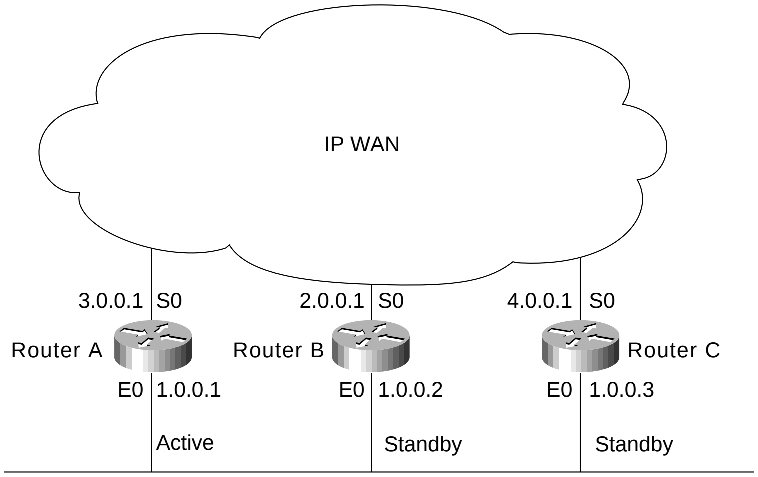
The following is the configuration for Router D:

```
hostname RouterD
!
interface ethernet 0
ip address 1.0.0.4 255.0 0.0
standby 3 ip 1.0.0.7
standby 1 priority 110
standby 1 preempt
standby 3 authentication svale
!
interface serial 0
ip address 4.0.0.1 255.0.0.0
!
router eigrp 1
network 1.0.0.0
network 5.0.0.0
```

Interface Tracking

For both HSRP and MHSRP, you can use the tracking feature to adjust the Hot Standby priority of a router based on whether certain of the router's interfaces are available. When a tracked interface becomes unavailable, the HSRP priority of the router is decreased. You can use tracking to automatically reduce the likelihood that a router that already has an unavailable key interface will become the active router. To configure tracking, use the **standby track** interface configuration command. Figure 9-5 shows a network for which tracking is configured.

Figure 9-5 A network with tracking configured.



In Figure 9-5, Router A is configured as the active router. Routers B and C are configured as standby routers for Router A. The following is the configuration for Router A:

```
hostname RouterA
!
interface ethernet 0
ip address 1.0.0.1 255.0.0.0
standby 1 ip 1.0.0.4
standby 1 preempt
standby 1 priority 110
standby authentication microdot
!
interface serial 0
ip address 2.0.0.1 255.0.0.0
!
router eigrp 1
network 1.0.0.0
network 3.0.0.0
```

The **standby ip** interface configuration command enables HSRP and establishes 1.0.0.4 as the IP address of the virtual router. The "1" establishes Hot Standby group 1. The **standby preempt** interface configuration command allows Router A to become the active router when its priority is higher than all other HSRP-configured routers in the Hot Standby group.

The **standby priority** interface configuration command sets the router's HSRP priority to 110, which is highest priority assigned to the three routers in this example. Because Router A has the highest priority, it is the active router under normal operation. The following is the configuration for Router B:

```
hostname RouterB
!
interface ethernet 0
ip address 1.0.0.2 255.0 0.0
standby 1 ip 1.0.0.4
standby 1 preempt
standby 1 priority 105
standby track serial 0
standby 1 authentication microdot

interface serial 0
ip address 3.0.0.1 255.0.0.0
!
router eigrp 1
network 1.0.0.0
network 2.0.0.0
```

The **standby preempt** interface configuration command allows Router B to become the active router immediately if its priority is highest, even before the current active router fails. The **standby priority** interface configuration command specifies a priority of 105 (lower than the priority of Router A and higher than the priority of Router C), so Router B is a standby router.

The **standby track** interface configuration command causes Ethernet interface 0 to track serial interface 0. If serial interface 0 becomes unavailable, the priority of Router B is reduced by 10 (the default). The following is the configuration for Router C:

```
hostname RouterC
!
interface ethernet 0
ip address 1.0.0.3 255.0 0.0
standby 1 ip 1.0.0.4
standby 1 preempt
standby 1 priority
standby track serial 0
standby 1 authentication microdot
!
interface serial 0
ip address 4.0.0.1 255.0.0.0
!
router eigrp 1
network 1.0.0.0
network 4.0.0.0
```

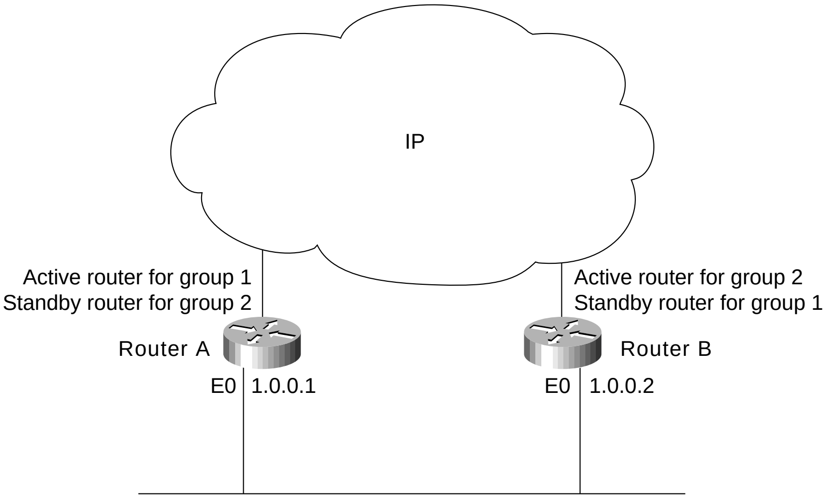
The **standby preempt** interface configuration command allows Router C to become the active router if its priority is highest when the active router fails. The **standby priority** interface configuration command does not specify a priority, so its priority is 100 (the default).

If Router A becomes unavailable and if serial interface 0 on Router B is available, Router B (with its priority of 105) will become the active router. However, if serial interface 0 on Router B becomes unavailable before Router A becomes unavailable, the HSRP priority of Router B will be reduced from 105 to 95. If Router A then becomes unavailable, Router C (whose priority is 100) will become the active router.

Load Sharing

You can use HSRP or MHSRP when you configure load sharing. In Figure 9-6, half of the workstations on the LAN are configured for Router A, and half of the workstations are configured for Router B.

Figure 9-6 Load sharing example.


IP
Active router for group 1
Standby router for group 2
Active router for group 2
Standby router for group 1
Router A
Router B
E0
1.0.0.1
E0
1.0.0.2

The following is a partial configuration for Router A:

```
hostname RouterA
!
interface ethernet 0
ip address 1.0.0.1 255.0.0.0
standby 1 ip 1.0.0.3
standby 1 priority 110
standby 1 preempt
standby 2 ip 1.0.0.4
standby 2 preempt
```

The following is a partial configuration for Router B:

```
hostname RouterB
!
interface ethernet 0
ip address 1.0.0.2 255.0.0.0
standby 1 ip 1.0.0.3
standby 1 preempt
standby 2 ip 1.0.0.4
standby 2 priority 110
standby 2 preempt
```

Together, the configuration files for Routers A and B establish two Hot Standby groups. For group 1, Router A is the default active router, and Router B is the standby router. For group 2, Router B is the default active router, and Router A is the standby router. During normal operation, the two routers share the IP traffic load. When either router becomes unavailable, the other router becomes active and assumes the packet-transfer functions of the router that is unavailable. The **standby preempt** interface configuration commands are necessary so that if a router goes down and then comes back up, preemption occurs and restores load sharing.

Using HSRP with Routed Protocols

This section describes the interaction between HSRP and the following routed protocols:

- AppleTalk, Banyan VINES, and Novell IPX
- DECnet and XNS

AppleTalk, Banyan VINES, and Novell IPX

You can configure HSRP in networks that, in addition to IP, run AppleTalk, Banyan VINES, and Novell IPX. AppleTalk and Novell IPX continue to function when the standby router becomes the active router, but they take time to adapt to topology changes. In general, AppleTalk hosts discover a new active router in less than 30 seconds. Novell 4.*x* hosts discover a new active router in 10 seconds, on average. Novell 2.*x* or Novell 3.*x* hosts might require more time to adapt.

Note Regardless of whether HSRP is configured, Banyan VINES does not respond well to topology changes. When HSRP is configured, the effect of a topology change varies, depending on the type of router that becomes the active router.

When the active router becomes unavailable, or its connection to the network goes down, all Banyan VINES sessions that rely on that router stop and must be reinitiated. If an AGS, AGS+, or Cisco 7000 series router becomes the active router, Banyan VINES traffic flowing through that router is not affected as it changes from standby to active. That is because these routers have special hardware that allows them to have more than one MAC address at the same time. If the router that becomes the active router is *not* an AGS, AGS+, or Cisco 7000 series router, Banyan VINES traffic flowing through that router pauses and resumes after no more than 90 seconds while the router changes from standby to active.

Regardless of which type of router becomes the active router, any Banyan VINES serverless clients that obtained their network-layer address from the unavailable router might need to reboot to obtain another network-layer address.

DECnet and XNS

DECnet and XNS are compatible with HSRP and MHSRP over Ethernet, FDDI, and Token Ring on the Cisco 7000 and Cisco 7500 routers. Some constraints apply when HSRP and MHSRP are configured on other routers, such as the Cisco 2500, Cisco 3000, Cisco 4000, and Cisco 4500 series routers, which do not have the hardware required to support multiple MAC addresses. Table 9-1 identifies the supported and unsupported combinations.

Table 9-1 HSRP and MHSRP Compatibility with DECnet and XNS

Protocol Combination per Interface	Cisc o250 0	Cisco 3000	Cisco 4000	Cisco 4500	Cisco 7000	Cisco 7500
MHSRP with or without DECnet or XNS	No	No	No	No	Yes	Yes
HSRP without DECnet or XNS	Yes	Yes	Yes	Yes	Yes	Yes
HSRP with DECnet or XNS	No	No	No	No	Yes	Yes

Summary

HSRP and MHSRP provide fault-tolerant routing of IP packets for networks that require nonstop access by hosts on all segments to resources on all segments. To provide fault tolerance, HSRP and MHSRP require a routing protocol that converges rapidly, such as Enhanced Interior Gateway Routing Protocol (Enhanced IGRP). A fast-converging protocol ensures that router state changes propagate fast enough to make the transition from standby to active mode transparent to network users.

LAN Switching

Today's local-area networks (LANs) are becoming increasingly congested and overburdened. In addition to an ever-growing population of network users, several factors have combined to stress the capabilities of traditional LANs:

- *Faster CPUs*—In the mid-1980s, the most common desktop workstation was a PC. At the time, most PCs could execute 1 million instructions per second (MIPS). Today, workstations with 50 to 75 MIPS of processing power are common, and I/O speeds have increased accordingly. Two modern engineering workstations on the same LAN can easily saturate it.
- *Faster operating systems*—Until recently, operating system design had constrained network access. Of the three most common desktop operating systems (DOS/Windows, the UNIX operating system, and the Mac OS), only the UNIX operating system could multitask. Multitasking allows users to initiate simultaneous network transactions. With the release of Windows 95, which reflected a redesign of DOS/Windows that included multitasking, PC users could increase their demands for network resources.
- *Network-intensive applications*—Use of client-server applications, such as Network File System (NFS), LAN Manager, NetWare, and World Wide Web is increasing. Client-server applications allow administrators to centralize information, thus making it easy to maintain and protect. Client-server applications free users from the burden of maintaining information and the cost of providing enough hard disk space to store it. Given the cost benefit of client-server applications, such applications are likely to become even more widely used in the future.

Switching is a technology that alleviates congestion in Ethernet, Token Ring, and Fiber Distributed Data Interface (FDDI) LANs by reducing traffic and increasing bandwidth. Such switches, known as *LAN switches*, are designed to work with existing cable infrastructures so that they can be installed with minimal disruption of existing networks. Often, they replace shared hubs. This case study describes how LAN switching works, how virtual LANs work, and how to configure virtual LANs (VLANs) in a topology that consists of Catalyst 5000 LAN switches.

Understanding Switching Basics

The term *switching* was originally used to describe packet-switch technologies, such as Link Access Procedure, Balanced (LAPB), Frame Relay, Switched Multimegabit Data Service (SMDS), and X.25. Today, switching refers to a technology that is similar to a bridge in many ways.

The term *bridging* refers to a technology in which a device (known as a *bridge*) connects two or more LAN segments. A bridge transmits datagrams from one segment to their destinations on other segments. When a bridge is powered and begins to operate, it examines the Media Access Control (MAC) address of the datagrams that flow through it to build a table of known destinations. If the bridge knows that the destination of a datagram is on the same segment as the source of the datagram, it drops the datagram because there is no need to transmit it. If the bridge knows that the destination is on another segment, it transmits the datagram on that segment only. If the bridge does not know the destination segment, the bridge transmits the datagram on all segments except the source segment (a technique known as *flooding*). The primary benefit of bridging is that it limits traffic to certain network segments.

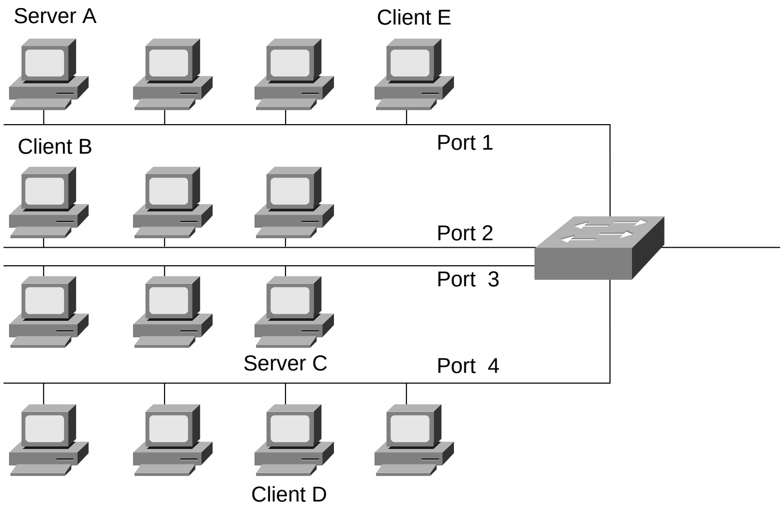
Like bridges, switches connect LAN segments, use a table of MAC addresses to determine the segment on which a datagram needs to be transmitted, and reduce traffic. Switches operate at much higher speeds than bridges, and can support new functionality, such as virtual LANs.

Switching in the Ethernet Environment

The most common LAN media is traditional Ethernet, which has a maximum bandwidth of 10 Mbps. Traditional Ethernet is a half-duplex technology. Each Ethernet host checks the network to determine whether data is being transmitted before it transmits and defers transmission if the network is in use. In spite of transmission deferral, two or more Ethernet hosts can transmit at the same time, which results in a collision. When a collision occurs, the hosts enter a back-off phase and retransmit later. As more hosts are added to the network, hosts must wait more often before they can begin transmitting, and collisions are more likely to occur because more hosts are trying to transmit. Today, throughput on traditional Ethernet LANs suffers even more because users are running network-intensive software, such as client-server applications, which cause hosts to transmit more often and for longer periods of time.

An Ethernet LAN switch improves bandwidth by separating collision domains and selectively forwarding traffic to the appropriate segments. Figure 10-1 shows the topology of a typical Ethernet network in which a LAN switch has been installed.

Figure 10-1 Ethernet switching.



In Figure 10-1, each Ethernet segment is connected to a port on the LAN switch. If Server A on port 1 needs to transmit to Client B on port 2, the LAN switch forwards Ethernet frames from port 1 to port 2, thus sparing port 3 and port 4 from frames destined for Client B. If Server C needs to send data to Client D at the same time that Server A sends data to Client B, it can do so because the LAN switch can forward frames from port 3 to port 4 at the same time it is forwarding frames from port 1 to port 2. If Server A needs to send data to Client E, which also resides on port 1, the LAN switch does not need to forward any frames.

Performance improves in LANs in which LAN switches are installed because the LAN switch creates isolated collision domains. By spreading users over several collision domains, collisions are avoided and performance improves. Many LAN switch installations assign just one user per port, which gives that user an effective bandwidth of 10 Mbps.

Understanding Virtual LANs

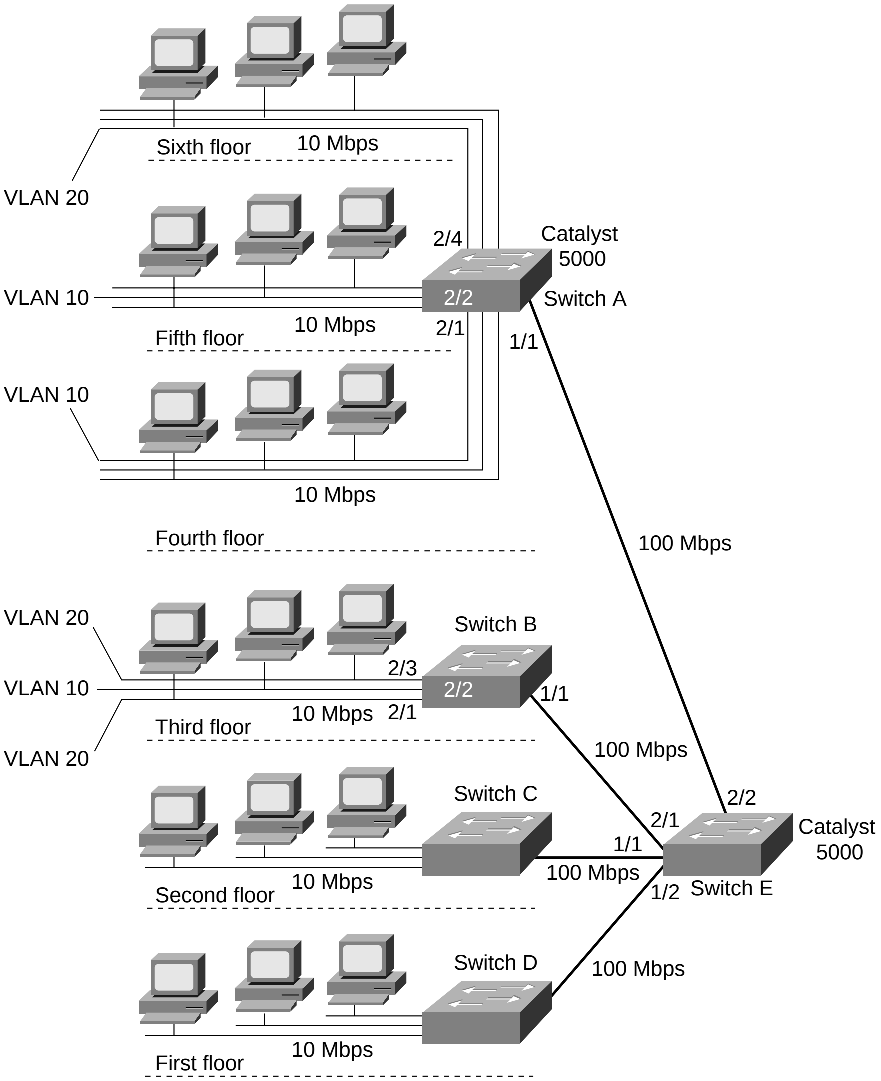
A virtual LAN (VLAN) is a group of hosts or network devices, such as routers (running transparent bridging) and bridges, that forms a single bridging domain. Layer 2 bridging protocols, such as IEEE 802.10 and Inter-Switch Link (ISL), allow a VLAN to exist across a variety of equipment, including LAN switches.

VLANs are formed to group related users regardless of the physical connections of their hosts to the network. The users can be spread across a campus network or even across geographically dispersed locations. A variety of strategies can be used to group users. For example, the users might be grouped according to their department or functional team. In general, the goal is to group users into VLANs so that most of their traffic stays within the VLAN. When you configure VLANs, the network can take advantage of the following benefits:

- *Broadcast control*—Just as switches physically isolate collision domains for attached hosts and only forward traffic out a particular port, VLANs provide logical collision domains that confine broadcast and multicast traffic to the bridging domain.
- *Security*—If you do not include a router in a VLAN, no users outside of that VLAN can communicate with the users in the VLAN and vice versa. This extreme level of security can be highly desirable for certain projects and applications.
- *Performance*—You can assign users that require high-performance networking to their own VLANs. You might, for example, assign an engineer who is testing a multicast application and the servers the engineer uses to a single VLAN. The engineer experiences improved network performance by being on a “dedicated LAN,” and the rest of the engineering group experiences improved network performance because the traffic generated by the network-intensive application is isolated to another VLAN.
- *Network management*—Software on the switch allows you to assign users to VLANs and, later, reassign them to another VLAN. Recabling to change connectivity is no longer necessary in the switched LAN environment because network management tools allow you to reconfigure the LAN logically in seconds.

Figure 10-2 shows an example of a switched LAN topology in which VLANs are configured.

Figure 10-2 Typical VLAN topology.



In Figure 10-2, a 10-Mbps Ethernet connects the hosts on each floor to Catalyst 5000 LAN switches. 100-Mbps Fast Ethernet connects switches A, B, C, and D to Switch E.

Note The Catalyst 5000 has five slots in which modules can be installed. The *supervisor engine* module is always installed in slot 1. The supervisor engine module is the main system processor switch; it provides a console port and two 100-Mbps Fast Ethernet ports. A variety of other modules providing 10-Mbps Ethernet and Fast Ethernet interfaces can be installed in slots 2 through 5. Ports are identified by their slot number and their position, from left to right, on the module. For example, port 2/2 is the second port from the left on the module in slot 2.

The switches in Figure 10-2 communicate with each other using ISL, which is a protocol that maintains VLAN information as traffic flows between the switches. With ISL, an Ethernet frame is encapsulated with a 30-byte header that contains a two-byte VLAN ID.

Figure 10-2 shows that VLAN 20 consists of port 4 in slot 2 on Switch A and ports 1 and 3 in slot 4 on Switch B. Frames exchanged between ports 1/4 and 3/4 are switched by Switch B as normal. On Switch B, any frame generated by ports 1/4 and 3/4 that is not destined for ports 1/4 and 3/4 is encapsulated in an ISL header that includes a VLAN 20 identifier and is sent to Switch E. Switch E examines the ISL header and determines that the frame is intended for VLAN 20 and sends the frame out on port 2/2 to Switch A. Switch A examines the ISL header to determine the VLAN for which the frame is destined, removes the header, and switches it to all ports in VLAN 20 (if the frame is broadcast or multicast) or to port 2/4 if the frame is a unicast.

Configuring the Switches

When a Catalyst 5000 switch first starts up, the following defaults are set:

- The console port is set to 9600 baud, 8 data bits, no parity, and 1 stop bit. If you want to change the baud rate, use the **set system baud** command.
- The Cisco Discovery Protocol (CDP) is enabled on every port to send a CDP message every 60 seconds. If you want to disable CDP on ports that do not have a Cisco device, use the **set cdp disable** command.
- The following Simple Network Management Protocol (SNMP) community strings are defined:
    - "public" for the read-only access type
    - "private" for the read-write access type
    - "secret" for the read-write-all access type

  If you want to set other SNMP community strings, use the **set snmp community** command.
- All modules and all ports are enabled. To disable a module, use the **set module disable** command, and to disable a port, use the **set port disable** command.
- All 10-Mbps Ethernet ports are set to half duplex. Use the **set port duplex** command to set a port to full duplex.

When you first start up a switch, you should set some values that apply to the switch as a whole. For example, you might enter the following commands at the console port of Switch A:

```
set system contact Terry Moran
set system location Norwich
set system name SwitchA
set time fri 9/15/95 14:08:34
set prompt SwitchA>
set password
set enablepass
set interface sc0 131.108.40.1
```

The **set system contact** command establishes "Terry Moran" as the person to contact for system administration. The **set system name** establishes "SwitchA" as the name of this switch. The **set time** command sets the current time, using a 24-hour clock format. The **set prompt** command sets the prompt to "SwitchA>". The default prompt is "Console>".

The **set password** command sets password protection for the administrative interface in normal mode. When you enter the **set password** command, the switch prompts you to enter a password and then prompts you to reenter the password.

The **set enablepass** command sets password protection for the administrative interface in privileged mode. When you enter the **set enablepass** command, the switch prompts you to enter a password and then prompts you to reenter the password.

The **set interface** command assigns an IP address and netmask to interface sc0. After you make this assignment, you can Telnet to the switch to perform administrative tasks. The switch supports up to eight simultaneous Telnet connections. Alternatively, you can use the **set interface** command to enable a Serial Line Interface Protocol (SLIP) connection on the console interface (sl0).

Configuring VLANs on Switch A

The following commands configure VLANs 10 and 20 on Switch A:

```
set vlan 10 2/1,2/2
set vlan 20 2/4
set trunk 1/1 10,20
```

The first **set vlan** command creates VLAN 10 and assigns ports 1 and 2 in slot 2 to it. The second **set vlan** command creates VLAN 20 and assigns port 4 in slot 2 to it.

The **set trunk** command configures port 1 in slot 1 as a trunk and adds VLANs 10 and 20 to it. Trunks are used for Fast Ethernet connections between switches. When a port is configured as a trunk, it runs in ISL mode. To detect and break loops, trunks use the spanning-tree protocol on all VLANs that are carried across the trunk.

Configuring VLANs on Switch B

The following commands configure VLANs 10 and 20 on Switch B:

```
set vlan 10 2/2
set vlan 20 2/1,2/3
set trunk 1/1 10,20
```

The first **set vlan** command creates VLAN 10 and assigns port 2 in slot 2 to it. The second **set vlan** command creates VLAN 20 and assigns ports 1 and 3 in slot 2 to it. The **set trunk** command configures port 1 in slot 1 as a trunk and adds VLANs 10 and 20 to it.

Configuring VLANs on Switch E

The following commands configure VLANs 10 and 20 on Switch E:

```
set trunk 2/1 10,20
set trunk 2/2 10,20
```

The first **set trunk** command configures port 1 in slot 2 as a trunk and adds VLANs 10 and 20 to it. This trunk is used to communicate with Switch B. The second **set trunk** command configures port 2 in slot 2 as a trunk and adds VLANs 10 and 20 to it. This trunk is used to communicate with Switch A.

Summary

LAN switching technology improves the performance of traditional Ethernet, FDDI, and Token Ring technologies without requiring costly wiring upgrades or time-consuming host reconfiguration. The low price per port allows the deployment of LAN switches so that they decrease segment size and increase available bandwidth. VLANs make it possible to extend the benefit of switching over a network of LAN switches and other switching devices.

CHAPTER 11

Multicasting in IP and AppleTalk Networks

Over the past few years, the concept of end-users being able to send and receive audio and video (known collectively as *multimedia*) at the desktop has gained considerable attention and acceptance. With high-performance 486, Pentium, and PowerPC CPUs, more than 80 percent of the personal computers sold during 1995 were multimedia capable. Today, it is not uncommon for end-users to run video editing and image processing applications from the desktop.

The proliferation of more and more multimedia-enabled desktop computers has spawned a new class of multimedia applications that operate in networked environments. These network multimedia applications leverage existing network infrastructure to deliver video and audio applications to end users. Most notable are videoconferencing and video server applications. With these applications, video and audio streams are transferred over the network between peers or between clients and servers. There are three types of multimedia applications:

- *Unicast*—Unicast applications send one copy of each packet to each host that wants to receive the packet. This type of application is easy to implement, but it requires extra bandwidth because the network has to carry the same packet multiple times—even on shared links. Because unicast applications make a copy of each packet, the number of receivers is limited to the number of copies of each packet that can be made by the CPU that runs the unicast application.
- *Broadcast*—Broadcast applications send each packet to a broadcast address. This type of application is easier to implement than unicast applications, but it can have serious effects on the network. Allowing the broadcast to propagate throughout the network is a significant burden on both the network (in terms of traffic volume) and the hosts connected to the network (in terms of the CPU time that each host that does not want to receive the transmission must spend processing and discarding unwanted broadcast packets). You can configure routers to stop broadcasts at the LAN boundary (a technique that is frequently used to prevent broadcast storms), but this technique limits the receivers according to their physical location.
- *Multicast*—Multicast applications send each packet to a multicast group address. Hosts that want to receive the packets indicate that they want to be members of the multicast group. This type of application expects that networks with hosts that have joined a multicast group will receive multicast packets. Multicast applications and underlying multicast protocols control multimedia traffic and shield hosts from having to process unnecessary broadcast traffic.

This case study examines multicast protocols that have been developed for the Internet Protocol (IP) and for AppleTalk, as well as Cisco Internetwork Operating System (Cisco IOS) features that can help your network deliver video and audio smoothly.

Implementing Multicast Applications in IP Networks

Currently, support for IP multicasting comes from three protocols:

- Internet Group Management Protocol (IGMP)
- Protocol-Independent Multicast (PIM)
- Distance Vector Multicast Routing Protocol (DVMRP)

Network multimedia applications for IP use IGMP to join multicast groups. PIM and DVMRP use IGMP to determine the location of hosts that have joined a multicast group.

This section covers the following topics:

- Addressing
- Internet Group Management Protocol
- Protocol-Independent Multicast

Addressing

IP multicasting applications use Class D addresses to address packets. The high-order four bits of a Class D address are set to 1110, and the remaining 28 bits are set to a specific multicast group ID. Class D addresses are typically written as dotted-decimal numbers and are in the range of 224.0.0.0 through 239.255.255.255.

Some multicast group addresses are assigned as well-known addresses by the Internet Assigned Numbers Authority (IANA). These multicast group addresses are called *permanent host groups* and are similar in concept to the well-known TCP and UDP port numbers. Table 11-1 lists the multicast address of three permanent host groups.

Table 11-1 Multicast Addresses for Permanent Host Groups

Permanent Host Group	Multicast Address
Network Time Protocol (NTP)	224.0.1.1
RIP-2	224.0.0.9
Silicon Graphics' Dogfight application	224.0.1.2

Internet Group Management Protocol

The Internet Group Management Protocol (IGMP) uses IP datagrams to allow IP multicast applications to join a multicast group. Membership in a multicast group is dynamic—that is, it changes over time as hosts join and leave the group.

Multicast routers that run IGMP use IGMP host-query messages to keep track of the hosts that belong to multicast groups. These messages are sent to the all-systems group address 224.0.0.1. The hosts then send IGMP report messages listing the multicast groups they would like to join. When the router receives a packet addressed to a multicast group, it forwards the packet to those interfaces that have hosts that belong to that group. If you want to prevent hosts on a particular interface from participating in a multicast group, you can configure a filter on that interface by using the **ip igmp access-group** interface configuration command.

Routers on which GMP is enabled periodically send IGMP host-query messages to refresh their knowledge of memberships present on their interfaces. If, after some number of queries, the router determines that no local hosts are members of a particular multicast group on a particular interface, the router stops forwarding packets for that group and sends a *prune* message upstream toward the source of the packet.

You can configure the router to be a member of a multicast group. This is useful for determining multicast reachability in a network. If a router is configured as a group member it can, for example, respond to an ICMP echo request packet addressed to a group for which it is a member. To configure the router as a member of a multicast group, use the **ip igmp join-group** interface configuration command.

Protocol-Independent Multicast

Protocol-Independent Multicast (PIM) is an IP multicast protocol that works with all existing unicast routing protocols. PIM has two modes that allow it to work effectively with two different types of multicast traffic distribution patterns: dense mode and sparse mode.

Dense mode PIM is designed for the following conditions:

- Senders and receivers are in close proximity to one another.
- There are few senders and many receivers.
- The volume of multicast traffic is high.

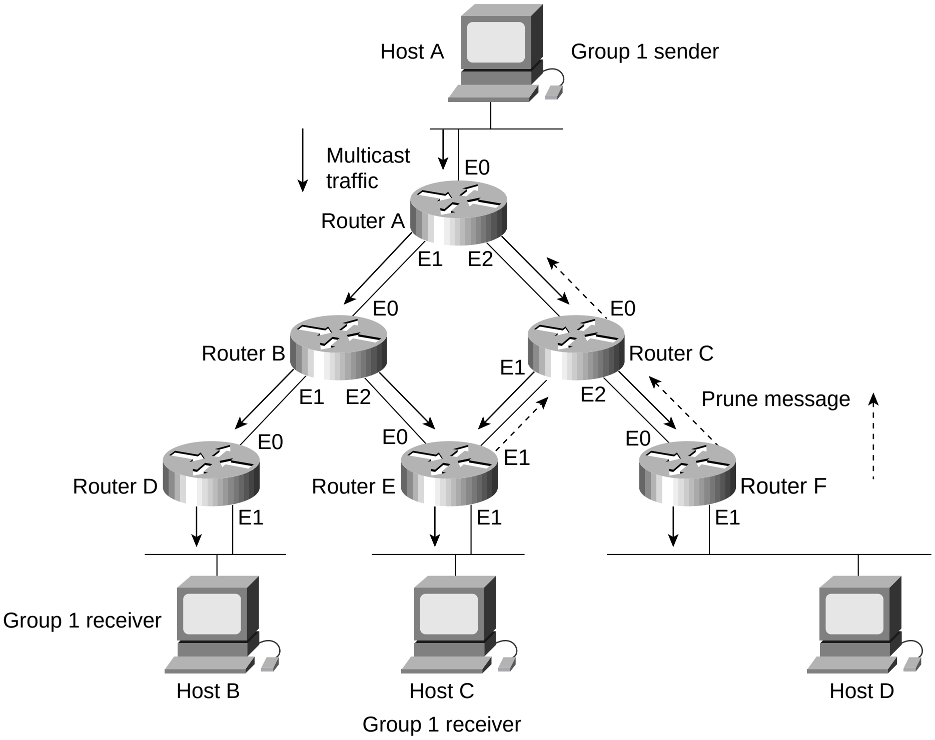
Sparse-mode PIM is designed for the following conditions:

- There are few receivers in a group.
- Senders and receivers are separated by WAN links.

Dense Mode

Dense-mode PIM uses a technique known as *reverse path forwarding*. When a router receives a packet, it sends the packet out all interfaces except the interface on which the packet was received. Reverse path forwarding allows a data stream to reach all LANs, possibly multiple times. If the router has interfaces for which no hosts are members of the multicast group for which the packet is intended or for which no downstream multicast router on that LAN has joined the group, the router sends a prune message up the distribution tree to inform the sender that it need not send subsequent packets for this multicast group. Figure 11-1 shows how PIM works in dense mode.

Figure 11-1 PIM dense-mode operation.



In Figure 11-1, Router A receives multicast traffic from Host A on Ethernet interface 0, duplicates each packet, and sends the packets out on Ethernet interface 1 and Ethernet interface 2 to Routers B and C. Routers B and C duplicate the packets and send them out to Routers D, E, and F. Router D has a host that is a member of Group 1, so Router D does not send a prune message. Router E also has a host that is a member of Group 1, but because it receives the packets on two interfaces, Router E sends a prune message to Router C. (The decision about which router should be pruned is reached through a negotiation process conducted by Router B and Router C. If the connection between Router E and Router B had been a point-to-point link, the prune message would have been sent to Router B automatically, thereby eliminating the need for Routers B and C to negotiate an agreement.)

Router F does not have any hosts that are members of Group 1, so it sends a prune message to Router C. Router C sends a prune message to Router A. After the prune messages are received, Router A sends multicast traffic for Group 1 to Router B only.

When you configure PIM in dense mode, you should enable IP multicast routing on every router over which multicast traffic will flow. The following commands configure dense mode PIM on Router B:

```
ip multicast-routing
interface ethernet 1
ip pim dense-mode
!
interface ethernet 2
ip pim dense-mode
```

The **ip multicast-routing** global configuration command enables IP multicast routing. You should include this command on every router that you want to participate in PIM. If some routers cannot be configured for IP multicast routing (for example, if they do not run a version of the Cisco IOS software release that supports PIM), you need to configure a tunnel so that multicast packets bypass these routers.

The **ip pim** interface configuration command enables PIM on the specified interface, and the **dense-mode** keyword enables dense mode. When you configure PIM in dense mode, you should apply the **ip pim** command with the **dense-mode** keyword to every interface that you want to forward multicast traffic.

Note Enabling PIM automatically enables IGMP.

In dense mode, the PIM-configured interface with the highest IP address on a LAN (subnet) is responsible for sending IGMP host-query messages to all hosts on the LAN. By default, the router that is responsible for sending PIM router-query messages sends them every 30 seconds. If you want to modify this interval, use the **ip pim query-interval** interface configuration command.

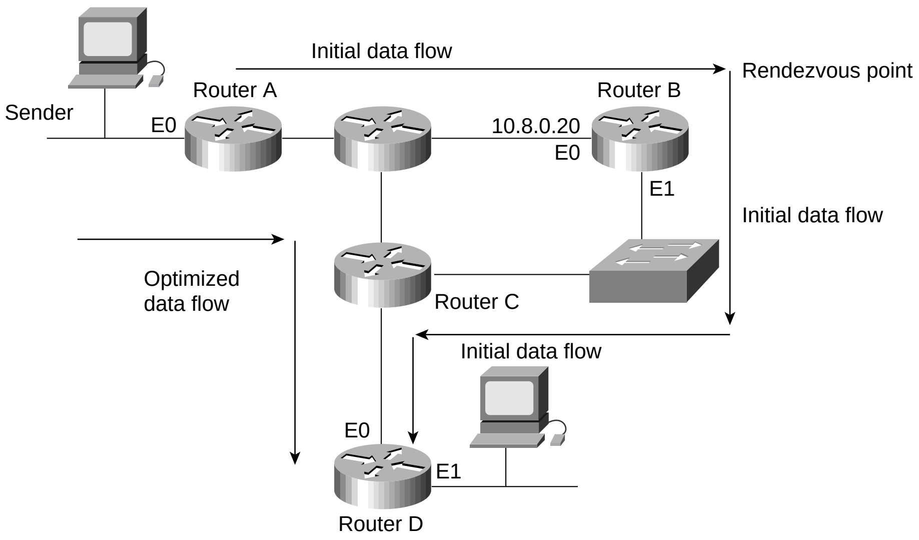
By default, a PIM-configured interface forwards all multicast packets. If you want to control the forwarding of packets, use the **ip multicast-threshold ttl** interface configuration command. The **ip multicast-threshold ttl** command changes the value of time-to-live (TTL) threshold, which the router compares with the TTL field in the IP header. Only those multicast packets that have a TTL greater than the TTL threshold are forwarded. You might, for example, want to set the TTL threshold to a very high value (such as 200) to prevent multicast packets from exiting an area.

Sparse Mode

Sparse-mode PIM is designed for environments in which many multipoint data streams go to a relatively small number of the LAN segments. For this type of environment, dense mode PIM would use bandwidth inefficiently.

Sparse-mode PIM assumes that no hosts want to receive multicast traffic unless they specifically request it. In sparse-mode PIM, a router is designated as a rendezvous point. The rendezvous point collects information about multicast senders and makes that information available to potential receivers. When a sender wants to send data, it first sends the data to the rendezvous point. When a receiver wants to receive data, it registers with the rendezvous point. When the data stream begins to flow from sender to rendezvous point to receiver, the routers in the path automatically optimize the path to remove any unnecessary hops. Figure 11-2 shows how PIM works in sparse mode.

Figure 11-2 PIM sparse-mode operation.



In Figure 11-2, Routers A and D are leaf routers. *Leaf routers* are routers that are directly connected either to a receiver or sender of multicast messages. The sparse-mode configuration of a leaf router designates one or more routers as rendezvous points. In this example, Router B is designated as the rendezvous point.

The leaf router that is directly connected to a sender (in this case, Router A) sends PIM register messages on behalf of the sender to the rendezvous point. The leaf router that is directly connected to a receiver (in this case, Router B) sends PIM join and prune messages to the rendezvous point to inform it about group membership. The following commands configure Router A for sparse mode:

```
ip multicast-routing
ip pim rp-address 10.8.0.20 1
!
interface ethernet 0
ip pim sparse-mode
!
interface ethernet 1
ip pim sparse-mode
!
access-list 1 permit 224.0.1.2
```

The following commands configure Router D for sparse mode:

```
ip multicast-routing
ip pim rp-address 10.8.0.20 1
!
interface ethernet 0
ip pim sparse-mode
!
interface ethernet 1
ip pim sparse-mode
!
access-list 1 permit 224.0.1.2
```

The **ip multicast-routing** global configuration command enables IP multicast routing. When you configure PIM, IP multicast routing must be enabled on every router over which multicast traffic will flow. If some routers cannot be configured for IP multicast routing (for example, if they do not run a version of the Cisco IOS Software that supports PIM), you need to configure a tunnel so that multicast packets bypass these routers.

The **ip pim rp-address** global configuration command specifies the IP address of an interface on the router that is to be the rendezvous point and specifies that access list 1 is to be used to define the multicast groups for which the rendezvous point is to be used. The **ip pim rp-address** command must be configured on every sparse-mode router.

The **ip pim** interface configuration command enables PIM on the interface, and the **sparse-mode** keyword enables sparse mode. When you configure PIM in sparse mode, you should apply the **ip pim** command with the **sparse-mode** keyword to every interface that you want to forward multicast traffic. The **access-list** global configuration command defines a standard IP access list that permits traffic using the multicast address 224.0.1.2 (the Silicon Graphics Dogfight application).

In sparse mode, the PIM-configured interface with the highest IP address on a LAN (subnet) is responsible for sending IGMP host-query messages to all hosts on the LAN and for sending PIM register and join messages toward the rendezvous point.

Note To configure a router as a rendezvous point, add the **ip multicast-routing** command and the **ip pim** command with the **sparse-mode** keyword to its configuration. The router recognizes its own IP address as the address of the rendezvous point and automatically assumes the functions of a rendezvous-point function.

Interoperability with Distance Vector Multicast Routing Protocol

The Distance Vector Multicast Routing Protocol (DVMRP) is another multicast protocol that has been developed for IP. DVMRP is similar to dense-mode PIM in that it uses reverse path forwarding. When a router receives a packet, it sends the packet out all interfaces except the interface that leads back to the source of the packet. If the router has interfaces for which no hosts are members of the multicast group for which the packet is intended, the router sends a prune message up the distribution tree to inform the sender that it need not send subsequent packets for this multicast group.

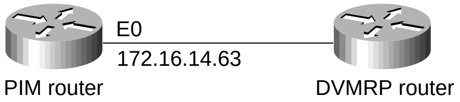
Although the Cisco IOS software does not support DVMRP, it does support interoperability with DVMRP-configured routers. PIM-configured routers dynamically discover DVMRP-configured routers on attached networks. When a DVMRP neighbor is discovered, PIM-configured routers periodically transmit DVMRP report messages advertising the unicast sources that are reachable in the PIM domain. By default, directly connected subnets and networks are advertised. The PIM-configured router forwards multicast packets that it receives from DVMRP routers into the PIM domain and, in turn, forwards multicast packets from the PIM domain to DVMRP routers.

Note When PIM-configured routers are directly connected to DVMRP routers or interoperate with DVMRP routers over a tunnel, the DVMRP routers should run *mrouted* Version 3.8. (The mrouted protocol is a public domain implementation of DVMRP.)

Interoperability Between Directly Connected Routers

Figure 11-3 illustrates a topology in which a PIM-configured router is directly connected to a DVMRP-configured router.

Figure 11-3 PIM and DVMRP interoperability.



The following commands configure the PIM router for interoperability with the DVMRP router:

```
ip multicast-routing
!
interface ethernet 0
ip address 172.16.14.63 255.255.0.0
ip pim dense-mode
ip dvmrp metric 1 list 1
ip dvmrp metric 0 list 2
!
access-list 1 permit 192.168.35.0 0.0.0.255
access-list 1 permit 192.168.36.0 0.0.0.255
access-list 1 permit 192.168.37.0 0.0.0.255
access-list 1 deny 0.0.0.0 255.255.255.255
access-list 2 permit 0.0.0.0 255.255.255.255
```

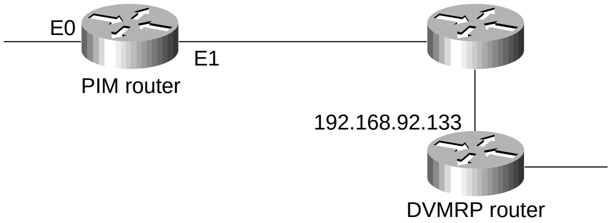
The **ip dvmrp metric** interface configuration commands configure the metric that is to be associated with a set of destinations for DVMRP reports. The first **ip dvmrp metric** command causes the routes specified by access list 1 to be advertised to the DVMRP router (in this case, networks 192.168.35.0, 192.168.36.0, and 192.168.37.0). The second **ip dvmrp metric** command indicates that the routes specified by access list 2 are not to be advertised (in this case, all other routes). If you do not specify the routes that are to be advertised, only those subnets and networks that are directly connected to the PIM router will be advertised.

Interoperability over a Tunnel

DVMRP tunnels are used when one or more routers on a path do not support multicast routing. The router then sends and receives multicast packets over the tunnel. This allows a PIM domain to connect to a DVMRP router.

When a PIM-configured router interoperates with DVMRP over a tunnel, it advertises source routes in DVMRP report messages. In addition, the router caches any DVMRP report messages that it receives. The router uses the cached report messages as part of its reverse path forwarding calculation. This allows the router to forward multicast packets that it receives over the tunnel. Figure 11-4 illustrates interoperability with DVMRP over a tunnel interface.

Figure 11-4 PIM and DVRMP interoperability over a tunnel interface.



The following commands configure the PIM router:

```
ip multicast-routing
!
interface tunnel 0
ip address 192.168.47.1 255.255.255.0
ip pim dense-mode
tunnel source ethernet 1
tunnel destination 192.168.92.133
tunnel mode dvmrp
!
interface ethernet 1
ip address 192.168.23.23 255.255.255.0 secondary
ip address 192.168.243.2 255.255.255.0
ip pim dense-mode
ip dvmrp accept-filter 1
!
access-list 1 permit 192.168.48.0 0.0.0.255
access-list 1 permit 192.168.49.0 0.0.0.255
access-list 1 permit 192.168.50.0 0.0.0.255
access-list 1 deny 0.0.0.0 255.255.255.255
```

The **interface tunnel** global configuration command creates a tunnel (that is, a virtual interface). The **tunnel source** interface configuration command specifies the interface that participates in the tunnel. The **tunnel destination** interface configuration command specifies the IP address of the mrouted multicast router at the other end of the tunnel.

The **tunnel mode** interface configuration command uses the **dvmrp** keyword to configure the tunnel as a DVMRP tunnel. The **ip address** interface configuration command assigns an address to the tunnel to enable the sending of IP packets over the tunnel and to cause the router to perform DVMRP summarization. Alternatively, the **ip unnumbered** interface configuration command can be used. Either method allows IP multicast packets to flow over the tunnel. If the tunnel has a different network number than the subnet, subnets will not be advertised over the tunnel. In this case, only the network number is advertised over the tunnel.

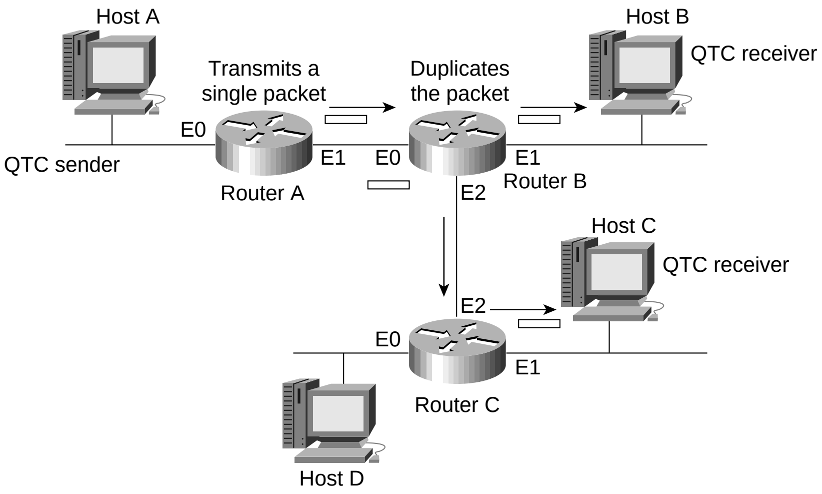
By specifying the **dense-mode** keyword, the **ip pim** interface configuration command configures dense-mode PIM on the interface. The **ip dvmrp accept-filter** interface configuration command configures an acceptance filter for incoming DVMRP reports. Routes that match the specified access list (in this case, access list 1) are stored in the DVMRP routing table (in this case, 192.168.48.0, 192.168.49.0, and 192.168.50.0). If a Cisco router is a neighbor to router running mrouted Version 3.6, the Cisco router can be configured to advertise network 0.0.0.0 to the DVMRP neighbor by using the **ip dvmrp default-information** command and specifying the **originate** keyword.

Using AppleTalk Multicasting

For AppleTalk, the Simple Multicast Routing Protocol (SMRP) supports the routing of multicast packets to multicast groups, with packet replication occurring only on those interfaces that have hosts that belong to the multicast group.

Network multimedia applications, such as QuickTime Conferencing (QTC), allow two or more hosts to participate in a QuickTime Conferencing session. End-users join the multicast group for the multicast transmissions they want to receive. SMRP conserves bandwidth by routing AppleTalk packets to all members of a multipoint group without producing duplicate packets on a particular network segment. Figure 11-5 shows how SMRP works in an AppleTalk network.

Figure 11-5 SMRP in an AppleTalk network.



Router A receives a multicast packet from Host A and sends it to Router B. Two interfaces on Router B have hosts that have registered to receive this multicast transmission, so Router B duplicates the packet and sends one packet out on Ethernet interface 1 and the other packet out on Ethernet interface 2. Only one interface on Router C has hosts that have registered to receive this multicast transmission, so Router C sends the packet out on Ethernet interface 1. The following commands configure SMRP on Router A:

```
smrp routing
!
interface ethernet 0
smrp protocol appletalk
!
interface ethernet 1
smrp protocol appletalk
```

The following commands configure SMRP on Router B:

```
smrp routing
!
interface ethernet 0
smrp protocol appletalk
!
interface ethernet 1
smrp protocol appletalk

interface ethernet 2
smrp protocol appletalk
```

The following commands configure SMRP on Router C:

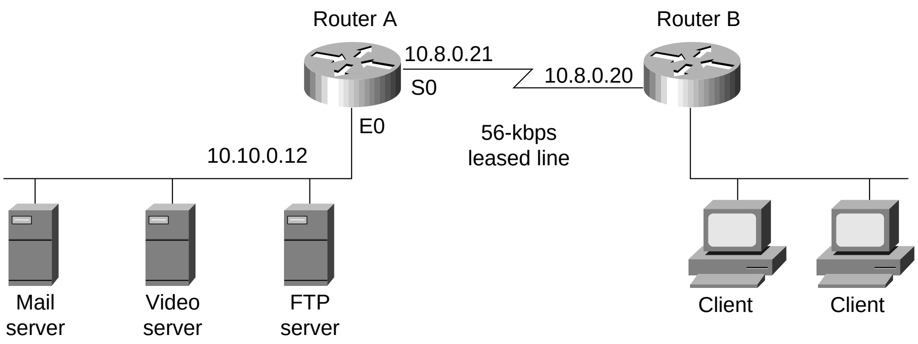
```
smrp routing
!
interface ethernet 0
smrp protocol appletalk
!
interface ethernet 1
smrp protocol appletalk
!
interface ethernet 2
smrp protocol appletalk
```

The **smrp routing** global configuration command enables SMRP routing. The **smrp protocol** interface configuration command enables SMRP on the interface, and the **appletalk** keyword specifies AppleTalk as the OSI Layer 3 protocol for SMRP.

Multicasting over WAN Connections

For the most part, users cannot detect the irregular arrival of data packets, but they can easily detect the irregular arrival of multimedia data, especially when that data includes an audio portion. Irregularly delivered video data is characterized by visible jitter and audible distortion. Smoothing jitter and distortion is especially desirable when multimedia data shares a low-bandwidth link with data traffic, as shown in Figure 11-6.

Figure 11-6 Multicast over WAN connections.



The Cisco IOS software provides three queuing algorithms that you can use to ensure that multicast traffic arrives at its destination without jitter and distortion: weighted fair queuing, priority queuing, and custom queuing. The queuing algorithm that is best for any particular network depends on the traffic flow characteristics of that network. You might want to try all three algorithms to determine the algorithm that provides the smoothest delivery for your particular network connection.

Weighted Fair Queuing

Weighted fair queuing (introduced in Cisco IOS Software Release 11.0) is enabled by default for all interfaces that have a bandwidth less than or equal to 2048 megabits per second (Mbps) and that do not use Link Access Procedure, Balanced (LAPB), X.25, PPP, or Synchronous Data Link Control (SDLC) encapsulations. (Weighted fair queuing cannot be enabled on interfaces that use these encapsulations.) Weighted fair queuing is a traffic priority management algorithm that identifies conversations (traffic streams) and breaks them up to ensure that capacity is shared fairly. The algorithm examines fields in the packet header to identify unique conversations. For example, for AppleTalk, the algorithm uses the source network, node, and socket number; the destination network, node, and socket number; and the type. For IP, the algorithm uses the protocol, source and destination IP address; source and destination port number; and the TOS (type of service) field.

The weighted fair queuing algorithm sorts conversations into two categories: those that have high bandwidth requirements with respect to the capacity of the interface (such as FTP traffic) and those that have low bandwidth requirements (such as interactive sessions). For streams that have low-bandwidth requirements, the algorithm provides access with little or no queuing, and it shares the remaining bandwidth among other conversations. In effect, weighted fair queuing gives low-bandwidth traffic priority over high-bandwidth traffic, and high-bandwidth traffic shares the transmission service proportionally.

When weighted fair queuing is enabled on an interface, new messages for high-bandwidth conversations are discarded when the congestive-messages threshold is reached (the default congestive-messages threshold is 64 messages). To change the congestive-messages threshold, enter the following command, in which **number** is a value between 1 and 512:

```
fair-queue number
```

Priority Queuing

Priority queuing allows you to establish queuing priorities based on protocol type. When you enable priority queuing on an interface, weighted fair queuing is disabled for that interface automatically. The following commands configure priority queuing to ensure a certain quality of service level for Intel ProShare videoconferencing on Router A in Figure 11-6:

```
interface serial 0
ip address 10.8.0.21 255.0.0.0
priority-group 1
!
access-list 101 permit ip any any
!
priority-list 1 protocol IP high UDP 5715
priority-list 1 protocol IP medium TCP 25
priority-list 1 protocol IP normal TCP 20
```

The **priority-group** interface configuration command assigns priority list 1 to serial interface 0. The **priority-list protocol** global configuration commands establish a priority list that is associated with priority group 1. The priority list gives high priority to UDP packets destined for port number 5715 (the port number used by Intel ProShare), medium priority to TCP packets destined for port number 25 (SMTP mail), and normal priority to TCP packets destined for port number 20 (FTP data).

Custom Queuing

Another way to assure the timely delivery of multicast packets is to use custom queuing. With custom queuing, you can define up to 16 queues, assigning normal data to queues 1 through 15 and assigning system messages, such as keepalive messages, to queue 16. The router services each queue sequentially, transmitting a configurable percentage of traffic on each queue before transmitting packets from the next queue.

Custom queuing guarantees that mission-critical data is always assigned a certain percentage of the bandwidth, and it also assures predictable throughput for other traffic. For that reason, custom queuing is recommended for networks that need to provide a guaranteed level of service for all traffic.

When you enable custom queuing on an interface, weighted fair queuing is disabled for that interface automatically. The following commands configure custom queuing for Router A in Figure 11-6:

```
interface serial 0
ip address 10.8.0.21 255.0.0.0
custom-queue-list 1
!
access-list 101 permit ip any any
!
queue-list 1 queue 1 byte-count 57900
queue-list 1 queue 2 byte-count 19300
queue-list 1 queue 3 byte-count 19300
!
queue-list 1 protocol IP 1 UDP 5715
queue-list 1 protocol IP 2 TCP 20
queue-list 1 protocol IP 3 TCP 25
```

The **custom-queue-list** interface configuration command assigns custom queue list 1 to serial interface 0. The **queue-list queue byte-count** global configuration commands specify the size in bytes for three custom queues (in this case, 57,900, 19,300, and 19,300). Together, these **queue-list queue byte-count** commands have the effect of assigning 60 percent of the interface's bandwidth to packets in queue 1, 20 percent of the interface's bandwidth to queue 2, and 20 percent of the interface's bandwidth to queue 3.

The first **queue-list protocol** global configuration command assigns UDP packets destined for port 5715 to queue 1. The second **queue-list protocol** command assigns TCP packets destined for port 20 (SMTP mail) to queue 2, and the third **queue-list protocol** command assigns TCP packets destined for port 25 (FTP data) to queue 3.

Summary

The current popularity of network multimedia applications, such as videoconferencing, is driving the development of protocols that channel the flow of multicast packets to the networks and hosts that want to receive them. As multicasting protocols are deployed, unicast and broadcast applications will be upgraded to take advantage of multicast support, and new multicast applications will be developed.

Scaling Dial-on-Demand Routing

This case study describes the design of an access network that allows a large number of remote sites to communicate with an existing central-site network. The remote sites consist of local-area networks (LANs) that support several workstations. The workstations run transaction processing software that accesses a database located at the central site. The following objectives guided the design of the access portion of the network:

- The existing network could not be modified to accommodate access by the remote sites.
- The central site must be able to connect to any remote site at any time, and any remote site must be able to connect to the central site at any time.
- When choosing between alternative technologies, choose the most cost-effective technology.
- The design must be flexible enough to accommodate additional remote sites in the future.

Network Design Considerations

The following considerations influenced the design of this network:

- Traffic Patterns
- Media Selection
- Application Protocol Requirements

Traffic Patterns

An analysis of the anticipated traffic indicated that each remote site would call the central site an average of four times an hour throughout the business day. This type of traffic pattern means that cost savings can be realized at the central site by providing one telephone line for every 2.5 remote sites (for a total of 48 telephone lines). To spread the calls evenly among the 48 lines, the remote sites connect through a hunt group. The hunt group provides an additional benefit in that all of the remote routers dial the same telephone number to access the central site, which makes the configurations of the remote site routers easier to maintain.

In order to complete a transaction initiated by a remote-site, the central site sometimes needs to call that remote site shortly after it has disconnected from the central site. To make this possible, the access network must converge rapidly. The central site also calls the remote sites periodically to update the transaction processing software on the remote workstations.

Media Selection

The designers chose asynchronous dial-up technology through the Public Switched Telephone Network (PSTN) for the following reasons:

- *Availability*—PSTN is available at all of the remote sites. Potential alternatives, such as Frame Relay and Integrated Digital Services Network (ISDN), were not available at some of the remote sites.
- *Bandwidth*—The transaction processing software causes a small amount of data to be transferred between the remote sites and the central site. For this type of low-bandwidth application, the bandwidth provided by asynchronous dial-up is acceptable. Occasionally, the central site dials the remote sites in order to maintain the transaction processing software on the remote clients. This activity will occur at night (in the absence of transaction processing activity), so the bandwidth provided by asynchronous dial-up is adequate.
- *Cost*—Given the low-bandwidth requirement, the cost of installing and operating Frame Relay or ISDN equipment could not be justified.

Note Although the network described in this case study uses asynchronous dial-up technology over the PSTN, most of the concepts, such as routing strategy and addressing, also apply when scaling other circuit-switched technologies (such as ISDN).

Application Protocol Requirements

The remote workstations run transaction processing software that uses the Transmission Control Protocol /Internet Protocol (TCP/IP) to connect to a database located at the central site. The remote workstations have no need to run any other network-layer protocol. Given this requirement, the most cost-effective choice of router for the remote site is a router that provides an Ethernet interface and an asynchronous interface, and that supports the Routing Information Protocol (RIP).

The Hardware Solution

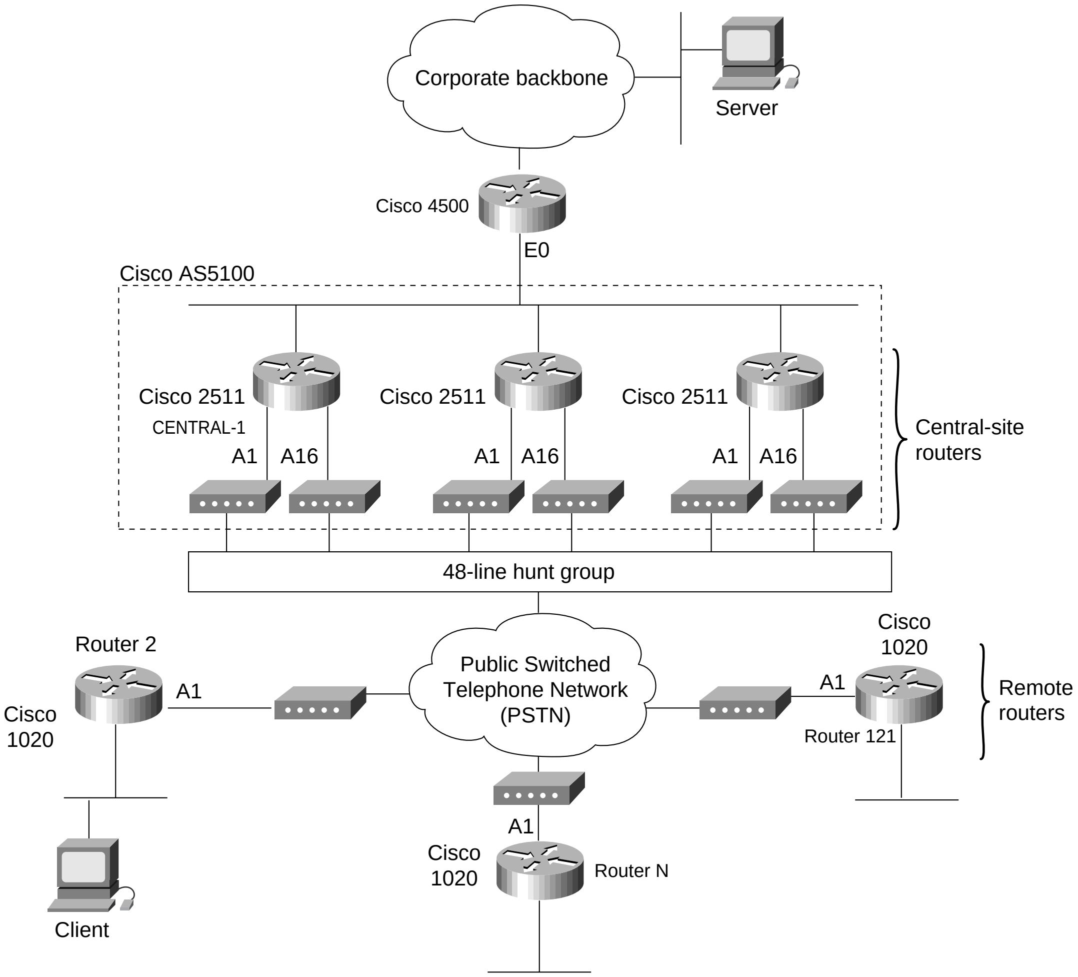
A Cisco AS5100 is installed at the central site to provide 48 asynchronous interfaces. The Cisco AS5100 consists of three access server cards based on the Cisco 2511 access server, making the Cisco AS5100 equivalent to three Cisco 2511 access servers. Each access server card provides 16 asynchronous lines. Each asynchronous line is equipped with a built-in U.S. Robotics Courier modem.

Note For the purposes of this case study, the three Cisco AS5100 access server cards are referred to as the central-site access routers.

Each remote site is equipped with a Cisco 1020 router. The Cisco 1020 provides a single asynchronous interface and an Ethernet interface for connecting to the remote site LAN. The Cisco 1020 runs a limited set of protocols, including TCP/IP and RIP. U.S. Robotics Sportster modems provide connectivity at the remote sites. Using the same brand of modem throughout the access network simplifies chat scripts and modem definition, and makes the network more manageable.

A Cisco 4500 controls routing between the new access portion of the network and the backbone. In particular, the Cisco 4500 ensures that when hosts on the other side of the backbone need to connect to a remote site, the connection is made through the optimum central-site access router. Figure 12-1 shows the topology of the access portion of the network.

Figure 12-1 Remote access topology.



The Software Solution

The configuration of the central-site access routers and the remote site routers must provide the following:

- Authentication
- Network Layer Addressing
- Routing Strategy

Authentication

Traffic between the remote sites and the central site includes confidential information. For that reason, authentication is a primary concern. There are two ways for sites to authenticate themselves:

- *Point-to-Point Protocol (PPP) authentication*—Either the Password Authentication Protocol (PAP) or the Challenge Handshake Authentication Protocol (CHAP) can be used.
- *Login authentication*—With login authentication, the router prompts for a host name and password when a remote router dials in. The remote router logs in and starts PPP.

In either case, the database of usernames and passwords can be stored locally or on an extended Terminal Access Controller Access System (TACACS+) server. TACACS+ provides centralized password management for all the central-site access routers and detailed accounting information about connections to and from the remote sites.

For the purposes of this network design, login authentication is used because it allows the remote sites to announce their IP addresses to the central-site access routers, as described in the section "Network Layer Addressing" later in this chapter. Alternatively, PPP could be started automatically if TACACS+ were used to support per-user IP address assignment.

Network Layer Addressing

Network layer addressing is accomplished through two strategies:

- Subnet Address Assignment
- Next Hop Address

Subnet Address Assignment

The remote routers and the central-site access routers have no need to connect to the Internet, so they use RFC 1597 addresses. The Class B address 172.16.0.0 is used for the entire access portion of the network, and Class C equivalent addresses are assigned to the remote routers. Each subnet gets one Class C equivalent (172.16.x.0 with a mask of 255.255.255.0), which makes addressing easy to manage. Network 172.16.1.0 is reserved for numbering the dialer cloud later if needed. (The dialer cloud is defined as the subnet to which all of the asynchronous interfaces are attached.)

Initially, the dialer cloud is unnumbered. If, in the future, the dialer cloud were to be numbered, the following questions must be considered:

- Can the dialer cloud use the same subnet mask as the remote sites? If not, variable length subnet mask (VLSM) support will be required. (RIP does not support VLSM.)
- Would the use of multiple subnetted Class C addresses cause discontiguous subnets at the remote sites? If so, discontiguous subnet support will be required. (RIP does not support discontiguous subnets.)

In this network, these issues are not a problem. A mask of 255.255.255.0 can be used everywhere, so there are no VLSM concerns. All subnets are from the same major Class B network, so there are no discontiguous subnet concerns. Table 12-1 summarizes the addressing for the access portion of the network.

Table 12-1 Addressing Summary

Site	Subnet	Mask
Central access site[1]	172.16.1.0	255.255.255.0
Router2	172.16.2.0	255.255.255.0
Router3	172.16.3.0	255.255.255.0
...	...	...
Router121	172.16.121.0	255.255.255.0

[1] Can be used for numbering the dialer cloud.

Next Hop Address

To facilitate an accurate routing table and successful IP Control Protocol (IPCP) address negotiation, all next-hop IP addressing must be accurate at all times. To accomplish this, the remote sites need to know the IP address that they will dial in to, and the central site needs to know the IP address of the remote site that has dialed in.

All central-site access routers use the same IP address on all of their asynchronous interfaces. This is accomplished by configuring the Dialer20 interface for IP unnumbered off of a loopback interface. The IP address of the loopback interface is the same on all of the central-site access routers. This way, the remote routers can be configured with the IP address of the router to which it connects, regardless of which router the remote router dials in to.

The remote router needs to announce its IP address to the central-site router when the remote router connects. This is accomplished by having the remote router start PPP on the central site using the EXEC command **ppp 172.16.x.1**. To support this, each central-site access router is configured with the **async dynamic address** interface configuration command.

Note The autoselect feature allows the router to start an appropriate process, such as PPP, automatically when it receives a starting character from the router that has logged in. To use autoselect, a mechanism for supporting dynamic IP address assignment would be required, such as per-user address support in TACACS+.

Routing Strategy

The development of the routing strategy for this network is based on the following two requirements:

- When a particular remote site *is not* dialed in to the central site, that remote site must be reachable through any central-site access router by means of a static route configured in each central-site access router.
- When a particular remote site router *is* logged in to a central-site access router, that remote site must be reachable through that central-site access router by means of the dynamic route that has been established for that connection and propagated to the backbone.

To meet these requirements, the central-site access routes advertise the major network route of the remote sites to the Cisco 4500. All routes to the remote sites are equal-cost through all of the central-site access routers. Each central-site access router is configured to have a static route to each remote site. To allow the Cisco 4500 to use all of the central-site access routers for connecting to the remote sites, the **no ip route-cache** interface configuration command is configured on Ethernet interface 0 of the Cisco 4500, disabling fast switching of IP to the subnet shared with the central-site access routers. This causes the Cisco 4500 to alternate between the three access routers when initiating outbound calls. This strategy increases network reliability for those cases when one of the access routers goes down.

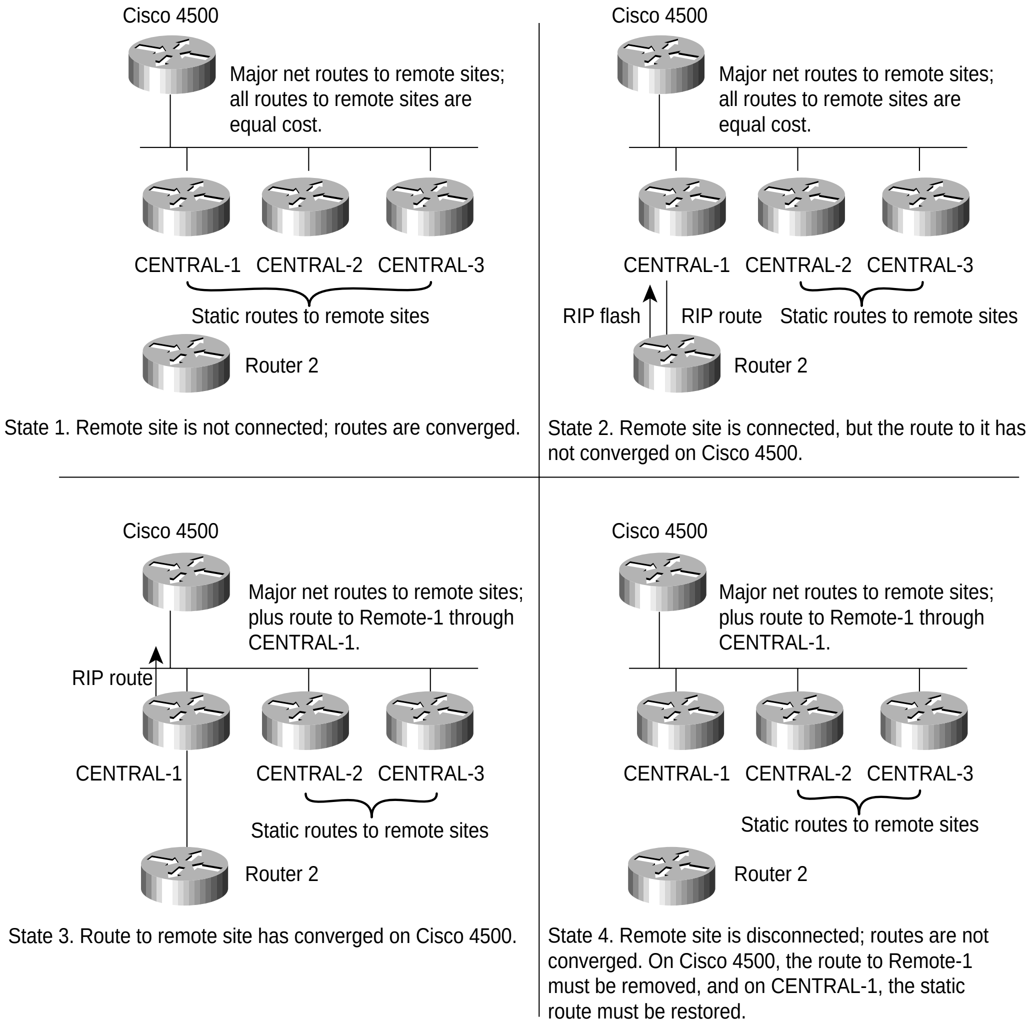
When a remote router logs in, it announces its IP address and sends a RIP flash. The RIP flash causes a dynamic route to the remote site to be installed immediately in the routing table of the central-site access router. The dynamic route overrides the static route for the duration of the connection.

Next, the central-site access router redistributes the RIP route into Open Shortest Path First (OSPF) and sends the route to all of its OSPF neighbors, including the Cisco 4500, which installs it in its routing table. The Cisco 4500 now has a major network route to all of the remote sites, plus a dynamic route to the specific remote site that has logged in. If a central-site host needs to communicate with a particular remote site that is currently logged in, it does so through the dynamic route.

When the remote site logs out, the dynamic route must be removed from the Cisco 4500, and the static route to the remote site must be restored on the central-site access router into which the remote router logged in.

If a central-site host requires communication with a remote site that is not logged in, it will use the major network route defined in the Cisco 4500. A central-site access router, selected in round-robin fashion, is used to initiate the call to the remote site via the static route that is defined for it in the configuration for the selected access router. As in the case of a remote site that calls the central site, once the connection is made, the remote-site router sends a RIP flash that causes a dynamic route to the remote site to be installed immediately in the routing table of the central-site access router. This dynamic route is redistributed into OSPF and is installed in the routing table of the Cisco 4500. Figure 12-2 uses a state diagram to summarize the routing strategy.

Figure 12-2 Routing strategy state diagram.


Cisco 4500
Major net routes to remote sites; all routes to remote sites are equal cost.
CENTRAL-1
CENTRAL-2
CENTRAL-3
Static routes to remote sites
Router 2
State 1. Remote site is not connected; routes are converged.
Cisco 4500
Major net routes to remote sites; all routes to remote sites are equal cost.
CENTRAL-1
CENTRAL-2
CENTRAL-3
RIP flash
RIP route
Static routes to remote sites
Router 2
State 2. Remote site is connected, but the route to it has not converged on Cisco 4500.
Cisco 4500
Major net routes to remote sites; plus route to Remote-1 through CENTRAL-1.
RIP route
CENTRAL-1
CENTRAL-2
CENTRAL-3
Static routes to remote sites
Router 2
State 3. Route to remote site has converged on Cisco 4500.
Cisco 4500
Major net routes to remote sites; plus route to Remote-1 through CENTRAL-1.
CENTRAL-1
CENTRAL-2
CENTRAL-3
Static routes to remote sites
Router 2
State 4. Remote site is disconnected; routes are not converged. On Cisco 4500, the route to Remote-1 must be removed, and on CENTRAL-1, the static route must be restored.

The following convergence issues pertain to the state diagram shown in Figure 12-2:

- During the time between State 2 and State 3, a host at the central site might initiate a call to the remote site. Until State 3, at which time the routes converge on the Cisco 4500, any central-site access router that dials the remote site will fail with a busy signal. In practice, only one call fails: by the time a second connection attempt is made, the routes will have converged in State 3, the dynamic route will be available for use, and there will be no need to make another call.
- When the remote site disconnects, at minimum 120 seconds will elapse before the static route is restored to the routing table of the central-site access router on which the remote site logged in. First, up to 35 seconds might elapse before RIP determines that the remote site has disconnected and is no longer sending RIP updates. Sixty seconds later, the central-site access router scans its routing table and restores one of the two static routes for the remote site, and sixty seconds after that, it scans its routing table again and restores the second of the two static routes. (For information about why there are two static routes for each remote site, see the section “Static Routing Configuration” later in this chapter.)

Note Fast install of static routes is a new feature in Cisco IOS Software Release 11.1 that quickly converges back to the static route when a remote site disconnects.

If, before convergence occurs, the Cisco 4500 directs a call through CENTRAL-1 to Router 2, the call will fail and must be retried. IP fast switching is turned off on the Cisco 4500, so the Cisco 4500 (which is using equal-cost paths to each of the central-site access routers) will send the next packet through CENTRAL-2 or CENTRAL-3 (which still have a static route for Router 2) and the call will go through.

Note When developing the routing strategy for this network, the designers considered the use of snapshot routing, which reduces connection cost by limiting the exchange of routing protocol updates. For snapshot routing to work, each remote site must connect to the same access router every time it dials into the central site. In this design, the remote routers connect to the central-site access routers through a hunt group, so there is no way to control to which central-site access router a remote router will connect for any particular connection. Therefore, snapshot routing cannot be used for this design.

Configuring the Central Site Access Routers

This section describes how the configuration of the central-site access routers implements authentication, network layer addressing, and the routing strategy. The configuration for each central-site access router is the same with the following exceptions:

- The IP address specified for loopback interface 0
- The IP address specified for Ethernet interface 0
- The name of the router as specified by the **hostname** global configuration command

This discussion is divided among the following topics:

- Username Configuration for the Remote Sites
- Dial-Up Configuration for the Remote Sites
- Asynchronous Line Configuration
- OSPF Routing Configuration
- RIP Routing Configuration
- Static Routing Configuration
- Security Issues
- Configuration File Size

For the complete configuration see the section “CENTRAL-1 Configuration” later in this chapter.

Username Configuration for the Remote Sites

The configuration of each central-site access router includes the following **username** global configuration commands:

```
username Router2 password 7 071C2D4359
...
username Router121 password 7 0448070918
```

Each remote router can dial in to any of the three central-site access routers, so there is a **username** global configuration command for each remote router. When a remote router logs in, it specifies a name (for example, Router2) and a password (for example, outthere) that must match the values specified by a **username** command. Each remote site uses a chat script to log in and specify its host name (which must match a value specified by the **username** command) and password. (For information about the chat script that the remote sites use, see the section "Chat Script Configuration for Dialing the Central Site" later in this chapter.)

Dial-Up Configuration for the Remote Sites

The configuration of each central-site access router includes the following **chat-script** global configuration commands:

```
chat-script CALL1020 ABORT ERROR ABORT BUSY TIMEOUT 30 "" "ATDT\T" "CONNECT" \c
chat-script REM TIMEOUT 40 "name:" "CENTRAL" "word:" "secret"
chat-script USRV32BIS "" "AT&F1S0=1&d2" "OK" ""
!
interface dialer 20
dialer map ip 172.16.2.1 name Router2 modem-script CALL1020 system-script REM 5551234
...
dialer map ip 172.16.121.1 name Router2 modem-script CALL1020 system-script REM 5555678
!
line 1 16
script reset USRV32BIS
```

The three **chat-script** global configuration commands establish three scripts named CALL1020, REM, and USRV32BIS. CALL1020 and REM are invoked by the **dialer map** commands to dial and log in to the remote sites, respectively. The **script reset** command specifies that the USRV32BIS script is to be run whenever an asynchronous line is reset in order to ensure that the central-site modems are always configured correctly.

Loopback Interface Configuration

The configuration of each central-site access router includes the commands for configuring loopback interfaces. The IP address for loopback interface 0 is unique for each access router and, to satisfy the rules by which OSPF selects the router ID, must be the highest loopback IP address on the router. The IP address for loopback interface 1 is the same for each central-site access router. The commands are as follows:

```
interface loopback 0
ip address 172.16.254.3 255.255.255.255
...
interface loopback 1
ip address 172.16.1.1 255.255.255.0
```

The goal is for all three access routers to appear to have the same IP address during IPCP negotiation with the remote sites. (IPCP is the part of PPP that brings up and configures IP support.) This goal is accomplished by creating a loopback interface, assigning to it the same IP address on each central-site access router, and running the **ip unnumbered** interface configuration command using the loopback interface address. The problem with this strategy is that OSPF takes its router ID from the IP address of a loopback interface, if one is configured, which would mean that all three access routers would have the same OSPF router ID.

The solution is to create loopback interface 0 and assign to it a unique IP address (which results in a unique OSPF router ID for each router). The configuration then creates loopback interface 1 and assigns to it the same IP address on each router. Loopback interface 1 allows the **ip unnumbered** command to be applied to dialer rotary group 20 later in the configuration.

Asynchronous Line Configuration

The configuration of each central-site access router includes the following commands for configuring each asynchronous interface:

```
interface async 1
ip unnumbered loopback 1
async dynamic address
async dynamic routing
async mode interactive
dialer in-band
dialer rotary-group 20
```

For each of the 16 asynchronous interfaces provided by the access router, the configuration uses the **ip unnumbered** interface configuration command to specify that the asynchronous interface is to use the IP address of loopback interface 1 as the source address for any IP packets that the asynchronous interface generates. The IP address of loopback interface 1 is also used to determine which routing processes are sending updates over the asynchronous interface.

The **async dynamic address** interface configuration command enables dynamic addressing on the asynchronous interface. This command is required to allow each remote router to specify its IP address when it logs in. The **async dynamic routing** interface configuration command allows the interface to run a routing protocol, in this case RIP.

The **async mode interactive** interface configuration command allows a remote router to dial in and access the EXEC command interface, which allows the remote router to start PPP and specify its IP address.

The **dialer in-band** interface configuration command allows chat scripts to be used on the asynchronous interface. The chat scripts allow the access router to dial the remote sites. The **dialer rotary-group** interface configuration command assigns each asynchronous interface to dialer rotary group 20.

Dialer Interface Configuration

The configuration of each central-site access router includes the following commands for configuring dialer rotary group 20:

```
interface dialer 20
ip unnumbered loopback 1
encapsulation ppp
dialer in-band
dialer idle-timeout 60
dialer map ip 172.16.2.1 name Router2 modem-script CALL1020 system-script REM 5551234
...
dialer map ip 172.16.121.1 name Router121 modem-script CALL1020 system-script REM
5555678
dialer-group 3
dialer-list 3 list 101
access-list 101 deny udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 520
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

The **interface dialer** global configuration command defines dialer rotary group 20. Any interface configuration commands that are applied to a dialer rotary group apply to the physical interfaces that are its members. When the router's configuration includes multiple destinations, any of the interfaces in the dialer rotary group can be used to place outgoing calls.

The **ip unnumbered** interface configuration command specifies that the IP address of loopback interface 1 is to be used as the source address for any IP packets that dialer rotary group 20 might generate. The **dialer idle-timeout** interface configuration command causes a disconnect if 60 seconds elapses without any interesting traffic.

The configuration includes a **dialer map** interface configuration command for each remote router that the central-site access router might dial. The **ip** keyword specifies that the dialer map is to be used for IP packets, the IP address is the next-hop address of the destination that is to be called, and the **name** keyword specifies the host name of the remote router that is to be called. The **modem-script** keyword specifies that the CALL1020 chat script is to be used, and the **system-script** keyword specifies that the REM chat script is to be used. The last value specified by the **dialer map** command is the telephone number for the remote router. The dialer map commands do not specify the **broadcast** keyword, so RIP updates are not sent to the remote sites.

For the Dialer20 interface, the **dialer-group** interface configuration command defines *interesting* packets to be those packets defined by the corresponding **dial-list** command. Interesting packets cause a call to be made or cause a call to be maintained. In this case, access list 101 defines RIP as uninteresting. (RIP uses User Datagram Protocol [UDP] port 520.) All other packets are defined as interesting.

OSPF Routing Configuration

Each central-site access router uses the following commands to configure OSPF. These commands limit the routes that are redistributed into OSPF to the major Class B static route and any dynamic subnet routes that may exist for currently connected remote sites. Limiting the routes that are redistributed into OSPF simplifies the routing table on the Cisco 4500 significantly.

```
router ospf 110
redistribute static subnets route-map STATIC-TO-OSPF
redistribute rip subnets route-map RIP-TO-OSPF
passive-interface async 1
...
passive-interface async 16
network 172.19.0.0 0.0.255.255 area 0
distance 210
!
route-map RIP-TO-OSPF permit
match ip address 20
!
access-list 20 permit 172.16.0.0 0.0.255.0
!
route-map STATIC-TO-OSPF permit
match ip address 21
!
access-list 21 permit 172.16.0.0
```

The **router ospf** global configuration command enables an OSPF routing process and assigns to it a process ID of 110.

The first **redistribute** router configuration command causes static IP routes to be redistributed into OSPF. The **subnets** keyword specifies that subnets are to be redistributed, and the **route-map** keyword specifies that only those routes that successfully pass through the route map named STATIC-TO-OSPF are to be redistributed. The STATIC-TO-OSPF route map permits the redistribution of routes that match access list 21. Access list 21 permits only major network 172.16.0.0.

The second **redistribute** router configuration command causes RIP routes to be redistributed into OSPF. The **subnets** keyword specifies that subnets are to be redistributed, and the **route-map** keyword specifies that only those routes that successfully pass through the route map named RIP-TO-OSPF are to be redistributed. The RIP-TO-OSPF route map permits the redistribution of routes that match access list 20. Access list 20 permits only routes that start with 172.16 and end with .0 (the third octet is wild). In effect, the RIP-TO-OSPF route map allows only subnets that match 172.16.x.0.

For each asynchronous interface, there is a **passive-interface** router configuration command, which means that OSPF routing information is neither sent nor received through the asynchronous interfaces. The **distance** router configuration command assigns the OSPF routing process an administrative distance of 210. This allows the central-site access routers to prefer their static routes (with an administrative distance of 200) over routes learned by OSPF.

Note When a remote site logs in and a dynamic route is established for it, the other access routers retain their static routes for that remote site. When a remote site logs out, the other access routers do not need to update their routing tables—their routing tables still contain the static routes that are necessary for dialing out to the remote site.

RIP Routing Configuration

Each access router uses the following commands to configure RIP:

```
router rip
timers basic 30 35 0 1
network 172.16.0.0
distribute-list 10 out Dialer20
!
access-list 10 deny 0.0.0.0 255.255.255.255
```

The **timers basic** router configuration adjusts the RIP update, invalid, holddown, and flush timers. The command specifies that RIP updates are to be sent every 30 seconds, that a route is to be declared invalid if an update for the route is not received within 35 seconds after the previous update, that the time during which better routes are to be suppressed is 0 seconds, and that one second must pass before an invalid route is removed from the routing table. These timer adjustments produce the fastest possible convergence when a remote site logs out.

The **network** router configuration command specifies that network 172.16.0.0 is to participate in the RIP routing process. There is no need to propagate RIP routes to the Cisco 1020s, so the **distribute-list out** router configuration command specifies that access list 10 is to be used to control the advertisement of networks in updates. Access list 10 prevents RIP routes from being sent to the remote site.

Static Routing Configuration

The configuration of each central-site access router includes the following commands for configuring static routes to the remote sites:

```
ip route 172.16.0.0 255.255.0.0 Dialer20
```

The first **ip route** global configuration command creates a static route for major network 172.16.0.0 and assigns it to the dialer interface 20. The route, when distributed into OSPF, tells the Cisco 4500 that this central-site access router can get to the remote sites. If the access router goes down, the Cisco 4500 learns that the route is not longer available and removes it from its routing table. This route is redistributed into OSPF by the STATIC-TO-OSPF filter. The first **ip route** command is followed by pairs of static routes, one pair for each remote site:

```
ip route 172.16.2.0 255.255.255.0 172.16.2.1 200
ip route 172.16.2.1 255.255.255.255 Dialer20
...
ip route 172.16.121.0 255.255.255.0 172.16.121.1 200
ip route 172.16.121.1 255.255.255.255 Dialer20
```

In unnumbered IP environments, two static routes are required for each remote site:

- One static route points to the next hop on the dialer map. Note that the “200” makes this route a floating static route, but that it is lower than OSPF routes (which are set to 210 by the **distance** command, earlier in the configuration). This means that a RIP route triggered by a connection to a remote site (whether the connection is initiated by the remote site or the central site) will override the static route. An OSPF update initiated by a remote site that dials in will not override a static route that points to the next hop address on the dialer map.
- One static route that defines the interface at which the next hop can be found (in this case, dialer interface 20). This static route is required for unnumbered interfaces. Note there is no need to make this a floating static route.

Security Issues

The configuration for each central-site access router includes the **login** line configuration command for each asynchronous line and specifies the **local** keyword. This command causes the access router to match the username and password specified by the **username** global configuration command against the username and password that the remote site specifies when it logs in. This security method is required to allow the remote sites to log in and specify their IP addresses.

Configuration File Size

As the number of remote sites increases, the size of the configuration file for each central-site access router might increase to a size at which it can no longer be stored in NVRAM. There are two ways to alleviate this problem:

- Compress the configuration file using the **service compress-config** global configuration command.
- Have the central-site access routers boot using configuration files stored on a Trivial File Transfer Protocol (TFTP) server.

Configuring the Remote Site Routers

With the exception of the host name and the IP address of the Ethernet interface of each remote site router, the configuration of each remote site router is the same. The discussion of the configuration is divided among the following topics:

- Chat Script Configuration for Dialing the Central Site
- Configuring the Asynchronous Interface
- Using the **site** Command
- Static Routing Configuration

For the complete configuration, see the section “Router2 Configuration” later in this chapter.

Chat Script Configuration for Dialing the Central Site

The configuration of each remote router includes the following **chat-script** global configuration commands:

```
chat-script CENTRALDIAL "" "ATDT 5551111" "CONNECT" "" "name:" "Router2" "word:"
"outthere" ">" "ppp 172.16.2.1"
```

The **chat-script** command defines a chat script named CENTRALDIAL that is used to place calls to the central site. The CENTRALDIAL chat script specifies the telephone number (555-1111) of the central site and the expect-send sequences that guide the modem through the dial-up process. A key feature of the chat script is that when the remote router receives the string > (the prompt indicating that the remote site router has successfully logged in to a central-site access router), the remote router sends the EXEC command **ppp 172.16.2.1**, which informs the central-site access router of the remote router's IP address.

Configuring the Asynchronous Interface

The configuration of each remote router includes the following commands that configure the asynchronous interface:

```
interface async 1
speed 38400
modem-type usr-sport-v32
dialer rotary-group 1
!
modem-def usr-sport-v32 "USR Sportster v.32bis" 38400  "" "AT&F1" "OK"
```

The **speed** line configuration command sets the baud rate to 38400 bits per second for both sending and receiving. The **modem-type** command specifies the initialization string sent to the modem when the interface is reset or when a **clear interface async** command is issued. The initialization string is defined by the **modem-def** command for usr-sport-v32. The **dialer rotary-group** interface configuration command assigns asynchronous interface 1 to dialer rotary group 1.

Using the Site Command

The configuration of each remote router includes the following **site** configuration commands:

```
site CENTRAL
dial-on demand
encapsulation ppp
ip address 172.16.1.1 255.255.255.0
routing rip broadcast
dialgroup 1
session-timeout 5
system-script CENTRALDIAL
password secret
max-ports 1
```

The **site** global configuration command defines a remote location that the router can dial in to or that can dial in to this router, or both, and names it CENTRAL. The name is used to authenticate the central site when it dials in.

The **dial-on** site configuration command uses the **demand** keyword to specify that the central site is to be dialed and a connection established only when packets are queued for the central site. The **encapsulation** site configuration command specifies that when the router establishes a connection with the central site, it is to use PPP encapsulation.

The **ip address** interface configuration command associates IP address 172.16.1.1 with the CENTRAL site. Note that IP address 172.16.1.1 is the address of the dialer 20 interface on each of the central-site access routers. The **routing rip** interface configuration command and the **broadcast** keyword specify that when the router is connected to the central site, IP routing updates are to be broadcast, but any incoming IP routing updates are to be ignored.

The **dialgroup** command specifies that dial group 1 is to be used when connecting to the central site. Earlier in the configuration, the **dialer rotary-group** command assigned asynchronous interface 1 to group 1.

The **session-timeout** site configuration command specifies that if a period of five minutes elapses during which there is no input or output traffic, the router is to close the connection. The **system-script** site configuration command specifies that the CENTRALDIAL chat script is to be used to dial the central site. The **password** site configuration command specifies that when a central-site access router logs in, its password must be the string "secret."

Static Routing Configuration

The configuration of each remote router includes the following **ip route** global configuration commands:

```
ip route 150.10.0.0 172.16.1.1 1
ip route 172.18.0.0 172.16.1.1 1
ip route 172.19.0.0 172.16.1.1 1
ip route 172.21.0.0 172.16.1.1 1
ip route 172.22.0.0 172.16.1.1 1
```

The **ip route** commands establish static IP routes for networks located at the central site, all reachable through a next-hop address of 172.16.1.1, which is the IP address shared by all of the access routers at the central site. All **ip route** commands specify an administrative distance of 1, which is the default.

The Complete Configurations

This section contains the complete configurations for CENTRAL-1 and Router2.

CENTRAL-1 Configuration

The complete configuration for CENTRAL-1 follows. Those portions of the configuration that must be unique to each central-site access router are highlighted in bold.

```
!
version 10.2
service timestamps debug datetime
service timestamps log datetime
service udp-small-servers
service tcp-small-servers
!
hostname CENTRAL-1
!
enable-password as5100
!
username Router2 password 7 071C2D4359
...
username Router121 password 7 0448070918
!
chat-script CALL1020 ABORT ERROR ABORT BUSY TIMEOUT 30 "" "ATDT\T" "CONNECT" \c
chat-script REM TIMEOUT 40 "name:" "CENTRAL" "word:" "secret"
chat-script USRV32BIS "" "AT&F1S0=1&d2" "OK" ""
!
interface loopback 0
ip address 172.16.254.3 255.255.255.255
!
interface loopback 1
ip address 172.16.1.1 255.255.255.0
!
interface ethernet 0
ip address 172.19.1.8 255.255.0.0
!
interface serial 0
no ip address
shutdown
!
interface async 1
ip unnumbered loopback 1
encapsulation ppp
async dynamic address
async dynamic routing
async mode interactive
dialer in-band
dialer idle-timeout 60
dialer rotary-group 20
...
interface async 16
ip unnumbered loopback 1
encapsulation ppp
async dynamic address
async dynamic routing
async mode interactive
dialer in-band
dialer idle-timeout 60
dialer rotary-group 20
!
interface dialer 20
ip unnumbered loopback 1
encapsulation ppp
dialer in-band
dialer idle-timeout 60
dialer fast-idle 60
dialer map ip 172.16.2.1 name Router2 modem-script CALL1020 system-script REM 5551234
...
```

```
dialer map ip 172.16.121.1 name Router121 modem-script CALL1020 system-script REM
5555678
dialer-group 3
!
router ospf 110
redistribute static subnets route-map STATIC-TO-OSPF
redistribute rip subnets route-map RIP-TO-OSPF
passive-interface async 1
...
passive-interface async 16
network 172.19.0.0 0.0.255.255 area 0
distance 210
!
router rip
timers basic 30 35 0 1
network 172.16.0.0
distribute-list 10 out Dialer20
!
ip default-gateway 172.19.1.10
!
ip route 172.16.0.0 255.255.0.0 Dialer20
ip route 172.16.2.0 255.255.255.0 172.16.2.1 200
ip route 172.16.2.1 255.255.255.255 Dialer20
...
ip route 172.16.121.0 255.255.255.0 172.16.121.1 200
ip route 172.16.121.1 255.255.255.255 Dialer20

access-list 10 deny 0.0.0.0 255.255.255.255
access-list 20 permit 172.16.0.0 0.0.255.0
access-list 21 permit 172.16.0.0
access-list 101 deny udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 520
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255

route-map RIP-TO-OSPF permit
match ip address 20
!
route-map STATIC-TO-OSPF permit
match ip address 21
!
snmp-server community public RO
snmp-server community private RW
dialer-list 3 list 101
!
line con 0
line 1 16
login local
modem inout
script reset USRV32BIS
transport input all
rxspeed 38400
txspeed 38400
flowcontrol hardware
line aux 0
transport input all
line vty 0 4
exec-timeout 20 0
password cisco
login
!
end
```

Router2 Configuration

The complete configuration for Router2 follows. Those portions of the configuration that must be unique to each remote site router are highlighted in bold.

```
version 1.1(2)
!
hostname Router2
!
enable-password cisco-a
!
chat-script CENTRALDIAL "" "ATDT 5551111" "CONNECT" "" "name:" "Router2" "word:"
"outthere" ">" "ppp 172.16.2.1"
!
interface ethernet 0
ip address 172.16.2.1 255.255.255.0
!
interface async 1
speed 38400
modem-type usr-sport-v32
dialer rotary-group 1
!
site CENTRAL
dial-on demand
encapsulation ppp
ip address 172.16.1.1 255.255.255.0
routing rip broadcast
dialgroup 1
session-timeout 5
system-script CENTRALDIAL
password secret
max-ports 1
!
modem-def usr-sport-v32 "USR Sportster v.32bis" 38400  "" "AT&F1" "OK"
!
ip route 150.10.0.0 172.16.1.1 1
ip route 172.18.0.0 172.16.1.1 1
ip route 172.19.0.0 172.16.1.1 1
ip route 172.21.0.0 172.16.1.1 1
ip route 172.22.0.0 172.16.1.1 1
```

Summary

This case study shows that it is possible to scale dial-on-demand routing to accommodate large dial-up networks. If, in the future, the number of remote sites exceeds the capacity of the 48 asynchronous interfaces, additional routers can be installed without modifying the routing strategy. Although this case study focuses on asynchronous media, many of the techniques can be applied to other dial-up technologies, such as ISDN.

CHAPTER 12

Using the Border Gateway Protocol for Interdomain Routing

The Border Gateway Protocol (BGP), defined in RFC 1771, provides loop-free interdomain routing between autonomous systems. (An autonomous system [AS] is a set of routers that operate under the same administration.) BGP is often run among the networks of Internet service providers (ISPs). This case study examines how BGP works and how you can use it to participate in routing with other networks that run BGP. The following topics are covered:

- BGP Fundamentals
- BGP Decision Algorithm
- Controlling the Flow of BGP Updates
- Practical Design Example

Note The version of BGP described in this case study is BGP Version 4.

BGP Fundamentals

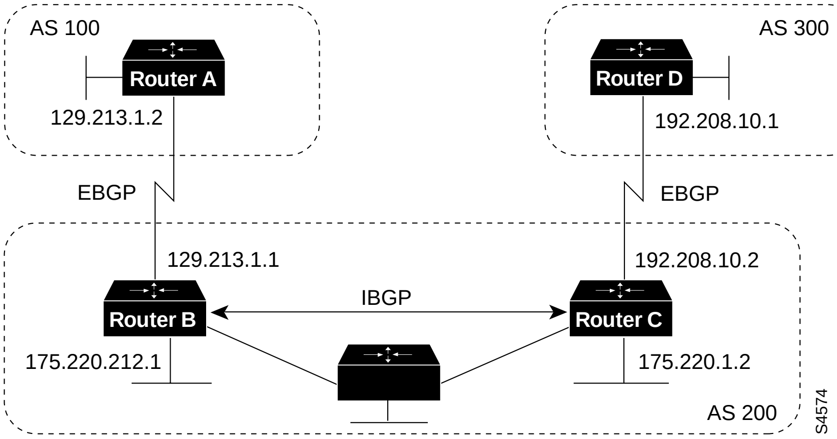
This section presents fundamental information about BGP, including the following topics:

- Internal BGP
- External BGP
- BGP and Route Maps
- Advertising Networks

Routers that belong to the same AS and exchange BGP updates are said to be running *internal BGP* (IBGP), and routers that belong to different ASs and exchange BGP updates are said to be running *external BGP* (EBGP). With the exception of the **neighbor ebgp-multihop** router configuration command (described in the section “External BGP” later in this chapter), the commands for configuring EBGP and IBGP are the same. This case study uses the terms EBGP and IBGP as a reminder that, for any particular context, routing updates are being exchanged between ASs (EBGP) or within an AS (IBGP).

Figure 12-1 shows a network that demonstrates the difference between EBGP and IBGP.

Figure 12-1 EBGP, IBGP, and Multiple ASs



Before it exchanges information with an external AS, BGP ensures that networks within the AS are reachable. This is done by a combination of internal BGP peering among routers within the AS and by redistributing BGP routing information to Interior Gateway Protocols (IGPs) that run within the AS, such as Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), Routing Information Protocol (RIP), and Open Shortest Path First (OSPF).

BGP uses the Transmission Control Protocol (TCP) as its transport protocol (specifically port 179). Any two routers that have opened a TCP connection to each other for the purpose of exchanging routing information are known as *peers* or *neighbors*. In Figure 12-1, Routers A and B are BGP peers, as are Routers B and C, and Routers C and D. The routing information consists of a series of AS numbers that describe the full path to the destination network. BGP uses this information to construct a loop-free map of ASs. Note that within an AS, BGP peers do not have to be directly connected.

BGP peers initially exchange their full BGP routing tables. Thereafter, BGP peers send incremental updates only. BGP peers also exchange keepalive messages (to ensure that the connection is up) and notification messages (in response to errors or special conditions).

In Figure 12-1, the following commands configure BGP on Router A:

```
router bgp 100
neighbor 129.213.1.1 remote-as 200
```

The following commands configure BGP on Router B:

```
router bgp 200
neighbor 129.213.1.2 remote-as 100
neighbor 175.220.1.2 remote-as 200
```

The following commands configure BGP on Router C:

```
router bgp 200
neighbor 175.220.212.1 remote-as 200
neighbor 192.208.10.1 remote-as 300
```

The following commands configure BGP on Router D:

```
router bgp 300
neighbor 192.208.10.2 remote-as 200
```

The **router bgp** global configuration command enables a BGP routing process and assigns to it an AS number.

The **neighbor remote-as** router configuration command adds an entry to the BGP neighbor table specifying that the peer identified by a particular IP address belongs to the specified AS. For routers that run EBGP, neighbors are usually directly connected, and the IP address is usually the IP address of the interface at the other end of the connection. (For the exception to this rule, see the section "EBGP Multihop," later in this chapter.) For routers that run IBGP, the IP address can be the IP address of any of the router's interfaces.

Note the following about the ASs shown in Figure 12-1:

- Routers A and B are running EBGP, and Routers B and C are running IBGP. Note that the EBGP peers are directly connected and that the IBGP peers are not. As long as there is an IGP running that allows the two neighbors to reach one another, IBGP peers do not have to be directly connected.
- All BGP speakers within an AS must establish a peer relationship with each other. That is, the BGP speakers within an AS must be fully meshed logically. BGP4 provides two techniques that alleviate the requirement for a logical full mesh: confederations and route reflectors. For information about these techniques, see the sections "Confederations" and "Route Reflectors," later in this chapter.
- AS 200 is a *transit AS* for AS 100 and AS 300—that is, AS 200 is used to transfer packets between AS 100 and AS 300.

To verify that BGP peers are up, use the **show ip bgp neighbors** EXEC command. Following is the output of this command on Router A:

```
RouterA# show ip bgp neighbors
BGP neighbor is 129.213.1.1, remote AS 200, external link
 BGP version 4, remote router ID 175.220.212.1
 BGP state = established, table version = 3, up for 0:10:59
 Last read 0:00:29, hold time is 180, keepalive interval is 60 seconds
 Minimum time between advertisement runs is 30 seconds
 Received 2828 messages, 0 notifications, 0 in queue
 Sent 2826 messages, 0 notifications, 0 in queue
 Connections established 11; dropped 10
```

Anything other than state = established indicates that the peers are not up. The remote router ID is the highest IP address on that router (or the highest loopback interface, if there is one). Notice the table version number: each time the table is updated by new incoming information, the table version number increments. A table version number that continually increments is an indication that a route is flapping, thereby causing routes to be updated continually.

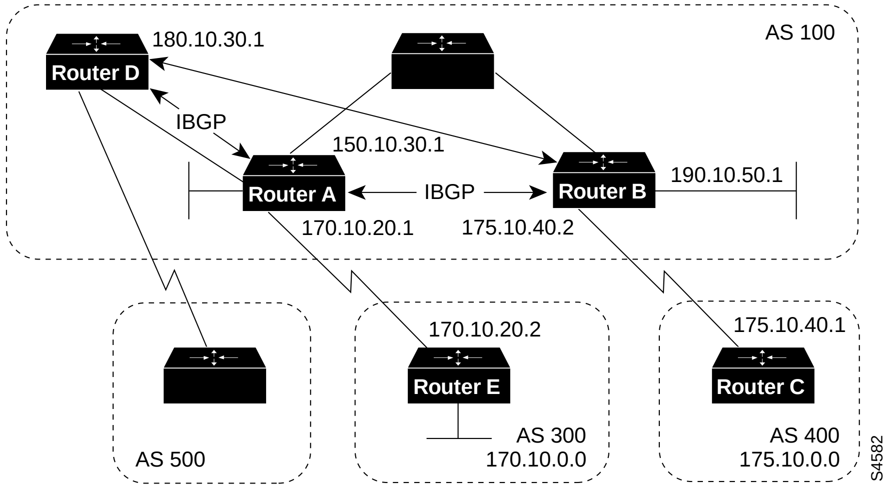
Note When you make a configuration change with respect to a neighbor for which a peer relationship has been established, be sure to reset the BGP session with that neighbor. To reset the session, at the system prompt, issue the **clear ip bgp** EXEC command specifying the IP address of that neighbor.

Internal BGP

Internal BGP (IBGP) is the form of BGP that exchanges BGP updates within an AS. Instead of IBGP, the routes learned via EBGP could be redistributed into IGP within the AS and then redistributed again into another AS. However, IBGP is more flexible, provides more efficient ways of controlling the exchange of information within the AS, and presents a consistent view of the AS to external neighbors. For example, IBGP provides ways to control the exit point from an AS.

Figure 12-2 shows a topology that demonstrates IBGP.

Figure 12-2 Internal BGP Example



The following commands configure Routers A and B in AS 100, and Router C in AS 400:

```
!Router A
router bgp 100
neighbor 180.10.30.1 remote-as 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0

!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 175.10.40.1 remote-as 400
neighbor 180.10.30.1 remote-as 100
network 190.10.50.0

!Router C
router bgp 400
neighbor 175.10.40.2 remote-as 100
network 175.10.0.0

!Router D
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 190.10.50.1 remote as 100
network 190.10.0.0
```

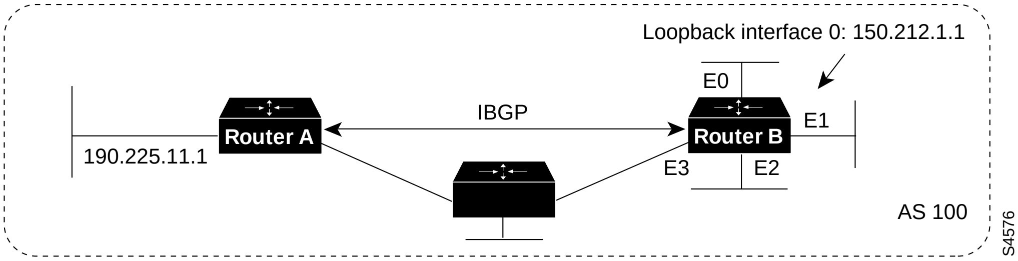
When a BGP speaker receives an update from other BGP speakers in its own AS (that is, via IBGP), the receiving BGP speaker uses EBGP to forward the update to external BGP speakers only. This behavior of IBGP is why it is necessary for BGP speakers within an AS to be fully meshed.

For example, in Figure 12-2, if there were no IBGP session between Routers B and D, Router A would send updates from Router B to Router E but not to Router D. If you want Router D to receive updates from Router B, Router B must be configured so that Router D is a BGP peer.

Loopback Interfaces

Loopback interfaces are often used by IBGP peers. The advantage of using loopback interfaces is that they eliminate a dependency that would otherwise occur when you use the IP address of a physical interface to configure BGP. Figure 12-3 shows a network in which using the loopback interface is advantageous.

Figure 12-3 Use of Loopback Interfaces



In Figure 12-3, Routers A and B are running IBGP within AS 100. If Router A were to specify the IP address of Ethernet interface 0, 1, 2, or 3 in the **neighbor remote-as** router configuration command, and if the specified interface were to become unavailable, Router A would not be able to establish a TCP connection with Router B. Instead, Router A specifies the IP address of the loopback interface that Router B defines. When the loopback interface is used, BGP does not have to rely on the availability of a particular interface for making TCP connections.

The following commands configure Router A for BGP:

```
!Router A
router bgp 100
neighbor 150.212.1.1 remote-as 100
```

The following commands configure Router B for BGP:

```
!Router B
loopback interface 0
ip address 150.212.1.1 255.255.0.0
!
router bgp 100
neighbor 190.225.11.1 remote-as 100
neighbor 190.225.11.1 update-source loopback 0
```

Router A specifies the IP address of the loopback interface (150.212.1.1) of Router B in the **neighbor remote-as** router configuration command. This use of the loopback interface requires that the configuration of Router B include the **neighbor update-source** router configuration command. When the **neighbor update-source** command is used, the source of BGP TCP connections for the specified neighbor is the IP address of the loopback interface instead of the IP address of a physical interface.

Note Loopback interfaces are rarely between EBGP peers because EBGP peers are usually directly connected and, therefore, depend on a particular physical interface for connectivity.

External BGP

When two BGP speakers that are not in the same AS run BGP to exchange routing information, they are said to be running EBGP. This section describes commands that solve configuration problems that arise when BGP routing updates are exchanged between different ASs:

- EBGP Multihop
- EBGP Load Balancing
- Synchronization

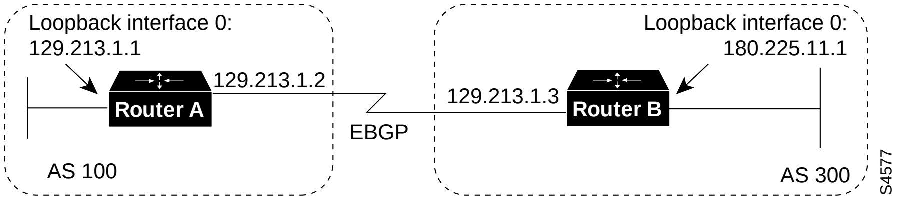
EBGP Multihop

Usually, the two EBGP speakers are directly connected (for example, over a wide-area network [WAN] connection). Sometimes, however, they cannot be directly connected. In this special case, the **neighbor ebgp-multihop** router configuration command is used.

Note Multihop is used only for EBGP, but not for IBGP.

Figure 12-4 illustrates a topology in which the **neighbor ebgp-multihop** command is useful.

Figure 12-4 EBGP Multihop



The following commands configure Router A to run EBGP:

```
!Router A
loopback interface 0
ip address 129.213.1.1
!
router bgp 100
neighbor 180.225.11.1 remote-as 300
neighbor 180.225.11.1 ebgp-multihop
neighbor 180.225.11.1 update-source loopback 0
```

The **neighbor remote-as** router configuration command specifies the IP address of an interface that is an extra hop away (180.225.11.1 instead of 129.213.1.3), and the **neighbor ebgp-multihop** router configuration command enables EGBP multihop. Because Router A references an external neighbor by an address that is not directly connected, its configuration must include static routes or must enable an IGP so that the neighbors can reach each other.

The following commands configure Router B:

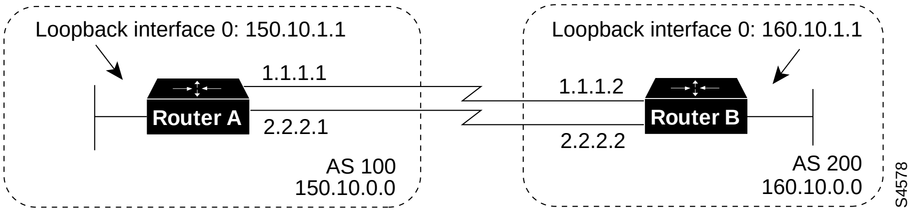
```
!Router B
loopback interface 0
ip address 180.225.11.1
```

```
router bgp 300
neighbor 129.213.1.1 remote-as 100
neighbor 129.213.1.1 ebgp-multihop
neighbor 129.213.1.1 update-source loopback 0
```

EBGP Load Balancing

The **neighbor ebgp-multihop** router configuration command and loopback interfaces are also useful for configuring load balancing between two ASs over parallel serial lines, as shown in Figure 12-5.

Figure 12-5 Load Balancing over Parallel Serial Lines



Without the **neighbor ebgp-multihop** command on each router, BGP would not perform load balancing in Figure 12-5, but with the **neighbor ebgp-multihop** command on each router, BGP uses both serial lines. The following commands configure load balancing for Router A:

```
!Router A
interface loopback 0
ip address 150.10.1.1 255.255.255.0
!
router bgp 100
neighbor 160.10.1.1 remote-as 200
neighbor 160.10.1.1 ebgp-multihop
neighbor 160.10.1.1 update-source loopback 0
network 150.10.0.0
!
ip route 160.10.0.0 255.255.0.0 1.1.1.2
ip route 160.10.0.0 255.255.0.0 2.2.2.2
```

The following commands configure load balancing for Router B:

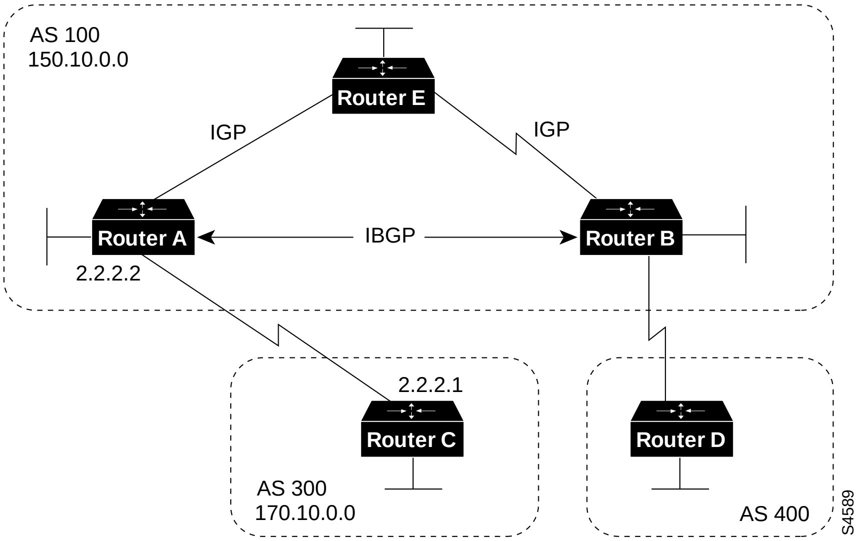
```
!Router B
interface loopback 0
ip address 160.10.1.1 255.255.255.0
!
router bgp 200
neighbor 150.10.1.1 remote-as 100
neighbor 150.10.1.1 ebgp-multihop
neighbor 150.10.1.1 update-source loopback 0
network 160.10.0.0
!
ip route 150.10.0.0 255.255.0.0 1.1.1.1
ip route 150.10.0.0 255.255.0.0 2.2.2.1
```

The **neighbor ebgp-multihop** and **neighbor update-source** router configuration commands have the effect of making the loopback interface the next hop for EBGP, which allows load balancing to occur. Static routes are used to introduce two equal-cost paths to the destination. (The same effect could also be accomplished by using an IGP.) Router A can reach the next hop of 160.10.1.1 in two ways: via 1.1.1.2 and via 2.2.2.2. Likewise, Router B can reach the next hop of 150.10.1.1 in two ways: via 1.1.1.1 and via 2.2.2.1.

Synchronization

When an AS provides transit service to other ASs and if there are non-BGP routers in the AS, transit traffic might be dropped if the intermediate non-BGP routers have not learned routes for that traffic via an IGP. The BGP synchronization rule states that if an AS provides transit service to another AS, BGP should not advertise a route until all of the routers within the AS have learned about the route via an IGP. The topology shown in Figure 12-6 demonstrates the synchronization rule.

Figure 12-6 Synchronization



In Figure 12-6, Router C sends updates about network 170.10.0.0 to Router A. Routers A and B are running IBGP, so Router B receives updates about network 170.10.0.0 via IBGP. If Router B wants to reach network 170.10.0.0, it sends traffic to Router E. If Router A does not redistribute network 170.10.0.0 into an IGP, Router E has no way of knowing that network 170.10.0.0 exists and will drop the packets.

If Router B advertises to AS 400 that it can reach 170.10.0.0 before Router E learns about the network via IGP, traffic coming from Router D to Router B with a destination of 170.10.0.0 will flow to Router E and be dropped.

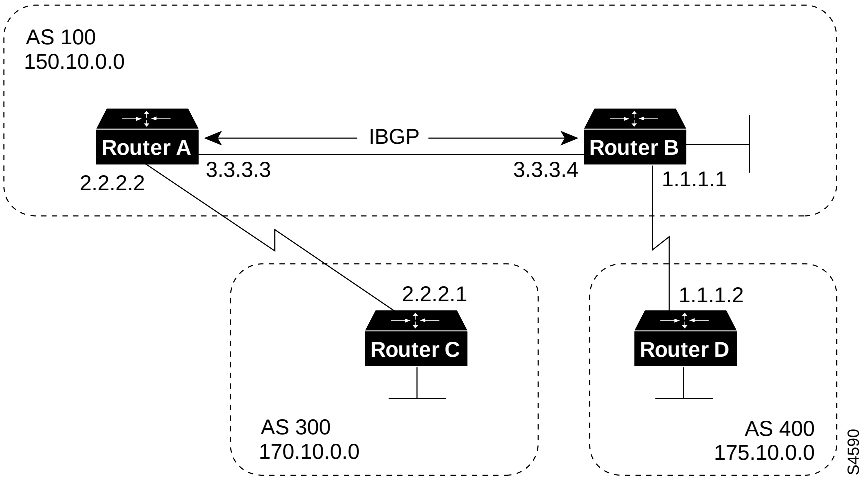
This situation is handled by the synchronization rule of BGP, which states that if an AS (such as AS 100 in Figure 12-6) passes traffic from one AS to another AS, BGP does not advertise a route before all routers within the AS (in this case, AS 100) have learned about the route via an IGP. In this case, Router B waits to hear about network 170.10.0.0 via an IGP before it sends an update to Router D. In some cases, you might want to disable synchronization. Disabling synchronization allows BGP to converge more quickly, but it might result in dropped transit packets.

You can disable synchronization if one of the following conditions is true:

- Your AS does not pass traffic from one AS to another AS.
- All the transit routers in your AS run BGP.

Figure 12-7 shows a topology in which it is desirable to disable synchronization.

Figure 12-7 Disabled Synchronization



The following commands configure Routers A, B, and C:

```
!Router A
network 150.10.0.0
neighbor 3.3.3.4 remote-as 100
neighbor 2.2.2.1 remote-as 300
no synchronization

!Router B
router bgp 100
network 150.10.0.0
neighbor 1.1.1.2 remote-as 400
neighbor 3.3.3.3 remote-as 100
no synchronization

!Router D
router bgp 400
neighbor 1.1.1.1 remote-as 100
network 175.10.0.0
```

The **no synchronization** router configuration command causes Router B to put 170.10.0.0 in its IP routing table and advertise it to Router D without learning network 170.10.0.0 via an IGP.

BGP and Route Maps

Route maps are used with BGP to control and modify routing information and to define the conditions by which routes are redistributed between routing domains. The format of a route map is as follows:

```
route-map map-tag [[permit | deny] | [sequence-number]]
```

The map tag is a name that identifies the route map, and the sequence number indicates the position that an instance of the route map is to have in relation to other instances of the same route map. (Instances are ordered sequentially.)

For example, you might use the following commands to define a route map named MYMAP:

```
route-map MYMAP permit 10
! First set of conditions goes here.
route-map MYMAP permit 20
! Second set of conditions goes here.
```

When BGP applies MYMAP to routing updates, it applies the lowest instance first (in this case, instance 10). If the first set of conditions is not met, the second instance is applied, and so on, until either a set of conditions has been met, or there are no more sets of conditions to apply.

The **match** and **set** route map configuration commands are used to define the condition portion of a route map. The **match** command specifies a criteria that must be matched, and the **set** command specifies an action that is to be taken if the routing update meets the condition defined by the **match** command.

Following is an example of a simple route map:

```
route-map MYMAP permit 10
match ip address 1.1.1.1
set metric 5
```

When an update matches IP address 1.1.1.1, BGP sets the metric for the update to 5, sends the update (because of the **permit** keyword), and breaks out of the list of route-map instances.

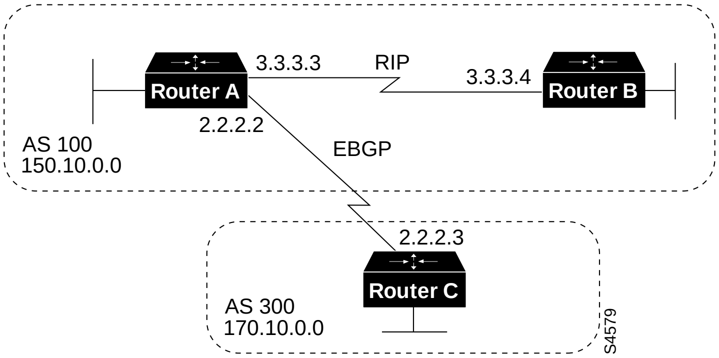
When an update does not meet the criteria of an instance, BGP applies the next instance of the route map to the update, and so on, until an action is taken, or there are no more route map instances to apply. If the update does not meet any criteria, the update is not redistributed or controlled.

When an update meets the match criteria, and the route map specifies the **deny** keyword, BGP breaks out of the list of instances, and the update is not redistributed or controlled.

Note Route maps cannot be used to filter incoming BGP updates based on IP address. You can, however, use route maps to filter outgoing BGP updates based on IP address.

Figure 12-8 shows a topology that demonstrates the use of route maps.

Figure 12-8 Route Map Example



In Figure 12-8, Routers A and B run RIP with each other, and Routers A and C run BGP with each other. If you want Router A to redistribute routes from 170.10.0.0 with a metric of 2 and to redistribute all other routes with a metric of 5, use the following commands for Router A:

```
!Router A
router rip
network 3.0.0.0
network 2.0.0.0
network 150.10.0.0
passive-interface serial 0
redistribute bgp 100 route-map SETMETRIC
!
router bgp 100
neighbor 2.2.2.3 remote-as 300
network 150.10.0.0
!
route-map SETMETRIC permit 10
match ip-address 1
set metric 2
!
route-map SETMETRIC permit 20
set metric 5
!
access-list 1 permit 170.10.0.0 0.0.255.255
```

When a route matches the IP address 170.10.0.0, it is redistributed with a metric of 2. When a route does not match the IP address 170.10.0.0, its metric is set to 5, and the route is redistributed.

Assume that on Router C you want to set to 300 the community attribute of outgoing updates for network 170.10.0.0. The following commands apply a route map to outgoing updates on Router C:

```
!Router C
router bgp 300
network 170.10.0.0
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map SETCOMMUNITY out
!
route-map SETCOMMUNITY permit 10
match ip address 1
set community 300
!
access-list 1 permit 0.0.0.0 255.255.255.255
```

Access list 1 denies any update for network 170.10.0.0 and permits updates for any other network.

Advertising Networks

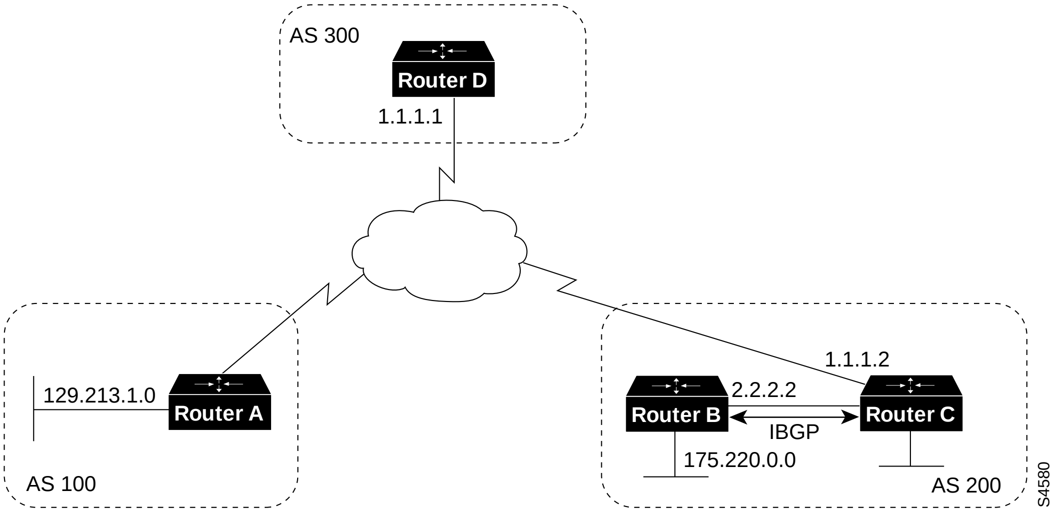
A network that resides within an AS is said to originate from that network. To inform other ASs about its networks, the AS advertises them. BGP provides three ways for an AS to advertise the networks that it originates:

- Redistributing Static Routes
- Redistributing Dynamic Routes
- Using the network Command

Note It is important to remember that routes advertised by the techniques described in this section are advertised *in addition* to other BGP routes that a BGP-configured router learns from its internal and external neighbors. BGP always passes on information that it learns from one peer to other peers. The difference is that routes generated by the **network** and **redistribute** router configuration commands specify the AS of the router as the originating AS for the network.

This section uses the topology shown in Figure 12-9 to demonstrate how networks that originate from an AS can be advertised.

Figure 12-9 Network Advertisement Example 1



Redistributing Static Routes

One way to advertise that a network or a subnet originates from an AS is to redistribute static routes into BGP. The only difference between advertising a static route and advertising a dynamic route is that when you redistribute a static route, BGP sets the origin attribute of updates for the route to Incomplete. (For a discussion of other values that can be assigned to the origin attribute, see the section “Origin Attribute,” later in this chapter.)

To configure Router C in Figure 12-9 to originate network 175.220.0.0 into BGP, use these commands:

```
!Router C
router bgp 200
neighbor 1.1.1.1 remote-as 300
redistribute static
!
ip route 175.220.0.0 0.0.255.255 null 0
```

The **redistribute** router configuration command and the **static** keyword cause all static routes to be redistributed into BGP.

The **ip route** global configuration command establishes a static route for network 175.220.0.0. In theory, the specification of the null 0 interface would cause a packet destined for network 175.220.0.0 to be discarded. In practice, there will be a more specific match for the packet than 175.220.0.0, and the router will send it out the appropriate interface. Redistributing a static route is the best way to advertise a supernet because it prevents the route from flapping.

Note Regardless of route type (static or dynamic), the **redistribute** router configuration command is the only way to inject BGP routes into an IGP.

Redistributing Dynamic Routes

Another way to advertise networks is to redistribute dynamic routes. Typically, you redistribute IGP routes (such as Enhanced IGRP, IGRP, IS-IS, OSPF, and RIP routes) into BGP. Some of your IGP routes might have been learned from BGP, so you need to use access lists to prevent the redistribution of routes back into BGP.

Assume that in Figure 12-9 Routers B and C are running IBGP, that Router C is learning 129.213.1.0 via BGP, and that Router B is redistributing 129.213.1.0 back into Enhanced IGRP. The following commands configure Router C:

```
!Router C
router eigrp 10
network 175.220.0.0
redistribute bgp 200
redistributed connected
default-metric 1000 100 250 100 1500
!
router bgp 200
neighbor 1.1.1.1 remote-as 300
neighbor 2.2.2.2 remote-as 200
neighbor 1.1.1.1 distribute-list 1 out
redistribute eigrp 10
!
access-list 1 permit 175.220.0.0 0.0.255.255
```

The **redistribute** router configuration command with the **eigrp** keyword redistributes Enhanced IGRP routes for process ID 10 into BGP. (Normally, distributing BGP into IGP should be avoided because too many routes would be injected into the AS.) The **neighbor distribute-list** router configuration command applies access list 1 to outgoing advertisements to the neighbor whose IP address is 1.1.1.1 (that is, Router D). Access list 1 specifies that network 175.220.0.0 is to be advertised. All other networks, such as network 129.213.1.0, are implicitly prevented from being advertised. The access list prevents network 129.213.1.0 from being injected back into BGP as if it originated from AS 200, and allows BGP to advertise network 175.220.0.0 as originating from AS 200.

Note Redistribution of dynamic routes requires careful use of access lists to prevent updates from being injected back into BGP. If possible, you should use the **network** command (described in the section “Using the network Command,” later in this chapter) or redistribute static routes instead of redistributing dynamic routes.

Using the network Command

Another way to advertise networks is to use the **network** router configuration command. When used with BGP, the **network** command specifies the networks that the AS originates. (By way of contrast, when used with an IGP such as RIP, the **network** command identifies the interfaces on which the IGP is to run.) The **network** command works for networks that the router learns dynamically or that are configured as static routes. The origin attribute of routes that are injected into BGP by means of the **network** command is set to IGP.

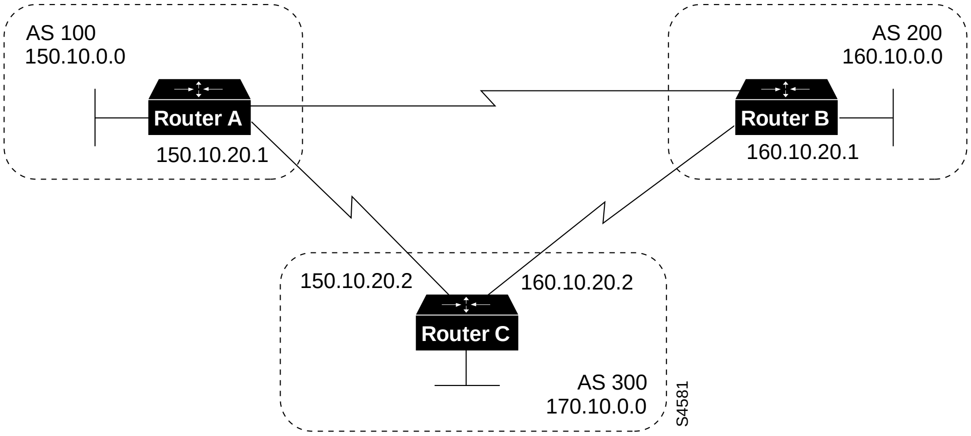
The following commands configure Router C to advertise network 175.220.0.0:

```
!Router C
router bgp 200
neighbor 1.1.1.1 remote-as 300
network 175.220.0.0
```

The **network** router configuration command causes Router C to generate an entry in the BGP routing table for network 175.220.0.0.

Figure 12-10 shows another topology that demonstrates the effects of the **network** command.

Figure 12-10 Network Advertisement Example 2



The following configurations use the **network** command to configure the routers shown in Figure 12-10:

```
!Router A
router bgp 100
neighbor 150.10.20.2 remote-as 300
network 150.10.0.0

!Router B
router bgp 200
neighbor 160.10.20.2 remote-as 300
network 160.10.0.0

!Router C
router bgp 300
neighbor 150.10.20.1 remote-as 100
neighbor 160.10.20.1 remote-as 200
network 170.10.0.0
```

To ensure a loop-free interdomain topology, BGP does not accept updates that originated from its own AS. For example, in Figure 12-10, if Router A generates an update for network 150.10.0.0 with the origin set to AS 100 and sends it to Router C, Router C will pass the update to Router B with the origin still set to AS 100. Router B will send the update (with the origin still set to AS 100) to Router A, which will recognize that the update originated from its own AS and will ignore it.

BGP Decision Algorithm

When a BGP speaker receives updates from multiple ASs that describe different paths to the same destination, it must choose the single best path for reaching that destination. Once chosen, BGP propagates the best path to its neighbors. The decision is based on the value of attributes (such as next hop, administrative weights, local preference, the origin of the route, and path length) that the update contains and other BGP-configurable factors. This section describes the following attributes and factors that BGP uses in the decision-making process:

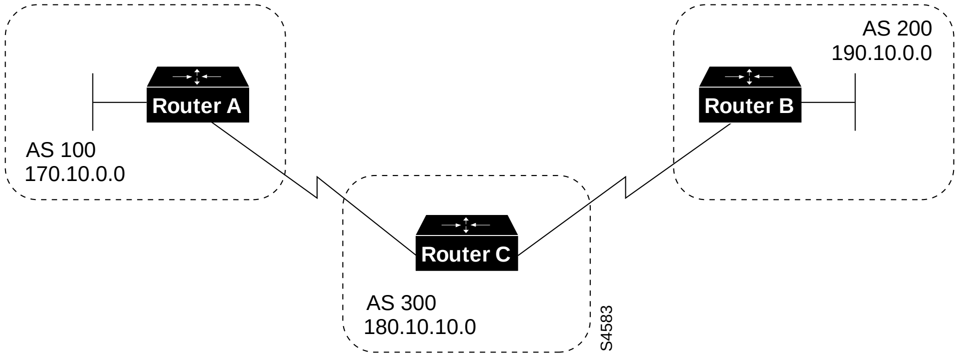
- AS_path Attribute
- Origin Attribute
- Next Hop Attribute
- Weight Attribute
- Local Preference Attribute
- Multi-Exit Discriminator Attribute
- Community Attribute

AS_path Attribute

Whenever an update passes through an AS, BGP prepends its AS number to the update. The AS_path attribute is the list of AS numbers that an update has traversed in order to reach a destination. An AS-SET is a mathematical set of all the ASs that have been traversed.

Consider the network shown in Figure 12-11.

Figure 12-11 AS_path Attribute



In Figure 12-11, Router B advertises network 190.10.0.0 in AS 200 with an AS_path of 200. When the update for 190.10.0.0 traverses AS 300, Router C prepends its own AS number to it, so when the update reaches Router A, two AS numbers have been attached to it: 200 and then 300. That is, the AS_path attribute for reaching network 190.10.0.0 from Router A is 300, 200. Likewise, the AS_path attribute for reaching network 170.10.0.0 from Router B is 300, 100.

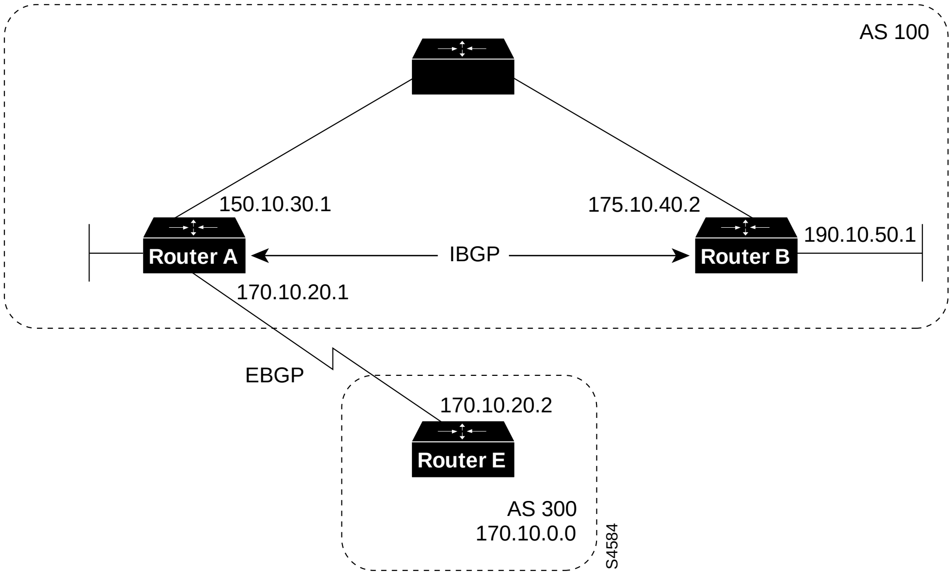
Origin Attribute

The origin attribute provides information about the origin of the route. The origin of a route can be one of three values:

- *IGP*—The route is interior to the originating AS. This value is set when the **network** router configuration command is used to inject the route into BGP. The IGP origin type is represented by the letter i in the output of the **show ip bgp** EXEC command.
- *EGP*—The route is learned via the Exterior Gateway Protocol (EGP). The EGP origin type is represented by the letter e in the output of the **show ip bgp** EXEC command.
- *Incomplete*—The origin of the route is unknown or learned in some other way. An origin of Incomplete occurs when a route is redistributed into BGP. The Incomplete origin type is represented by the ? symbol in the output of the **show ip bgp** EXEC command.

Figure 12-12 shows a network that demonstrates the value of the origin attribute.

Figure 12-12 Origin Attribute



The following commands configure the routers shown in Figure 12-12:

```
!Router A
router bgp 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0
redistribute static
!
ip route 190.10.0.0 255.255.0.0 null 0
```

```
!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100
network 190.10.50.0

!Router E
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

Given these configurations, the following is true:

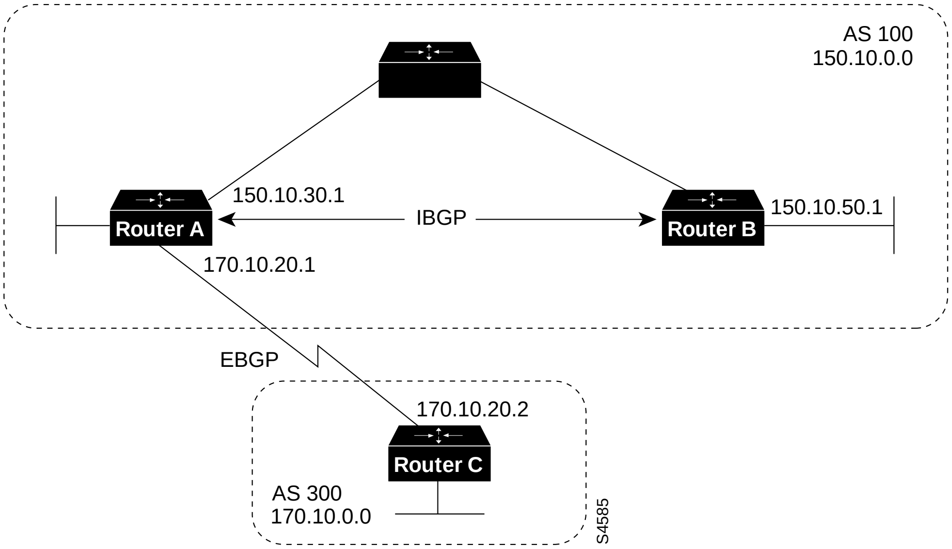
- From Router A, the route for reaching 170.10.0.0 has an AS_path of 300 and an origin attribute of IGP.
- From Router A, the route for reaching 190.10.50.0 has an empty AS_path (the route is in the same AS as Router A) and an origin attribute of IGP.
- From Router E, the route for reaching 150.10.0.0 has an AS_path of 100 and an origin attribute of IGP.
- From Router E, the route for reaching 190.10.0.0 has an AS_path of 100 and an origin attribute of Incomplete (because 190.10.0.0 is a redistributed route).

Next Hop Attribute

The BGP next hop attribute is the IP address of the next hop that is going to be used to reach a certain destination.

For EBGP, the next hop is usually the IP address of the neighbor specified by the **neighbor remote-as** router configuration command. (The exception is when the next hop is on a multiaccess media, in which case, the next hop could be the IP address of the router in the same subnet.) Consider the network shown in Figure 12-13.

Figure 12-13 Next Hop Attribute



In Figure 12-13, Router C advertises network 170.10.0.0 to Router A with a next hop attribute of 170.10.20.2, and Router A advertises network 150.10.0.0 to Router C with a next hop attribute of 170.10.20.1.

BGP specifies that the next hop of EBGP-learned routes should be carried without modification into IBGP. Because of that rule, Router A advertises 170.10.0.0 to its IBGP peer (Router B) with a next hop attribute of 170.10.20.2. As a result, according to Router B, the next hop to reach 170.10.0.0 is 170.10.20.2, instead of 150.10.30.1. For that reason, the configuration must ensure that Router B can reach 170.10.20.2 via an IGP. Otherwise, Router B will drop packets destined for 170.10.0.0 because the next hop address is inaccessible.

For example, if Router B runs IGRP, Router A should run IGRP on network 170.10.0.0. You might want to make IGRP passive on the link to Router C so that only BGP updates are exchanged.

The following commands configure the routers shown in Figure 12-13:

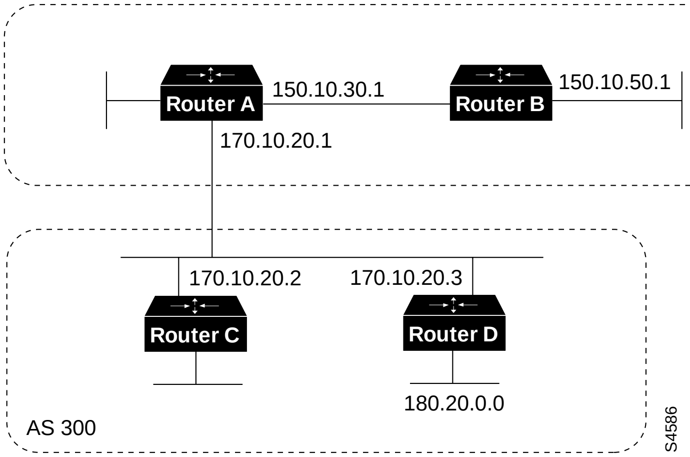
```
!Router A
router bgp 100
neighbor 170.10.20.2 remote-as 300
neighbor 150.10.50.1 remote-as 100
network 150.10.0.0

!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100

!Router C
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

Note Router C advertises 170.10.0.0 to Router A with a next hop attribute of 170.10.20.2, and Router A advertises 170.10.0.0 to Router B with a next hop attribute of 170.10.20.2. The next hop of EBGP-learned routes is passed to the IBGP neighbor.

Next Hop Attribute and Multiaccess Media

BGP might set the value of the next hop attribute differently on multiaccess media, such as Ethernet. Consider the network shown in Figure 12-14.

Figure 12-14 Next Hop Attribute and Multiaccess Media

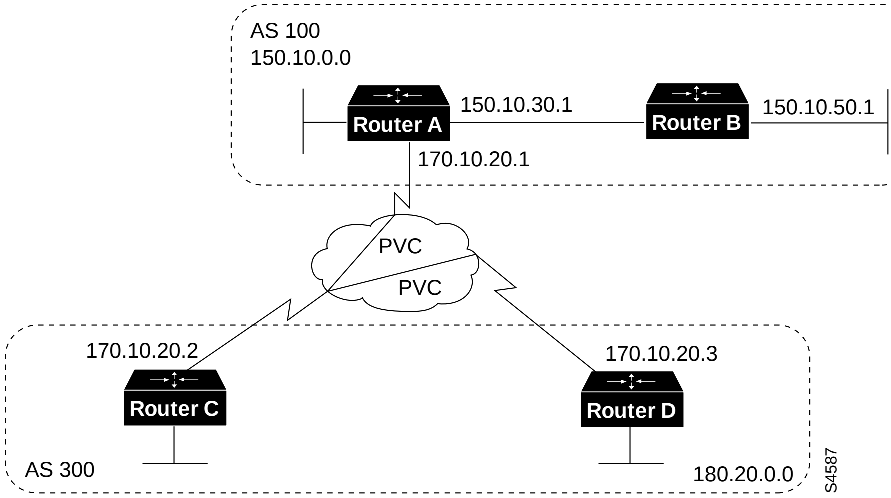


In Figure 12-14, Routers C and D in AS 300 are running OSPF. Router C is running BGP with Router A. Router C can reach network 180.20.0.0 via 170.10.20.3. When Router C sends a BGP update to Router A regarding 180.20.0.0, it sets the next hop attribute to 170.10.20.3, instead of its own IP address (170.10.20.2). This is because Routers A, B, and C are in the same subnet, and it makes more sense for Router A to use Router D as the next hop rather than taking an extra hop via Router C.

Next Hop Attribute and Nonbroadcast Media Access

In Figure 12-15, three networks are connected by a nonbroadcast media access (NBMA) cloud, such as Frame Relay.

Figure 12-15 Next Hop Attribute and Nonbroadcast Media Access



If Routers A, C, and D, use a common media such as Frame Relay (or any NBMA cloud), Router C advertises 180.20.0.0 to Router A with a next hop of 170.10.20.3, just as it would do if the common media were Ethernet. The problem is that Router A does not have a direct permanent virtual connection (PVC) to Router D and cannot reach the next hop, so routing will fail. To remedy this situation, use the **neighbor next-hop-self** router configuration command, as shown in the following configuration for Router C:

```
!Router C
router bgp 300
neighbor 170.10.20.1 remote-as 100
neighbor 170.10.20.1 next-hop-self
```

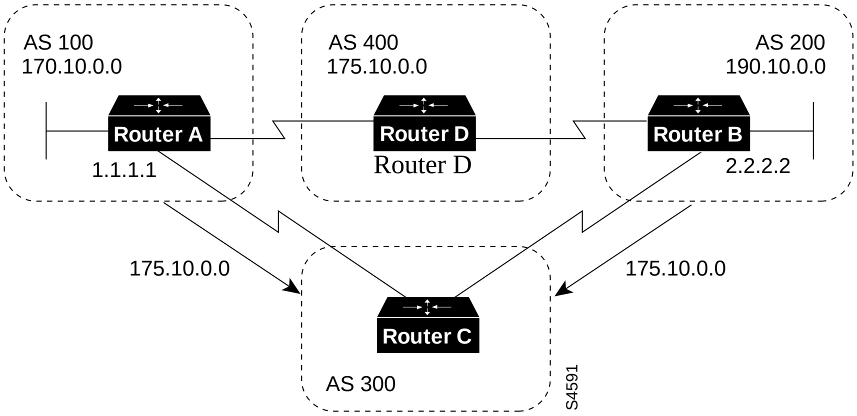
The **neighbor next-hop-self** command causes Router C to advertise 180.20.0.0 with the next hop attribute set to 170.10.20.2.

Weight Attribute

The weight attribute is a special Cisco attribute that is used in the path selection process when there is more than one route to the same destination. The weight attribute is local to the router on which it is assigned, and it is not propagated in routing updates. By default, the weight attribute is 32768 for paths that the router originates and zero for other paths. Routes with a higher weight are preferred when there are multiple routes to the same destination.

Consider the network shown in Figure 12-16.

Figure 12-16 Weight Example



In Figure 12-16, Routers A and B learn about network 175.10.0.0 from AS 400, and each propagates the update to Router C. Router C has two routes for reaching 175.10.0.0 and has to decide which route to use. If, on Router C, you set the weight of the updates coming in from Router A to be higher than the updates coming in from Router B, Router C will use Router A as the next hop to reach network 175.10.0.0.

There are three ways to set the weight for updates coming in from Router A:

- Using an Access List to Set the Weight Attribute
- Using a Route Map to Set the Weight Attribute
- Using the neighbor weight Command to Set the Weight Attribute

Using an Access List to Set the Weight Attribute

The following commands on Router C use access lists and the value of the AS_path attribute to assign a weight to route updates:

```
!Router C
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 filter-list 5 weight 2000
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 filter-list 6 weight 1000
!
ip as-path access-list 5 permit ^100$
ip as-path access-list 6 permit ^200$
```

In this example, 2000 is assigned to the weight attribute of updates from the neighbor at IP address 1.1.1.1 that are permitted by access list 5. Access list 5 permits updates whose AS_path attribute starts with 100 (as specified by ^) and ends with 100 (as specified by $). (The ^ and $ symbols are used to form regular expressions. For a complete explanation of regular expressions, see the appendix on regular expressions in the Cisco Internetwork Operating System (Cisco IOS) software configuration guides and command references.

This example also assigns 1000 to the weight attribute of updates from the neighbor at IP address 2.2.2.2 that are permitted by access list 6. Access list 6 permits updates whose AS_path attribute starts with 200 and ends with 200.

In effect, this configuration assigns 2000 to the weight attribute of all route updates received from AS 100 and assigns 1000 to the weight attribute of all route updates from AS 200.

Using a Route Map to Set the Weight Attribute

The following commands on Router C use a route map to assign a weight to route updates:

```
!Router C
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-map SETWEIGHTIN in
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 route-map SETWEIGHTIN in
!
ip as-path access-list 5 permit ^100$
!
route-map SETWEIGHTIN permit 10
match as-path 5
set weight 2000
route-map SETWEIGHTIN permit 20
set weight 1000
```

This first instance of the SETWEIGHTIN route map assigns 2000 to any route update from AS 100, and the second instance of the SETWEIGHTIN route map assigns 1000 to route updates from any other AS.

Using the neighbor weight Command to Set the Weight Attribute

The following configuration for Router C uses the **neighbor weight** router configuration command:

```
!Router C
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 weight 2000
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 weight 1000
```

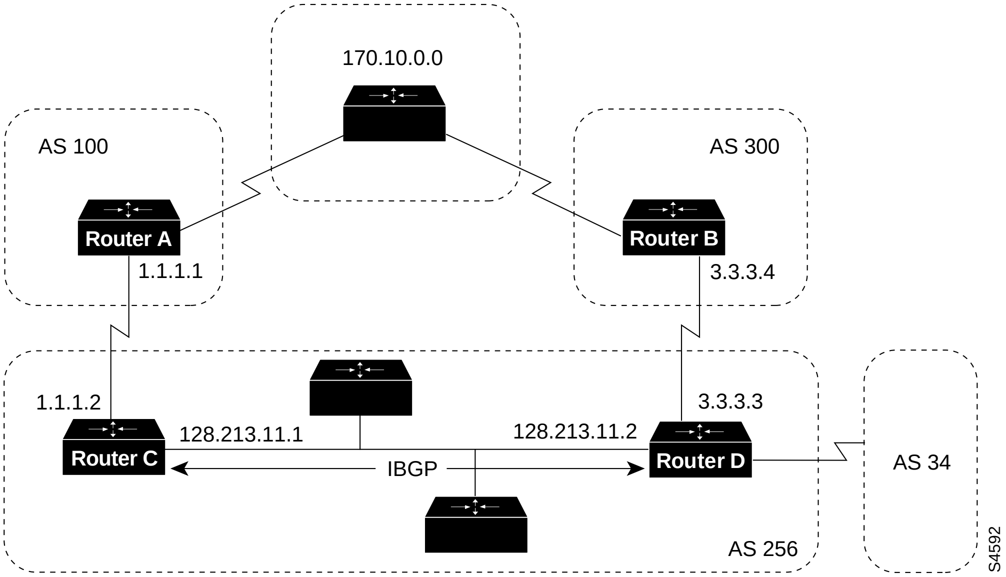
This configuration sets the weight of all route updates from AS 100 to 2000, and the weight of all route updates coming from AS 200 to 1000. The higher weight assigned to route updates from AS 100 causes Router C to send traffic through Router A.

Local Preference Attribute

When there are multiple paths to the same destination, the local preference attribute indicates the preferred path. The path with the higher preference is preferred (the default value of the local preference attribute is 100). Unlike the weight attribute, which is only relevant to the local router, the local preference attribute is part of the routing update and is exchanged among routers in the same AS.

The network shown in Figure 12-17 demonstrates the local preference attribute.

Figure 12-17 Local Preference



In Figure 12-17, AS 256 receives route updates for network 170.10.0.0 from AS 100 and AS 300. There are two ways to set local preference:

- Using the bgp default local-preference Command
- Using a Route Map to Set Local Preference

Using the bgp default local-preference Command

The following configurations use the **bgp default local-preference** router configuration command to set the local preference attribute on Routers C and D:

```
!Router C
router bgp 256
neighbor 1.1.1.1 remote-as 100
neighbor 128.213.11.2 remote-as 256
bgp default local-preference 150
```

```
!Router D
router bgp 256
neighbor 3.3.3.4 remote-as 300
neighbor 128.213.11.1 remote-as 256
bgp default local-preference 200
```

The configuration for Router C causes it to set the local preference of all updates from AS 300 to 150, and the configuration for Router D causes it to set the local preference for all updates from AS 100 to 200. Because local preference is exchanged within the AS, both Routers C and D determine that updates regarding network 170.10.0.0 have a higher local preference when they come from AS 300 than when they come from AS 100. As a result, all traffic in AS 256 destined for network 170.10.0.0 is sent to Router D as the exit point.

Using a Route Map to Set Local Preference

Route maps provide more flexibility than the **bgp default local-preference** router configuration command. When the **bgp default local-preference** command is used on Router D in Figure 12-17, the local preference attribute of all updates received by Router D will be set to 200, including updates from AS 34.

The following configuration uses a route map to set the local preference attribute on Router D specifically for updates regarding AS 300:

```
!Router D
router bgp 256
neighbor 3.3.3.4 remote-as 300
route-map SETLOCALIN in
neighbor 128.213.11.1 remote-as 256
!
ip as-path 7 permit ^300$
route-map SETLOCALIN permit 10
match as-path 7
set local-preference 200
!
route-map SETLOCALIN permit 20
```

With this configuration, the local preference attribute of any update coming from AS 300 is set to 200. Instance 20 of the SETLOCALIN route map accepts all other routes.

Multi-Exit Discriminator Attribute

The multi-exit discriminator (MED) attribute is a hint to external neighbors about the preferred path into an AS when there are multiple entry points into the AS. A lower MED value is preferred over a higher MED value. The default value of the MED attribute is 0.

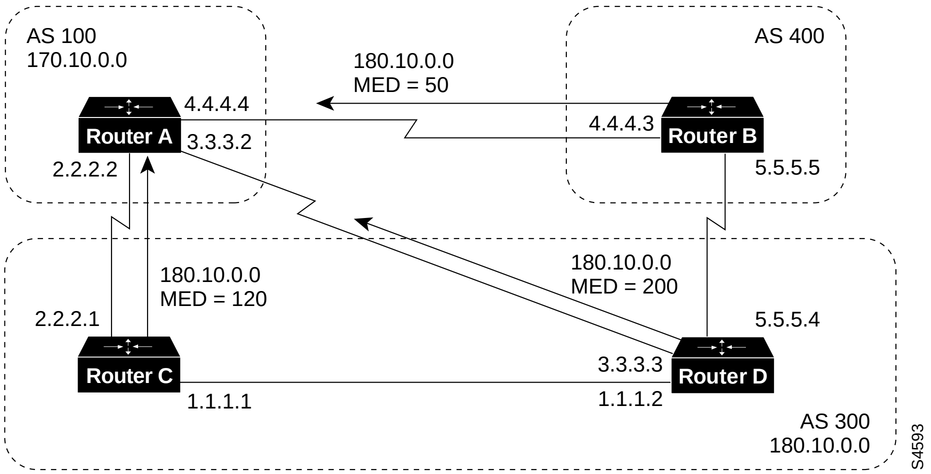
Note In BGP Version 3, MED is known as Inter-AS_Metric.

Unlike local preference, the MED attribute is exchanged between ASs, but a MED attribute that comes into an AS does not leave the AS. When an update enters the AS with a certain MED value, that value is used for decision making within the AS. When BGP sends that update to another AS, the MED is reset to 0.

Unless otherwise specified, the router compares MED attributes for paths from external neighbors that are in the same AS. If you want MED attributes from neighbors in other ASs to be compared, you must configure the **bgp always-compare-med** command.

The network shown in Figure 12-18 demonstrates the use of the MED attribute.

Figure 12-18 MED Example



In Figure 12-18, AS 100 receives updates regarding network 180.10.0.0 from Routers B, C, and D. Routers C and D are in AS 300, and Router B is in AS 400.

The following commands configure Routers A, B, C, and D:

```
!Router A
router bgp 100
neighbor 2.2.2.1 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400

!Router B
router bgp 400
neighbor 4.4.4.4 remote-as 100
neighbor 4.4.4.4 route-map SETMEDOUT out
neighbor 5.5.5.4 remote-as 300
!
route-map SETMEDOUT permit 10
set metric 50

!Router C
router bgp 300
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map SETMEDOUT out
neighbor 5.5.5.5 remote-as 400
neighbor 1.1.1.2 remote-as 300
!
route-map SETMEDOUT permit 10
set metric 120

!Router D
router bgp 300
neighbor 3.3.3.2 remote-as 100
neighbor 3.3.3.2 route map SETMEDOUT out
neighbor 1.1.1.1 remote-as 300
route-map SETMEDOUT permit 10
set metric 200
```

By default, BGP compares the MED attributes of routes coming from neighbors in the same external AS (such as AS 300 in Figure 12-18). Router A can only compare the MED attribute coming from Router C (120) to the MED attribute coming from Router D (200) even though the update coming from Router B has the lowest MED value.

Router A will choose Router C as the best path for reaching network 180.10.0.0. To force Router A to include updates for network 180.10.0.0 from Router B in the comparison, use the **bgp always-compare-med** router configuration command, as in the following modified configuration for Router A:

```
!Router A
router bgp 100
neighbor 2.2.2.1 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400
bgp always-compare-med
```

Router A will choose Router B as the best next hop for reaching network 180.10.0.0 (assuming that all other attributes are the same).

You can also set the MED attribute when you configure the redistribution of routes into BGP. For example, on Router B you can inject the static route into BGP with a MED of 50 as in the following configuration:

```
!Router B
router bgp 400
redistribute static
default-metric 50
!
ip route 160.10.0.0 255.255.0.0 null 0
```

The preceding configuration causes Router B to send out updates for 160.10.0.0 with a MED attribute of 50.

Community Attribute

The community attribute provides a way of grouping destinations (called *communities*) to which routing decisions (such as acceptance, preference, and redistribution) can be applied.

Route maps are used to set the community attribute. A few predefined communities are listed in Table 12-1.

Table 12-1 Predefined Communities

Community	Meaning
no-export	Do not advertise this route to EBGP peers.
no-advertise	Do not advertise this route to any peer.
internet	Advertise this route to the internet community; all routers in the network belong to it.

The following route maps set the value of the community attribute:

```
route-map COMMUNITYMAP
match ip address 1
set community no-advertise
!
route-map SETCOMMUNITY
match as-path 1
set community 200 additive
```

If you specify the **additive** keyword, the specified community value is added to the existing value of the community attribute. Otherwise, the specified community value replaces any community value that was set previously.

To send the community attribute to a neighbor, you must use the **neighbor send-community** router configuration command, as in the following example:

```
router bgp 100
neighbor 3.3.3.3 remote-as 300
neighbor 3.3.3.3 send-community
neighbor 3.3.3.3 route-map setcommunity out
```

For examples of how the community attribute is used to filter updates, see the section “Community Filtering,” later in this chapter.

Summary of the BGP Path Selection Process

BGP selects only one path as the best path. When the path is selected, BGP puts the selected path in its routing table and propagates the path to its neighbors. BGP uses the following criteria, in the order presented, to select a path for a destination:

1. If the path specifies a next hop that is inaccessible, drop the update.
2. Prefer the path with the largest weight.
3. If the weights are the same, prefer the path with the largest local preference.
4. If the local preferences are the same, prefer the path that was originated by BGP running on this router.
5. If no route was originated, prefer the route that has the shortest AS_path.
6. If all paths have the same AS_path length, prefer the path with the lowest origin type (where IGP is lower than EGP, and EGP is lower than Incomplete).
7. If the origin codes are the same, prefer the path with the lowest MED attribute.
8. If the paths have the same MED, prefer the external path over the internal path.
9. If the paths are still the same, prefer the path through the closest IGP neighbor.
10. Prefer the path with the lowest IP address, as specified by the BGP router ID.

Controlling the Flow of BGP Updates

This section describes techniques for controlling the flow of BGP updates. The techniques include the following:

- Administrative Distance
- BGP Filtering
- BGP Peer Groups
- CIDR and Aggregate Addresses
- Confederations
- Route Reflectors
- Route Flap Dampening

Administrative Distance

Normally, a route could be learned via more than one protocol. Administrative distance is used to discriminate between routes learned from more than one protocol. The route with the lowest administrative distance is installed in the IP routing table. By default, BGP uses the administrative distances shown in Table 12-2.

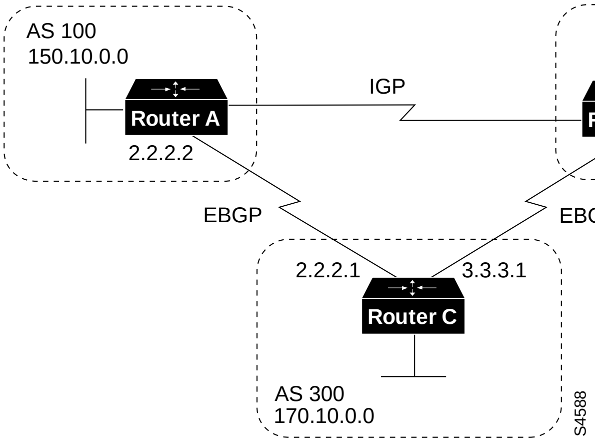
Table 12-2 BGP Default Distances

Distance	Default Value	Function
External	20	Applied to routes learned from EBGP
Internal	200	Applied to routes learned from IBGP
Local	200	Applied to routes originated by the router

Note Distance does not influence the BGP path selection algorithm, but it does influence whether BGP-learned routes are installed in the IP routing table.

Usually when a route is learned via EBGP, it is installed in the IP routing table because of its distance (20). Sometimes, however, two ASs have an IGP-learned backdoor route and an EBGP-learned route. Their policy might be to use the IGP-learned path as the preferred path and to use the EBGP-learned path when the IGP path is down. The network in Figure 12-19 shows this situation.

Figure 12-19 Back Door Example



In Figure 12-19, Routers A and C are running EBGP, as are Routers B and C. Routers A and B are running an IGP (such as RIP, IGRP, Enhanced IGRP, or OSPF). The default distances for RIP, IGRP, Enhanced IGRP, and OSPF are 120, 100, 90, and 110, respectively. All of these default distances are higher than the default distance of EBGP (which is 20). Usually, the route with the lowest distance is preferred.

Router A receives updates about 160.10.0.0 from two routing protocols: EBGP and an IGP. Because the default distance for EBGP is lower than the default distance of the IGP, Router A will choose the EBGP-learned route from Router C. If you want Router A to learn about 160.10.0.0 from Router B (IGP), you could use one of the following techniques:

- Change the external distance of EBGP. (*Not recommended because the distance will affect all updates, which might lead to undesirable behavior when multiple routing protocols interact with one another.*)
- Change the distance of the IGP. (*Not recommended because the distance will affect all updates, which might lead to undesirable behavior when multiple routing protocols interact with one another.*)
- Establish a BGP back door. (*Recommended*)

To establish a BGP back door, use the **network backdoor** router configuration command.

The following commands configure Router A in Figure 12-19:

```
!Router A
router eigrp 10
network 150.10.0.0
router bgp 100
neighbor 2.2.2.1 remote-as 300
network 160.10.0.0 backdoor
```

With the **network backdoor** command, Router A treats the EBGP-learned route as local and installs it in the IP routing table with a distance of 200. The network is also learned via Enhanced IGRP (with a distance of 90), so the Enhanced IGRP route is successfully installed in the IP routing table and is used to forward traffic. If the Enhanced IGRP-learned route goes down, the EBGP-learned route will be installed in the IP routing table and used to forward traffic.

Note Although BGP treats network 160.10.0.0 as a local entry, it does not advertise network 160.10.0.0 as it normally would advertise a local entry.

BGP Filtering

You can control the sending and receiving of updates by using the following filtering methods:

- Prefix Filtering
- AS_path Filtering
- Route Map Filtering
- Community Filtering

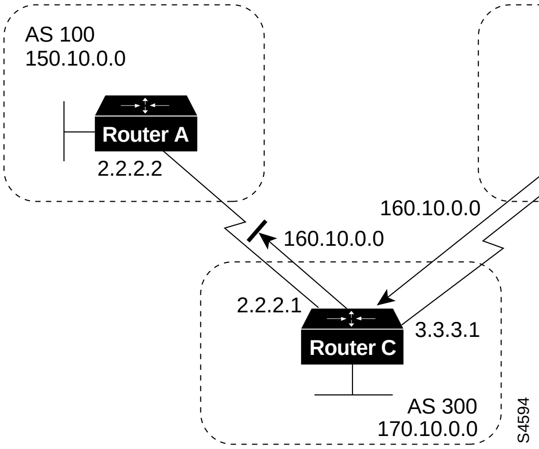
Each method can be used to achieve the same result—the choice of method depends on the specific network configuration.

Prefix Filtering

To restrict the routing information that the router learns or advertises, you can filter based on routing updates to or from a particular neighbor. The filter consists of an access list that is applied to updates to or from a neighbor.

The network shown in Figure 12-20 demonstrates the usefulness of prefix filtering.

Figure 12-20 Route Filtering



In Figure 12-20, Router B is originating network 160.10.0.0 and sending it to Router C. If you want to prevent Router C from propagating updates for network 160.10.0.0 to AS 100, you can apply an access list to filter those updates when Router C exchanges updates with Router A, as demonstrated by the following configuration for Router C:

```
!Router C
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 distribute-list 1 out
!
access-list 1 deny 160.10.0.0 0.0.255.255
access-list 1 permit 0.0.0.0 255.255.255.255
```

In the preceding configuration, the combination of the **neighbor distribute-list** router configuration command and access list 1 prevents Router C from propagating routes for network 160.10.0.0 when it sends routing updates to neighbor 2.2.2.2 (Router A).

Using access lists to filter supernets is a bit trickier. Assume, for example, that Router B in Figure 12-20 has different subnets of 160.10.x.x, and you want to advertise 160.0.0.0/8 only. The following access list would permit 160.0.0.0/8, 160.0.0.0/9, and so on:

```
access-list 1 permit 160.0.0.0 0.255.255.255
```

To restrict the update to 160.0.0.0/8 only, you have to use an extended access list, such as the following:

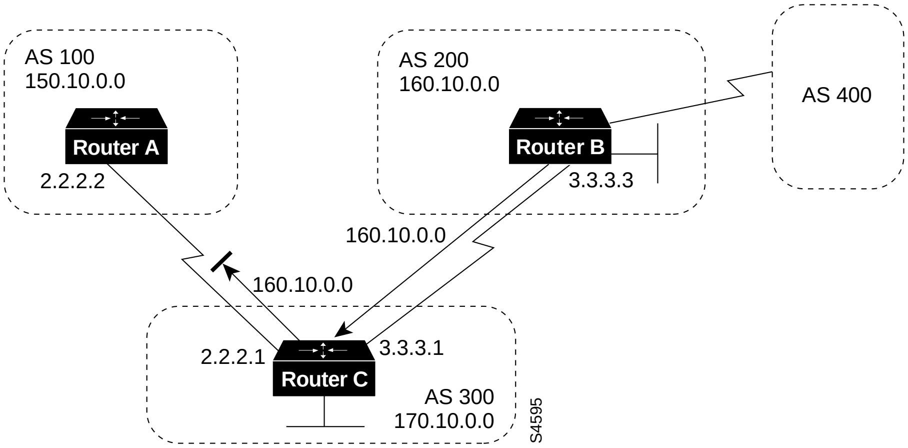
```
access-list 101 permit ip 160.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
```

AS_path Filtering

You can specify an access list on both incoming and outgoing updates based on the value of the AS_path attribute.

The network shown in Figure 12-21 demonstrates the usefulness of AS_path filters.

Figure 12-21 AS_path Filtering



```
!Router C
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 filter-list 1 out
!
ip as-path access-list 1 deny ^200$
ip as-path access-list 1 permit .*
```

In this example, access list 1 denies any update whose AS_path attribute starts with 200 (as specified by ^) and ends with 200 (as specified by $). Because Router B sends updates about 160.10.0.0 whose AS_path attributes start with 200 and end with 200, such updates will match the access list and will be denied. By specifying that the update must also end with 200, the access list permits updates from AS 400 (whose AS_path attribute is 200, 400). If the access list specified ^200 as the regular expression, updates from AS 400 would be denied.

In the second access-list statement, the period (.) symbol means any character, and the asterisk (*) symbol means a repetition of that character. Together, .* matches any value of the AS_path attribute, which in effect permits any update that has not been denied by the previous access-list statement.

If you want to verify that your regular expressions work as intended, use the following EXEC command:

show ip bgp regexp *regular-expression*

The router displays all of the paths that match the specified regular expression.

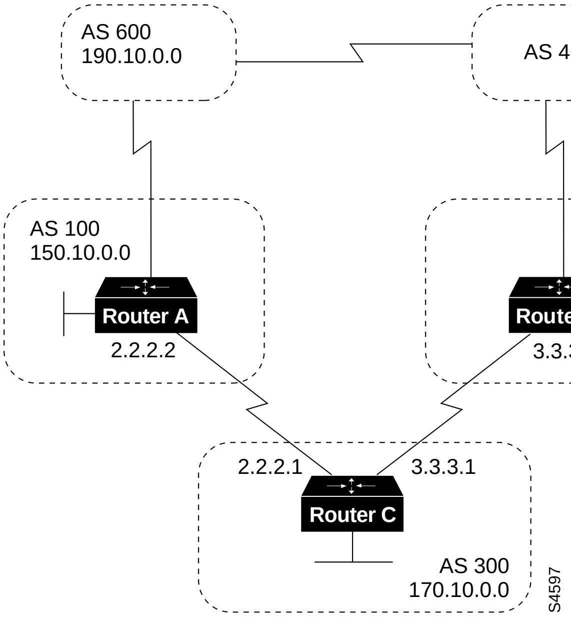
Route Map Filtering

The **neighbor route-map** router configuration command can be used to apply a route map to incoming and outgoing routes.

Note The **neighbor route-map** command has no effect on incoming updates when matching is based on IP address.

The network shown in Figure 12-22 demonstrates using route maps to filter BGP updates.

Figure 12-22 BGP Route Map Filtering



Assume that in Figure 12-22, you want Router C to learn about networks that are local to AS 200 only. (That is, you do not want Router C to learn about AS 100, AS 400, or AS 600 from AS 200.) Also, on those routes that Router C accepts from AS 200, you want the weight attribute to be set to 20. The following configuration for Router C accomplishes this goal:

```
!Router C
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map STAMP in
!
route-map STAMP permit 10
match as-path 1
set weight 20
!
ip as-path access-list 1 permit ^200$
```

In the preceding configuration, access list 1 permits any update whose AS_path attribute begins with 200 and ends with 200 (that is, access list 1 permits updates that originate in AS 200). The weight attribute of the permitted updates is set to 20. All other updates are denied and dropped.

Assume that in Figure 12-22, you want Router C to do the following:

- Accept updates that originate from AS 200 and change their weight attribute to 20.
- Deny updates that contain AS 400.
- Accept any other updates and change their weight attribute to 10.

The following configuration for Router C accomplishes this goal:

```
!Router C
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map STAMP in
```

```
route-map STAMP permit 10
match as-path 1
set weight 20
!
route-map STAMP permit 20
match as-path 2
!
route-map STAMP permit 30
set weight 10
!
ip as-path access-list 1 permit ^200$
ip as-path access-list 2 deny _400_
```

In the preceding configuration, access list 1 permits any update whose AS_path attribute begins with 200 and ends with 200 (that is, access list 1 permits updates that originate in AS 200). The weight attribute of the permitted updates is set to 20. Access list 2 denies updates whose AS_path attribute contains 400. All other updates will have a weight of 10 (by means of instance 30 of the STAMP route map) and will be permitted.

Suppose that in Figure 12-22 Router C advertises its own network (170.10.0.0) to AS 100 and AS 200. When updates about network 170.10.0.0 arrive in AS 600, the routers in AS 600 will have network reachability information via two routes: via AS 100 with an AS_path attribute of (100, 300) and via AS 400 with an AS_path attribute of (400, 200, 300). Assuming that the values of all other attributes are the same, the routers in AS 600 will pick the shortest AS_path attribute: the route through AS 100.

If you want to use the configuration of Router C to influence the choice of paths in AS 600, you can do so by prepending extra AS numbers to the AS_path attribute for routes that Router C advertises to AS 100. A common practice is to repeat the AS number, as in the following configuration:

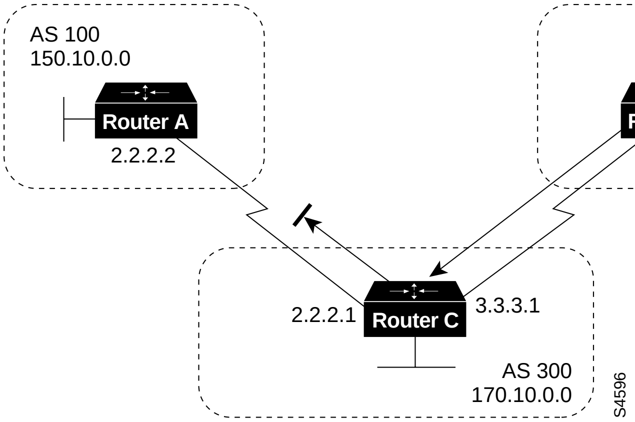
```
!Router C
router bgp 300
network 170.10.0.0
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map SETPATH out
!
route-map SETPATH permit 10
set as-path prepend 300 300
```

The **set as-path** route map configuration command with the **prepend** keyword causes Router C to prepend 300 twice to the value of the AS_path attribute before it sends updates to the neighbor at IP address 2.2.2.2 (Router A). As a result, the AS_path attribute of updates for network 170.10.0.0 that AS 600 receives via AS 100 will be 100, 300, 300, 300, which is longer than the value of the AS_path attribute of updates for network 170.10.0.0 that AS 600 receives via AS 400 (400, 200, 300). AS 600 will choose (400, 200, 300) as the better path.

Community Filtering

The network shown in Figure 12-23 demonstrates the usefulness of community filters.

Figure 12-23 Community Filtering



Assume that you do not want Router C to propagate routes learned from Router B to Router A. You can do this by setting the community attribute on updates that Router B sends to Router C, as in the following configuration for Router B:

```
!Router B
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300
neighbor 3.3.3.1 send-community
neighbor 3.3.3.1 route-map SETCOMMUNITY out
!
route-map SETCOMMUNITY permit 10
match ip address 1
set community no-export
!
route-map SETCOMMUNITY permit 20
!
access list 1 permit 0.0.0.0 255.255.255.255
```

For routes that are sent to the neighbor at IP address 3.3.3.1 (Router C), Router B applies the route map named SETCOMMUNITY. The SETCOMMUNITY route map sets the community attribute of any update (by means of access list 1) destined for 3.3.3.1 to no-export. The **neighbor send-community** router configuration command is required to include the community attribute in updates sent to the neighbor at IP address 3.3.3.1.

When Router C receives the updates from Router B, it does not propagate them to Router A because the value of the community attribute is no-export.

Another way to filter updates based on the value of the community attribute is to use the **ip community-list** global configuration command. Assume that Router B has been configured as follows:

```
!Router B
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300
neighbor 3.3.3.1 send-community
neighbor 3.3.3.1 route-map SETCOMMUNITY out
!
route-map SETCOMMUNITY permit 10
match ip address 2
set community 100 200 additive
```

```
route-map SETCOMMUNITY permit 20
!
access list 2 permit 0.0.0.0 255.255.255.255
```

In the preceding configuration, Router B adds 100 and 200 to the community value of any update destined for the neighbor at IP address 3.3.3.1. To configure Router C to use the **ip community-list** global configuration command to set the value of the weight attribute based on whether the community attribute contains 100 or 200, use the following configuration:

```
!Router C
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map check-community in
!
route-map check-community permit 10
match community 1
set weight 20
!
route-map check-community permit 20
match community 2 exact
set weight 10
!
route-map check-community permit 30
match community 3
!
ip community-list 1 permit 100
ip community-list 2 permit 200
ip community-list 3 permit internet
```

In the preceding configuration, any route that has 100 in its community attribute matches community list 1 and has its weight set to 20. Any route whose community attribute is only 200 (by virtue of the **exact** keyword) matches community list 2 and has its weight set to 10. In the last community list (list 3) the use of the **internet** keyword permits all other updates without changing the value of an attribute. (The **internet** keyword specifies all routes because all routes are members of the internet community.)

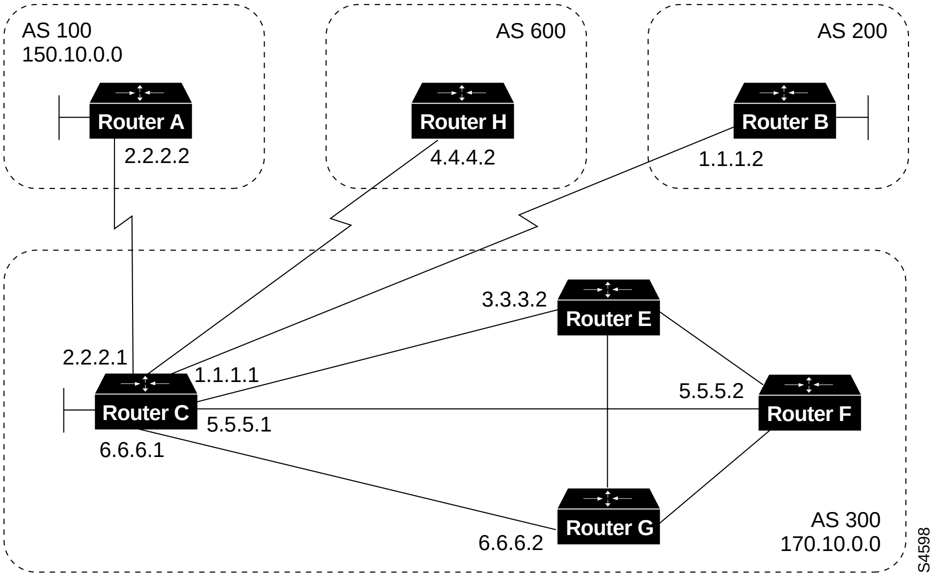
BGP Peer Groups

A BGP peer group is a group of BGP neighbors that share the same update policies. Update policies are usually set by route maps, distribution lists, and filter lists. Instead of defining the same policies for each individual neighbor, you define a peer group name and assign policies to the peer group.

Members of a peer group inherit all of the configuration options of the peer group. Peer group members can also be configured to override configuration options if the options do not affect outgoing updates. That is, you can only override options that are set for incoming updates.

The use of BGP peer groups is demonstrated by the network shown in Figure 12-24.

Figure 12-24 BGP Peer Groups



The following commands configure a BGP peer group named INTERNALMAP on Router C and apply it to the other routers in AS 300:

```
!Router C
router bgp 300
neighbor INTERNALMAP peer-group
neighbor INTERNALMAP remote-as 300
neighbor INTERNALMAP route-map INTERNAL out
neighbor INTERNALMAP filter-list 1 out
neighbor INTERNALMAP filter-list 2 in
neighbor 5.5.5.2 peer-group INTERNALMAP
neighbor 6.6.6.2 peer-group INTERNALMAP
neighbor 3.3.3.2 peer-group INTERNALMAP
neighbor 3.3.3.2 filter-list 3 in
```

The preceding configuration defines the following policies for the internalmap peer group:

- A route map named INTERNAL
- A filter list for outgoing updates (filter list 1)
- A filter list for incoming updates (filter list 2)

The configuration applies the peer group to all internal neighbors—Routers E, F, and G. The configuration also defines a filter list for incoming updates from the neighbor at IP address 3.3.3.2 (Router E). This filter list can only be used to override options that affect incoming updates.

The following commands configure a BGP peer group named EXTERNALMAP on Router C and apply it to routers in AS 100, 200, and 600:

```
!Router C
router bgp 300
neighbor EXTERNALMAP peer-group
neighbor EXTERNALMAP route-map SETMED
neighbor EXTERNALMAP filter-list 1 out
neighbor EXTERNALMAP filter-list 2 in
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 peer-group EXTERNALMAP
neighbor 4.4.4.2 remote-as 600
neighbor 4.4.4.2 peer-group EXTERNALMAP
neighbor 1.1.1.2 remote-as 200
neighbor 1.1.1.2 peer-group EXTERNALMAP
neighbor 1.1.1.2 filter-list 3 in
```

In the preceding configuration, the **neighbor remote-as** router configuration commands are placed outside of the **neighbor peer-group** router configuration commands because different external ASs have to be defined. Also note that this configuration defines filter list 3, which can be used to override configuration options for incoming updates from the neighbor at IP address 1.1.1.2 (Router B).

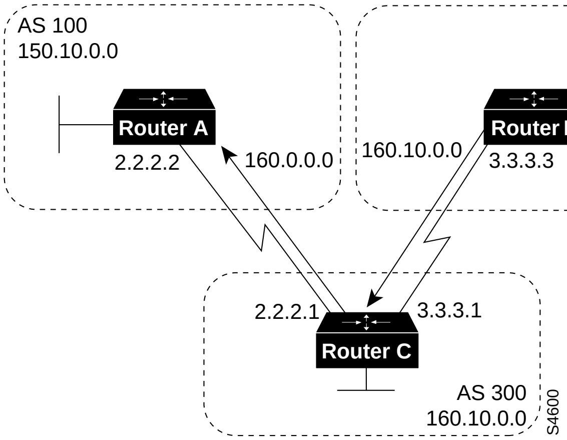
CIDR and Aggregate Addresses

BGP4 supports classless interdomain routing (CIDR), which is a major improvement over BGP3. (CIDR is also known as *supernetting*.) CIDR is a new way of looking at IP addresses that eliminates the concept of classes (Class A, Class B, and so on). For example, network 192.213.0.0, which is an illegal Class C network number, is a legal supernet when it is represented in CIDR notation as 192.213.0.0/16. The /16 indicates that the subnet mask consists of 16 bits (counting from the left). Therefore, 192.213.0.0/16 is similar to 192.213.0.0 255.255.0.0.

CIDR makes it easy to aggregate routes. Aggregation is the process of combining several different routes in such a way that a single route can be advertised, which minimizes the size of routing tables.

Consider the network shown in Figure 12-25.

Figure 12-25 Aggregation



In Figure 12-25, Router B in AS 200 is originating network 160.11.0.0 and advertising it to Router C in AS 300. To configure Router C to propagate the aggregate address 160.0.0.0 to Router A, use the following commands:

```
!Router C
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
network 160.10.0.0
aggregate-address 160.0.0.0 255.0.0.0
```

The **aggregate-address** router configuration command advertises the prefix route (in this case, 160.0.0.0/8) and all of the more specific routes.

Note A router cannot aggregate an address if it does not have a more specific route of that address in the BGP routing table. The more specific route can be injected in the BGP routing table by incoming updates from other ASs, can be redistributed from an IGP, or can be established by the **network** router configuration command.

If you want Router C to propagate the prefix route only, and you do not want it to propagate a more specific route, use the following command:

```
aggregate-address 160.0.0.0 255.0.0.0 summary-only
```

This command propagates the prefix (160.0.0.0/8) and suppresses any more specific routes that the router may have in its BGP routing table.

Note If you use the **network** command to advertise a network, the entry for that network is always injected into BGP updates, even if you specify the **summary-only** keyword with the **aggregate-address** router configuration command.

If you want to suppress specific routes when aggregating routes, you can define a route map and apply it to the aggregate. If, for example, you want Router C in Figure 12-25 to aggregate 160.0.0.0 and suppress the specific route 160.20.0.0, but propagate route 160.10.0.0, use the following commands:

```
!Router C
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
network 160.10.0.0
aggregate-address 160.0.0.0 255.0.0.0 suppress-map CHECK
!
route-map CHECK permit 10
match ip address 1
!
access-list 1 deny 160.20.0.0 0.0.255.255
access-list 1 permit 0.0.0.0 255.255.255.255
```

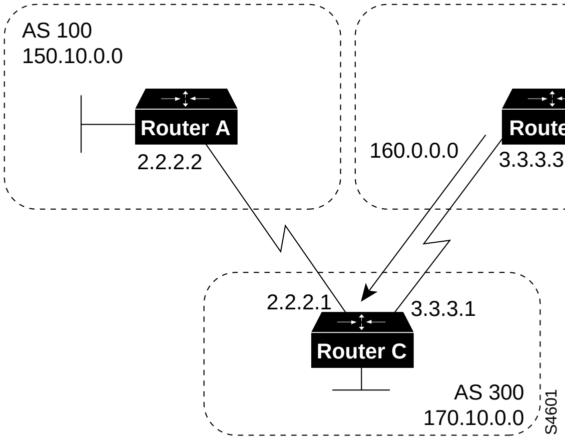
If you want the router to set the value of an attribute when it propagates the aggregate route, use an attribute map, as demonstrated by the following commands:

```
route-map SETORIGIN permit 10
set origin igp
!
aggregate-address 160.0.0.0 255.0.0.0 attribute-map SETORIGIN
```

Aggregation and Static Routes

The network shown in Figure 12-26 demonstrates how static routes can be used to generate aggregates.

Figure 12-26 CIDR Aggregation Example



In Figure 12-26, you want Router B to advertise the prefix 160.0.0.0 and suppress all of the more specific routes.

The following configuration for Router B redistributes a static aggregate route into BGP:

```
!Router B
router bgp 200
neighbor 3.3.3.1 remote-as 300
redistribute static
!
ip route 160.0.0.0 255.0.0.0 null 0
```

As a result of this configuration, Router B advertises the aggregate with an origin attribute whose value is Incomplete.

Using the **network** router command instead of the **redistribute** command, as in the following configuration, has the same effect as the preceding configuration except that the origin attribute of updates for network 160.0.0.0 will be set to IGP instead of Incomplete.

```
!Router B
router bgp 200
network 160.0.0.0 mask 255.0.0.0
neighbor 3.3.3.1 remote-as 300
!
ip route 160.0.0.0 255.0.0.0 null 0
```

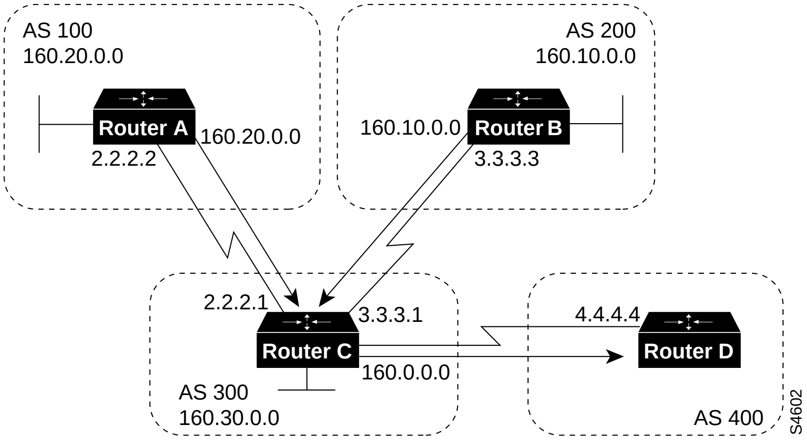
Note The use of static routes (as shown in these two examples) is the preferred method of injecting an aggregate route because using static routes avoids unnecessary route flaps.

Aggregation and AS-SET

When aggregates are generated from more specific routes, the AS_path attributes of the more specific routes are combined to form a set called the AS-SET. This set is useful for preventing routing information loops.

The network shown in Figure 12-27 demonstrates the use of AS-SET when aggregating addresses.

Figure 12-27 CIDR Aggregation Example with AS-SET



In Figure 12-27, Router C is receiving updates about network 160.20.0.0 from Router A and is receiving updates about network 160.10.0.0 from Router B. If Router C aggregates network 160.0.0.0/8 and sends updates for it to Router D, the AS_path attribute of those updates will indicate that AS 300 is the origin of network 160.0.0.0. If Router D has another route to AS 100, the updates from AS 300 may cause a routing loop. To prevent this problem, use the **aggregate-address** router configuration command with the **as-set** keyword, as in the following configuration for Router C:

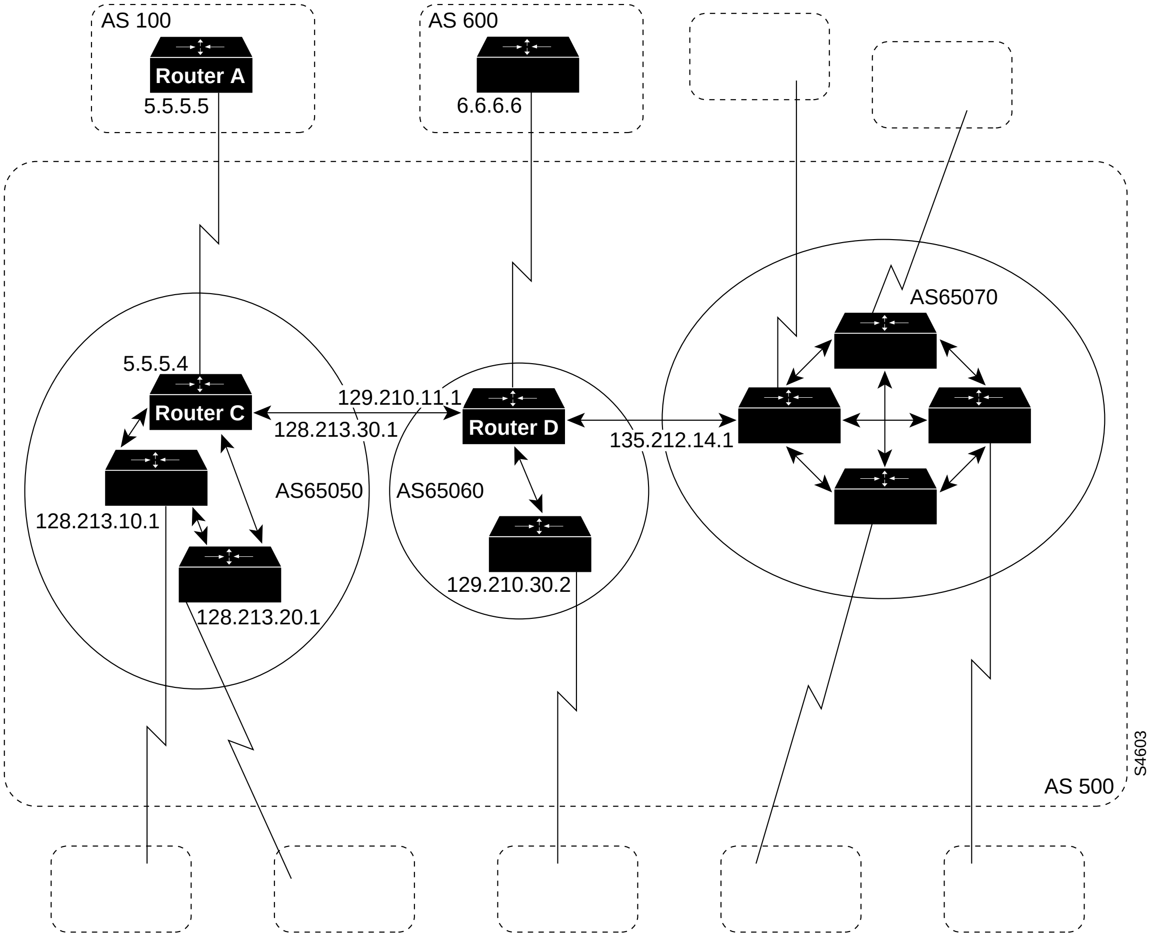
```
!Router C
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 4.4.4.4 remote-as 400
aggregate-address 160.0.0.0 255.0.0.0 as-set
```

The **as-set** keyword causes Router C to generate updates for network 160.0.0.0/8 that include information indicating that network 160.0.0.0 belongs to a set (in this case, the set of 100 and 200).

Confederations

A confederation is a technique for reducing the IBGP mesh inside the AS. Consider the network shown in Figure 12-28.

Figure 12-28 Confederations



In Figure 12-28, AS 500 consists of nine BGP speakers (although there might be other routers that are not configured for BGP). Without confederations, BGP would require that the routers in AS 500 be fully meshed. That is, each router would need to run IBGP with each of the other eight routers, and each router would need to connect to an external AS and run EBGP, for a total of nine peers for each router.

Confederations reduce the number of peers within the AS, as shown in Figure 12-28. You use confederations to divide the AS into multiple mini-ASs and assign the mini-ASs to a confederation. Each mini-AS is fully meshed, and IBGP is run among its members. Each mini-AS has a connection to the other mini-ASs within the confederation. Even though the mini-ASs have EBGP peers to ASs within the confederation, they exchange routing updates as if they were using IBGP—that is, the next hop, MED, and local preference information is preserved. To the outside world, the confederation looks like a single AS.

The following commands configure Router C:

```
!Router C
router bgp 65050
bgp confederation identifier 500
bgp confederation peers 65060 65070
neighbor 128.213.10.1 remote-as 65050
neighbor 128.213.20.1 remote-as 65050
neighbor 128.210.11.1 remote-as 65060
neighbor 135.212.14.1 remote-as 65070
neighbor 5.5.5.5 remote-as 100
```

The **router bgp** global configuration command specifies that Router C belongs to AS 50.

The **bgp confederation identifier** router configuration command specifies that Router C belongs to confederation 500.

The first two **neighbor remote-as** router configuration commands establish IBGP connections to the other two routers within AS 65050. The second two **neighbor remote-as** commands establish BGP connections with confederation peers 65060 and 65070. The last **neighbor remote-as** command establishes an EBGP connection with external AS 100.

The following commands configure Router D:

```
!Router D
router bgp 65060
bgp confederation identifier 500
bgp confederation peers 65050 65070
neighbor 129.210.30.2 remote-as 65060
neighbor 128.213.30.1 remote-as 65050
neighbor 135.212.14.1 remote-as 65070
neighbor 6.6.6.6 remote-as 600
```

The **router bgp** global configuration command specifies that Router D belongs to AS 65060.

The **bgp confederation identifier** router configuration command specifies that Router D belongs to confederation 500.

The first **neighbor remote-as** router configuration command establishes an IBGP connection to the other router within AS 65060. The second two **neighbor remote-as** commands establish BGP connections with confederation peers 65050 and 65070. The last **neighbor remote-as** command establishes an EBGP connection with AS 600.

The following commands configure Router A:

```
!Router A
router bgp 100
neighbor 5.5.5.4 remote-as 500
```

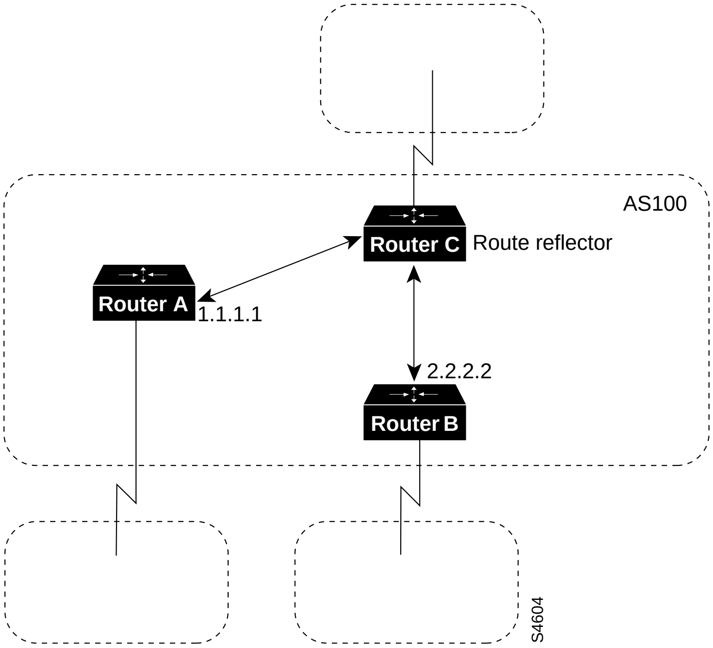
The **neighbor remote-as** command establishes an EBGP connection with Router C. Router A is unaware of AS 65050, AS 65060, or AS 65070. Router A only has knowledge of AS 500.

Route Reflectors

Route reflectors are another solution for the explosion of IBGP peering within an AS. As described earlier in the section “Synchronization,” a BGP speaker does not advertise a route learned from another IBGP speaker to a third IBGP speaker. Route reflectors ease this limitation and allow a router to advertise (reflect) IBGP-learned routes to other IBGP speakers, thereby reducing the number of IBGP peers within an AS.

The network shown in Figure 12-29 demonstrates how route reflectors work.

Figure 12-29 Simple Route Reflector Example



Without a route reflector, the network shown in Figure 12-29 would require a full IBGP mesh (that is, Router A would have to be a peer of Router B). If Router C is configured as a route reflector, IBGP peering between Routers A and B is not required because Router C will reflect updates from Router A to Router B and from Router B to Router A. To configure Router C as a route reflector, use the following commands:

```
!Router C
router bgp 100
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
```

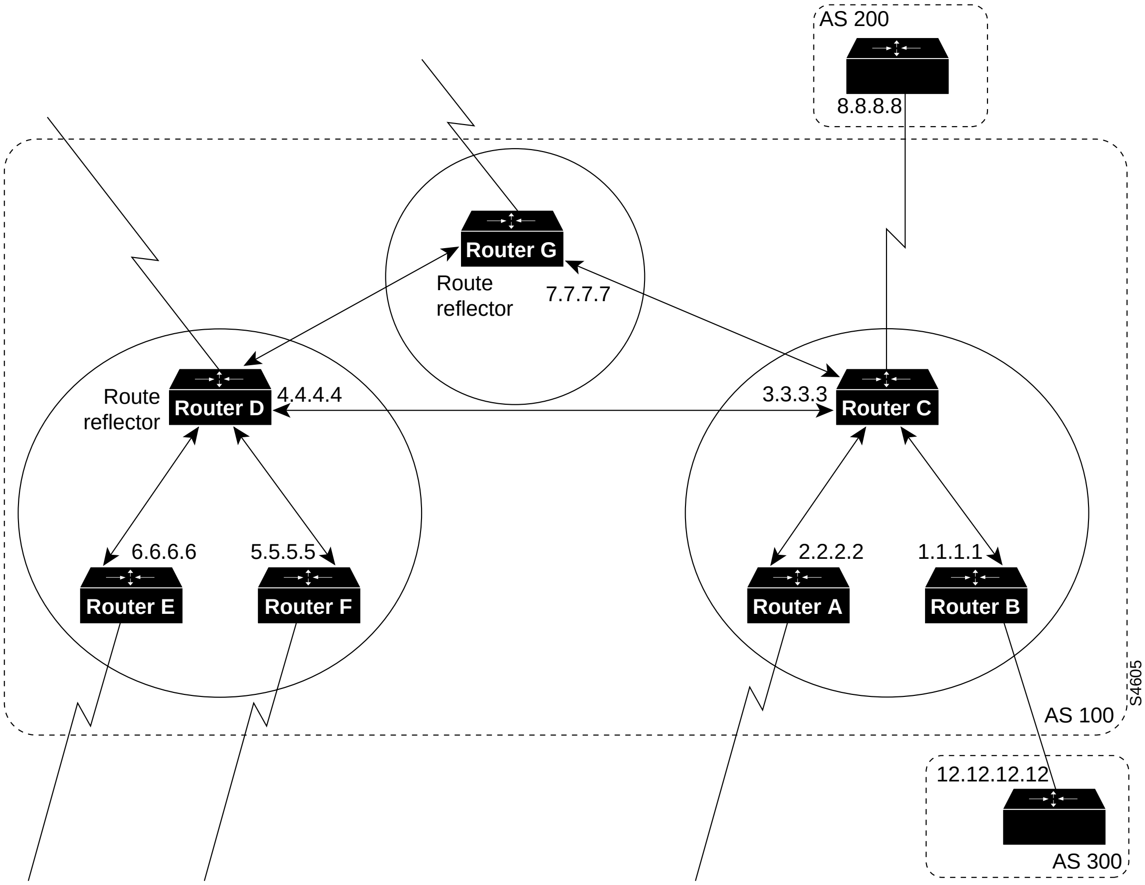
The router whose configuration includes **neighbor route-reflector-client** router configuration commands is the route reflector. The routers identified by the **neighbor route-reflector-client** commands are clients of the route reflector. When considered as a whole, the route reflector and its clients are called a *cluster*. Other IBGP peers of the route reflector that are not clients are called *nonclients*.

An AS can have more than one route reflector. When an AS has more than one route reflector, each route reflector treats other route reflectors as normal IBGP speakers. There can be more than one route reflector in a cluster, and there can be more than one cluster in an AS.

In the advanced configuration shown in Figure 12-30, the AS is divided into multiple clusters, with each cluster having one route reflector. Each route reflector is configured as a nonclient peer of each other route reflector in a fully meshed topology.

Note Route reflector clients should not establish peer relationships with IBGP speakers outside of their cluster.

Figure 12-30 Advanced Route Reflectors Example



In Figure 12-30, Routers A, B, and C form a cluster, and Router C is the route reflector. Routers D, E, and F form a second cluster, of which Router D is the route reflector. Router G forms a third cluster. Note that Routers C, D, and G are fully meshed and that the routers within a cluster are not fully meshed.

When a route reflector in Figure 12-30 receives an update, it takes the following actions, depending on the type of peer that sent the update:

- Update from a nonclient peer—Send the update to all clients in the cluster.
- Update from a client peer—Send the update to all nonclient peers and to all client peers.
- Update from EBGP peer—Send the update to all nonclient peers and to all client peers.

The following configurations establish the route reflectors in AS 100:

```
!Router C
router bgp 100
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 4.4.4.4 remote-as 100
neighbor 8.8.8.8 remote-as 200
```

```
!Router B
router bgp 100
neighbor 3.3.3.3 remote-as 100
neighbor 12.12.12.12 remote-as 300

!Router D
router bgp 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 3.3.3.3 remote-as 100
neighbor 7.7.7.7 remote-as 100
```

If a set clause is used to modify an attribute, a routing loop may occur when the IBGP-learned routes are reflected. BGP automatically prevents the set clause of outgoing route maps from affecting routes reflected to IBGP peers. Another automatic restriction concerns the **neighbor next-hop-self** router configuration command. Because the next hop of reflected routes should not be changed, the **neighbor next-hop-self** command only affects the next hop of EBGP-learned routes when used with route reflectors.

Two techniques prevent routing loops in route reflector configurations:

- Using an Originator ID
- Using a Cluster List

Using an Originator ID

The originator ID is a 4-byte BGP attribute that is created by the route reflector. This attribute carries the router ID of the originator of the route in the local AS. If, because of poor configuration, the update comes back to the originator, the originator ignores it.

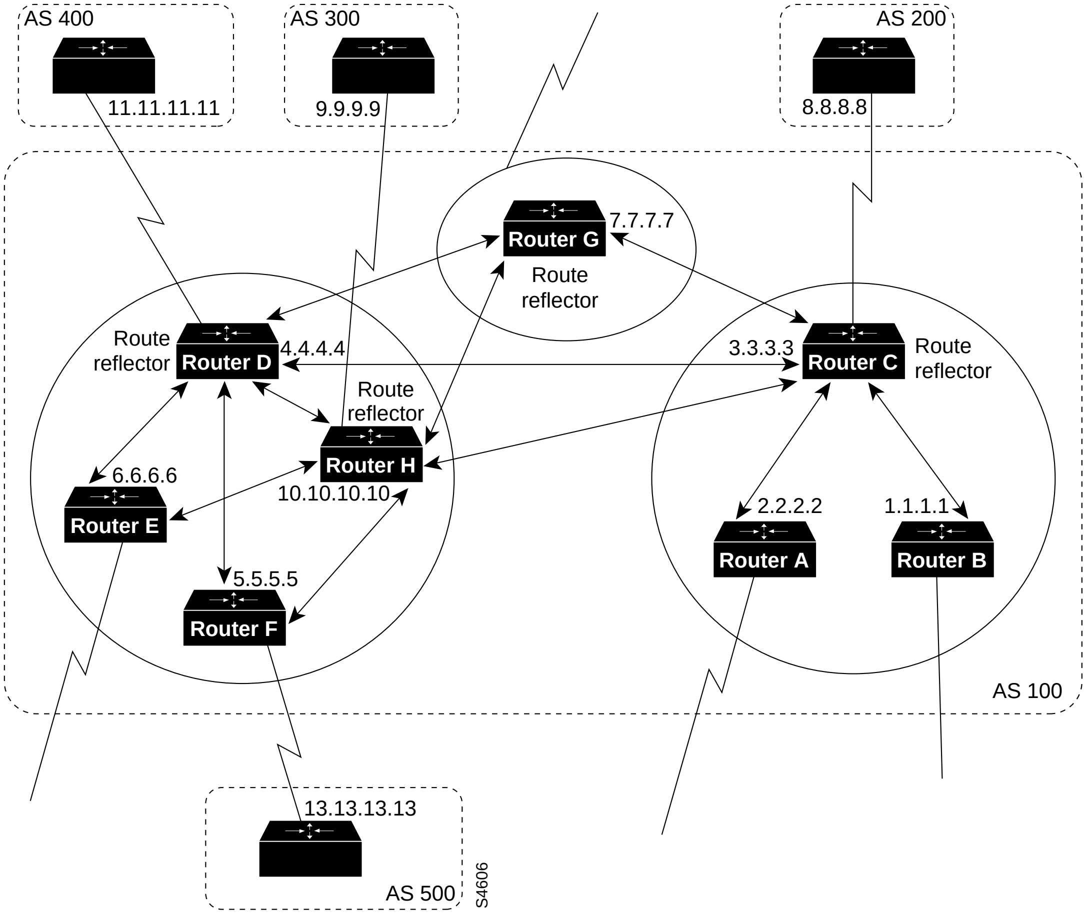
Using a Cluster List

Usually a cluster has a single route reflector, in which case, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid single points of failure, a cluster might have more than one route reflector. When a cluster has more than one route reflector, all of the route reflectors in the cluster need to be configured with a 4-byte cluster ID. The cluster ID allows route reflectors to recognize updates from other route reflectors in the same cluster.

A cluster list is a sequence of cluster IDs that an update has traversed. When a route reflector sends a route from its clients to nonclients outside of the cluster, it appends the local cluster ID to the cluster list. If the route reflector receives an update whose cluster list contains the local cluster ID, the update is ignored.

In Figure 12-31, Routers D, E, F, and H belong to the same cluster; Routers D and H are route reflectors for the same cluster. Note that Routers D and H maintain a fully meshed peering relationship with the other route reflectors in AS 100 (that is, with Routers C and G). If Router D goes down, Router H is prepared to take its place.

Figure 12-31 Route Reflectors and Cluster Lists



The following commands configure Routers C, D, F, and H:

```
!Router C
router bgp 100
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
neighbor 4.4.4.4 remote-as 100
neighbor 7.7.7.7 remote-as 100
neighbor 10.10.10.10 remote-as 100
neighbor 8.8.8.8 remote-as 200

!Router D
neighbor 10.10.10.10 remote-as 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 3.3.3.3 remote-as 100
neighbor 7.7.7.7 remote-as 100
neighbor 11.11.11.11 remote-as 400
bgp cluster-id 10
```

```
!Router F
router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 4.4.4.4 remote-as 100
neighbor 13.13.13.13 remote-as 500

!Router H
router bgp 100
neighbor 4.4.4.4 remote-as 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 3.3.3.3 remote-as 100
neighbor 9.9.9.9 remote-as 300
bgp cluster-id 10
```

The configurations for Routers D and H include the **bgp cluster-id** router configuration command, which sets the cluster ID to 10. The configuration for Router C does not include the **bgp cluster-id** command because Router C is the only route reflector in its cluster.

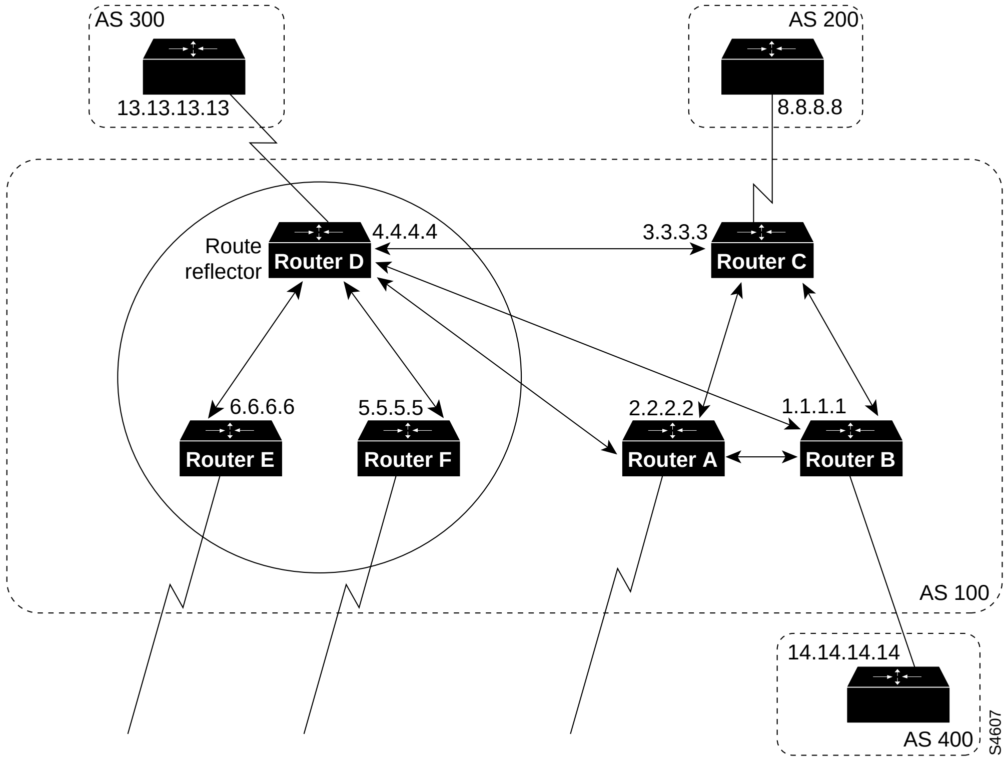
Note You should not configure a peer group within a route reflector cluster. Clients inside a cluster do not have direct IBGP peers; instead, they exchange updates through the route reflector. Configuring peer groups within such a cluster might cause a withdrawal to the source of a route on the route reflector to be sent to all clients in the cluster. If you use the **no bgp client-to-client reflection** command to turn off client-to-client reflection on the route reflector and if you enable redundant BGP peering between the clients, you can use peer groups within a cluster.

Route Reflectors and Conventional BGP Speakers

It is normal for an AS in which route reflectors are configured to have BGP speakers that do not support route reflection. Such routers are known as conventional BGP speakers.

In Figure 12-32, Routers D, E, and F form a route reflector cluster, and Routers A, B, and C are conventional BGP speakers.

Figure 12-32 Route Reflectors and Conventional BGP Speakers



In Figure 12-32, each conventional BGP speaker is peered with the route reflector (Router D), and Routers A, B, and C are peered among each other.

The following commands configure Routers C and D:

```
!Router C
router bgp 100
neighbor 4.4.4.4 remote-as 100
neighbor 2.2.2.2 remote-as 100
neighbor 1.1.1.1 remote-as 100
neighbor 8.8.8.8 remote-as 200

!Router D
router bgp 100
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 3.3.3.3 remote-as 100
neighbor 2.2.2.2 remote-as 100
neighbor 1.1.1.1 remote-as 100
neighbor 13.13.13.13 remote-as 300
```

When it is time to make the conventional BGP speakers members of a cluster, Router C can be configured to be the route reflector, and Routers A and B can be its clients.

Route Flap Dampening

Route flap dampening (introduced in Cisco Internetwork Operating System [Cisco IOS] Release 11.0) is a mechanism for minimizing the instability caused by route flapping. The following terms are used to describe route flap dampening:

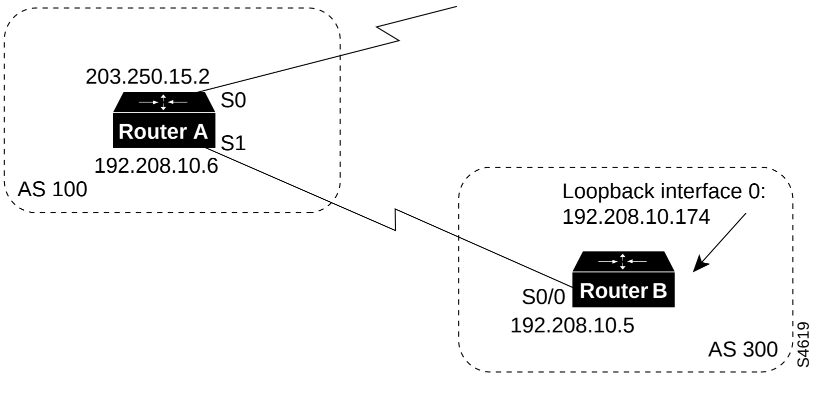
- Penalty—A numeric value that is assigned to a route when it flaps.
- Half-life time—A configurable numeric value that describes the time required to reduce the penalty by one half.
- Suppress limit—A numeric value that is compared with the penalty. If the penalty is greater than the suppress limit, the route is suppressed.
- Suppressed—A route that is not advertised even though it is up. A route is suppressed if the penalty is more than the suppressed limit.
- Reuse limit—A configurable numeric value that is compared with the penalty. If the penalty is less than the reuse limit, a suppressed route that is up will no longer be suppressed.
- History entry—An entry that is used to store flap information about a route that is down.

A route that is flapping receives a penalty of 1000 for each flap. When the accumulated penalty reaches a configurable limit, BGP suppresses advertisement of the route even if the route is up. The accumulated penalty is decremented by the half-life time. When the accumulated penalty is less than the reuse limit, the route is advertised again (if it is still up).

Note Dampening is not applied to routes that are learned via IBGP. This restriction avoids forwarding loops and prevents IBGP peers from having a higher penalty for routes that are external to the AS.

The network shown in Figure 12-33 demonstrates route flap dampening.

Figure 12-33 Route Flap Dampening



The following commands configure Routers A and B:

```
!RouterA
hostname RouterA
!
interface serial 0
ip address 203.250.15.2 255.255.255.252
```

```
interface serial 1
ip address 192.208.10.6 255.255.255.252
!
router bgp 100
bgp dampening
network 203.250.15.0
neighbor 192.208.10.5 remote-as 300

!RouterB
hostname RouterB
!
interface loopback 0
ip address 192.208.10.174 255.255.255.192
!
interface serial 0/0
ip address 192.208.10.5 255.255.255.252
!
router bgp 300
network 192.208.10.0
neighbor 192.208.10.6 remote-as 100
```

Router A is configured for route dampening. Assuming that the EBGP link to Router B is stable, the BGP table on Router A looks like this:

```
RouterB# show ip bgp
table version is 24, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network          Next Hop          Metric LocPrf Weight Path
*>   192.208.10.0     192.208.10.5           0             0 300 i
*>   203.250.15.0     0.0.0.0                0         32768 i
```

To simulate a route flap, enter this command on Router B:

```
clear ip bgp 192.208.10.6
```

Now, the BGP table on Router A looks like this:

```
RouterA# show ip bgp
table version is 24, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network          Next Hop          Metric LocPrf Weight Path
 h   192.208.10.0     192.208.10.5           0             0 300 i
*>   203.250.15.0     0.0.0.0                0         32768 i
```

Because the route for 192.208.10.0. has flapped, the BGP entry for 192.208.10.0 has been withdrawn and put into the history state.

The output of the **show ip bgp** EXEC command for network 192.208.10.0 is as follows:

```
RouterA# show ip bgp 192.208.10.0
BGP routing table entry for 192.208.10.5 255.255.255.0, version 25
Paths: (1 available, no best path)
300 (history entry)
    192.208.10.5 from 192.208.10.5 (192.208.10.174)
Origin IGP, metric 0, external
Dampinfo: penalty 1000, flapped 1 times in 0:02:03
```

The route has been given a penalty (1000) for flapping but the penalty is still below the suppress limit (default 2000). Because the route is down, it is marked as a history entry. If the route flaps a few more times, the **show ip bgp** command displays the following:

```
RouterA# show ip bgp
table version is 32, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network          Next Hop          Metric LocPrf Weight Path
*d   192.208.10.0     192.208.10.5           0             0 300 i
*>   203.250.15.0     0.0.0.0                0         32768 i
```

The output of the **show ip bgp** command for network 192.208.10.0 is as follows:

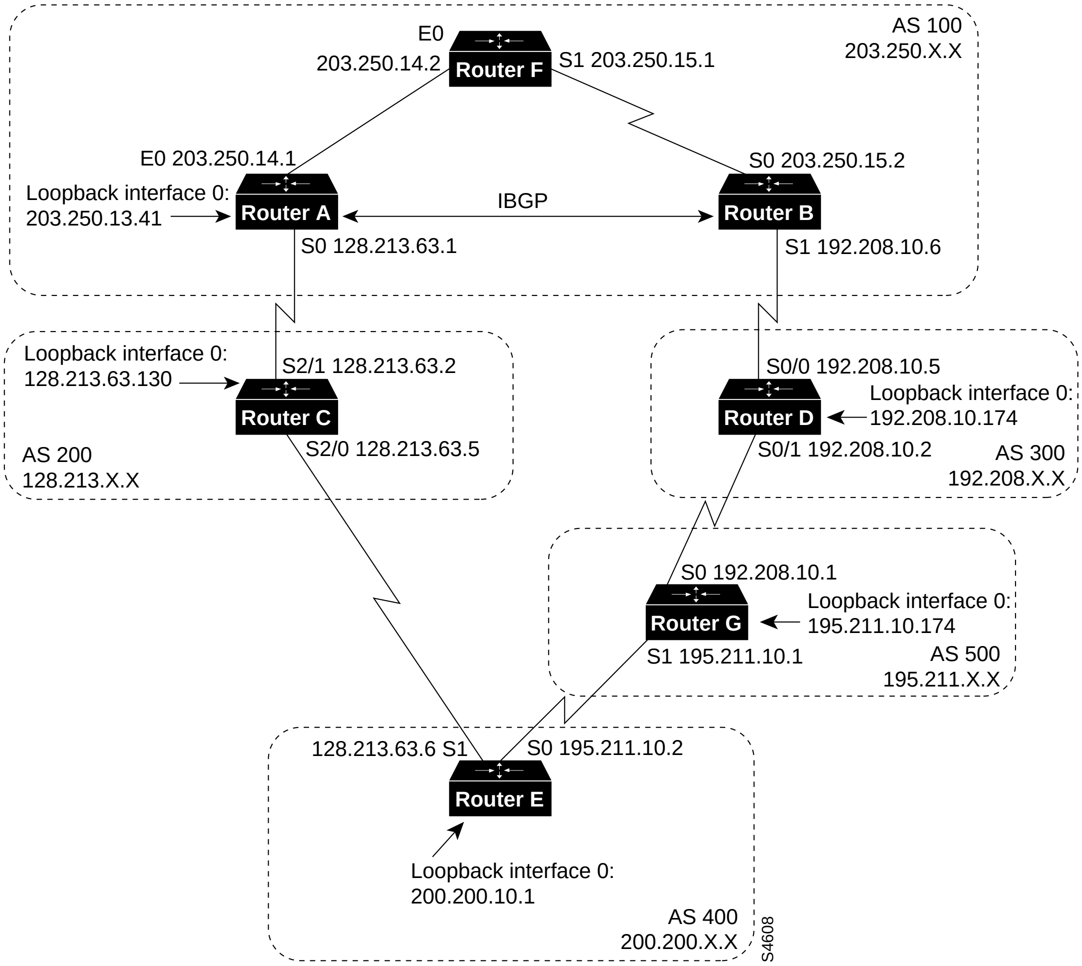
```
RouterA# show ip bgp 192.208.10.0
BGP routing table entry for 192.208.10.5 255.255.255.0, version 32
Paths: (1 available, no best path)
300, (suppressed due to dampening)
    192.208.10.5 from 192.208.10.5 (192.208.10.174)
Origin IGP, metric 0, external
Dampinfo: penalty 2615, flapped 3 times in 0:05:18, reuse in 0:27:00
```

The route is up, but because the penalty is greater than the suppress limit, it is suppressed. The route will be reused when the penalty reaches the reuse limit (default 750). The dampening information will be purged when the penalty becomes less than half of the reuse limit (750/2 = 350).

Practical Design Example

Figure 12-34 shows a BGP network that demonstrates the types of topologies that are typical among ISPs.

Figure 12-34 Practical Design Example for ISPs



Whenever an AS is connected to two ISPs via EBGP, IBGP should be run within the AS for better control over routes. The following configurations for the routers shown in Figure 12-34 run OSPF as the IGP and run IBGP between Routers A and B inside AS 100.

The following configurations are preliminary configurations for the routers shown in Figure 12-34. These preliminary configurations are incomplete so that BGP troubleshooting techniques can be demonstrated. For the complete configurations, see the section, “Final Configurations,” later in this chapter.

```
!Router A
hostname RouterA
!
interface loopback 0
ip address 203.250.13.41 255.255.255.0
!
interface ethernet 0
ip address 203.250.14.1 255.255.255.0
!
interface serial 0
ip address 128.213.63.1 255.255.255.252
!
router ospf 10
network 203.250.0.0 0.0.255.255 area 0
```

```
router bgp 100
network 203.250.13.0 mask 255.255.255.0
network 203.250.14.0 mask 255.255.255.0
neighbor 128.213.63.2 remote-as 200
neighbor 203.250.15.2 remote-as 100
neighbor 203.250.15.2 update-source loopback 0

!Router B
hostname RouterB
!
interface serial 0
ip address 203.250.15.2 255.255.255.252
!
interface serial 1
ip address 192.208.10.6 255.255.255.252
!
router ospf 10
network 203.250.0.0 0.0.255.255 area 0
!
router bgp 100
network 203.250.15.0
neighbor 192.208.10.5 remote-as 300
neighbor 203.250.13.41 remote-as 100

!Router C
hostname RouterC
!
interface loopback 0
ip address 128.213.63.130 255.255.255.192
!
interface serial 2/0
ip address 128.213.63.5 255.255.255.252
!
interface serial 2/1
ip address 128.213.63.2 255.255.255.252
!
router bgp 200
network 128.213.0.0
neighbor 128.213.63.1 remote-as 100
neighbor 128.213.63.6 remote-as 400

!Router D
hostname RouterD
!
interface loopback 0
ip address 192.208.10.174 255.255.255.192
!
interface serial 0/0
ip address 192.208.10.5 255.255.255.252
!
interface serial 0/0
ip address 192.208.10.5 255.255.255.252
!
router bgp 300
network 192.208.10.0
neighbor 192.208.10.1 remote-as 500
neighbor 192.208.10.6 remote-as 100

!Router E
hostname RouterE
!
interface loopback 0
ip address 200.200.10.1 255.255.255.0
```

```
interface serial 0
ip address 195.211.10.2 255.255.255.252
!
interface serial 1
ip address 128.213.63.6 255.255.255.252
!
router bgp 400
network 200.200.10.0
neighbor 128.213.63.5 remote-as 200
neighbor 195.211.10.1 remote-as 500

!Router F
hostname RouterF
!
interface ethernet 0
ip address 203.250.14.2 255.255.255.0
!
interface serial 1
ip address 203.250.15.1 255.255.255.252
!
router ospf 10
network 203.250.0.0 0.0.255.255 area 0

!Router G
hostname RouterG
!
interface loopback 0
ip address 195.211.10.174 255.255.255.192
!
interface serial 0
ip address 192.208.10.1 255.255.255.252
!
interface serial 1
ip address 195.211.10.1 255.255.255.252
!
router bgp 500
network 195.211.10.0
neighbor 192.208.10.2 remote-as 300
neighbor 195.211.10.2 remote-as 400
```

When you redistribute IGP routes into BGP, you need to control the routes that are injected into BGP. For that reason, it is always better to advertise routes by using the **network** router configuration command or by redistributing static routes, as shown in the examples in this section. This method also avoids route flaps.

Determining the State of BGP

Assume that in Figure 12-34 the connection between Routers B and D is down. The following information is displayed when you enter the **show ip bgp** EXEC command on Router B:

```
RouterB# show ip bgp
table version is 4, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network          Next Hop          Metric LocPrf Weight Path
*i128.213.0.0      128.213.63.2             0    100        0  200 i
*i192.208.10.0     128.213.63.2                  100        0  200 400 500 300 i
*i195.211.10.0     128.213.63.2                  100        0  200 400 500 i
*i200.200.10.0     128.213.63.2                  100        0  200 400 i
*>i203.250.13.0    203.250.13.41            0    100        0  i
*>i203.250.14.0    203.250.13.41            0    100        0  i
*> 203.250.15.0    0.0.0.0                  0           32768  i
```

The letter i at the beginning of a line means that the entry was learned via an internal BGP peer. The letter i at the end of a line indicates that the path information comes from an IGP. The first entry reads as follows: Network 128.213.0.0 is learned via path 200 and has a next hop of 128.213.63.2. Note that any locally generated entry, such as 203.250.15.0 has a next hop of 0.0.0.0.

The > symbol indicates that BGP has chosen the best route based on the decision steps described in the section “Summary of the BGP Path Selection Process,” earlier in this chapter. BGP picks only the one route that it determines to be the best route. It installs this route in the IP routing table and advertises it to other BGP peers. Note the next hop attribute of 128.213.63.2, which is the EBGP next hop carried into IBGP.

Following is the contents of the IP routing table on Router B:

```
RouterB# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/75] via 203.250.15.1, 02:50:45, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 02:40:46, Serial0
```

Note than none of the BGP entries appears in the IP routing table. One problem is that the next hop for these entries (128.213.63.2) is unreachable. This is because there is no way to reach that next hop via the IGP (in this case, OSPF). Router B has not learned about 128.213.63.0 via OSPF.

Correcting Next Hop Problems

For the network shown in Figure 12-34, the next hop problem can be corrected in one of two ways:

- On Router A, use the **neighbor next-hop-self** router configuration command to change the next hop between Router A and Router B.
- On Router A, run OSPF on interface serial 0 and make it passive. This way, Router B will know how to reach the next hop 128.213.63.2.

The following configuration for Router A runs OSPF on interface serial 0 and makes it passive:

```
!Router A
hostname RouterA
!
interface loopback 0
ip address 203.250.13.41 255.255.255.0
!
interface ethernet 0
ip address 203.250.14.1 255.255.255.0
!
interface serial 0
ip address 128.213.63.1 255.255.255.252
!
router ospf 10
passive-interface serial 0
network 203.250.0.0 0.0.255.255 area 0
network 128.213.0.0 0.0.255.255 area 0
!
router bgp 100
network 203.250.13.0 mask 255.255.255.0
network 203.250.14.0 mask 255.255.255.0
neighbor 128.213.63.2 remote-as 200
neighbor 203.250.15.2 remote-as 100
neighbor 203.250.15.2 update-source loopback 0
```

Now the BGP neighbor table on Router B contains the following routes:

```
RouterB# show ip bgp
table version is 10, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network          Next Hop          Metric LocPrf Weight Path
*>i128.213.0.0      128.213.63.2             0    100      0 200 i
*>i192.208.10.0     128.213.63.2                  100      0 200 400 500 300 i
*>i195.211.10.0     128.213.63.2                  100      0 200 400 500 i
*>i200.200.10.0     128.213.63.2                  100      0 200 400 i
*>i203.250.13.0     203.250.13.41            0    100      0 i
*>i203.250.14.0     203.250.13.41            0    100      0 i
*> 203.250.15.0     0.0.0.0                  0         32768 i
```

Note that a > symbol appears in all of the entries, which means that BGP is satisfied with the next hop address.

Now the IP routing table on Router B contains the following routes:

```
RouterB# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/75] via 203.250.15.1, 00:04:46, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:04:46, Serial0
     128.213.0.0 255.255.255.252 is subnetted, 1 subnets
O       1.28.213.63.0 [110/138] via 203.250.15.1, 00:04:47, Serial 0
```

Note that the BGP entries still do not appear in the IP routing table. The only difference is that 128.213.63.0 is now reachable via OSPF. The problem is synchronization: BGP is not synchronized with the IGP, so it does not put the entries in the IP routing table, and it does not send the entries in BGP updates. Router F is not aware of networks 192.208.10.0 or 195.211.10.0 because BGP routes are not redistributed into OSPF yet.

Turning Off Synchronization

If you enter the **no synchronization** router configuration command on Router B and then examine the IP routing table on Router B, you see the following contents of the IP routing table on Router B:

```
RouterB# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort not set

B    200.200.10.0 [200/0] via 128.213.63.2, 00:01:07
B    195.211.10.0 [200/0] via 128.213.63.2, 00:01:07
B    192.208.10.0 [200/0] via 128.213.63.2, 00:01:07
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/75] via 203.250.15.1, 00:12:37, Serial0
B       203.250.13.0 255.255.255.0 [200/0] via 203.250.13.41, 00:01:08
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:12:37, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
B       128.213.0.0 255 255.0.0 [200/0] via 128.213.63.2, 00:01:08
O       128.213.63.0 255.255.255.252
           [110/138] via 203.250.15.1, 00:12:37, Serial0
```

The routing table looks fine, but there is no way to reach those networks because Router F in the middle does not know how to reach them, as shown by the following output of the **show ip route** EXEC command on Router F:

```
RouterF# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/11] via 203.250.14.1, 00:14:15, EthernetO
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Seriall
C    203.250.14.0 is directly connected, EthernetO
     128.213.0.0 255.255.255.252 is subnetted, 1 subnets
O       128.213.63.0 [110/74] via 203.250 14 1, 00:14:15, EthernetO
```

If packets to the BGP network are forwarded to Router F, they will be dropped, so turning off synchronization does not resolve this particular problem. OSPF still needs to be redistributed into BGP on Router A so that Router F learns about BGP routes.

Redistributing OSPF

The following configuration for Router A has been modified to redistribute OSPF (the new command is in bold):

```
!Router A
hostname RouterA
!
interface loopback 0
ip address 203.250.13.41 255.255.255.0
```

```
interface ethernet 0
ip address 203.250.14.1 255.255.255.0
!
interface serial 0
ip address 128.213.63.1 255.255.255.252
!
router ospf 10
redistribute bgp 100 metric 2000 subnets
passive-interface serial 0
network 203.250.0.0 0.0.255.255 area 0
network 128.213.0.0 0.0.255.255 area 0
!
router bgp 100
network 203.250.0.0 mask 255.255.0.0
neighbor 128.213.63.2 remote-as 200
neighbor 203.250.15.2 remote-as 100
neighbor 203.250.15.2 update-source loopback 0
```

Now the routing table looks as follows:

```
RouterB# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort not set
O E2 200.200.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 195.211.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 192.208.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/75] via 203.250.15.1, 00:00:15, Serial0
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.15.1, 00:00:15, Serial0
        203.250.15.0 255.255.255.252 is subnetted, 2 subnets
C          203.250.15.8 is directly connected, Loopbackl
C          203.250.15.0 is directly connected, Serial0
O       203.250.14.0 [110/74] via 203.250.15.1, 00:00:15, Serial0
        128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2       128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:00:l5,Serial0
O       128.213.63.0 255.255.255.252
           [110/138] via 203.250.15.1, 00:00:16, Serial0
```

The BGP entries have disappeared because OSPF has a better distance (110) than IBGP (200).

Turning off synchronization on Router A will cause Router A to advertise network 203.250.15.0. This step is required because Router A will not synchronize with OSPF because of mask differences. For the same reason, synchronization should also be turned off on Router B so that it can advertise network 203.250.13.0.

In addition, OSPF should be enabled on interface serial 1 on Router B and made passive so that Router A learns about next hop 192.208.10.5 via an IGP.

The modified configurations for Routers A and B are as follows. (New commands are in bold.)

```
!Router A
hostname RouterA
!
interface loopback 0
ip address 203.250.13.41 255.255.255.0
!
interface ethernet 0
ip address 203.250.14.1 255.255.255.0
!
interface serial 0
ip address 128.213.63.1 255.255.255.252
!
router ospf 10
redistribute bgp 100 metric 2000 subnets
passive-interface serial 0
network 203.250.0.0 0.0.255.255 area 0
network 128.213.0.0 0.0.255.255 area 0
!
router bgp 100
no synchronization
network 203.250.13.0 mask 255.255.255.0
network 203.250.14.0 mask 255.255.255.0
neighbor 128.213.63.2 remote-as 200
neighbor 203.250.15.2 remote-as 100
neighbor 203.250.15.2 update-source loopback 0

!Router B
hostname RouterB
!
interface serial 0
ip address 203.250.15.2 255.255.255.252
!
interface serial 1
ip address 192.208.10.6 255.255.255.252
!
router ospf 10
redistribute bgp 100 metric 1000 subnets
passive-interface serial 1
network 203.250.0.0 0.0.255.255 area 0
network 192.208.0.0 0.0.255.255 area 0
!
router bgp 100
network 203.250.15.0
neighbor 192.208.10.5 remote-as 300
neighbor 203.250.13.41 remote-as 100
```

Now bring up interface serial 1 on Router B and see what the BGP neighbor table looks like on Router A:

```
RouterA# show ip bgp
table version is 117, local router ID is 203.250.13.41
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network          Next Hop          Metric LocPrf Weight Path
*> 128.213.0.0      128.213.63.2             0    100      0 200 i
*>i192.208.10.0     192.208.10.5             0    100      0 300 i
*>i195.211.10.0     192.208.10.5                  100      0 300 500 i
*                   128.213.63.2                           0 200 400 500 i
*> 203.250.13.0     0.0.0.0                  0         32768 i
*> 203.250.14.0     0.0.0.0                  0         32768 i
*>i203.250.15.0     203.250.15.2             0    100      0 i
```

Following is the output of the **show ip route** EXEC command on Router A:

```
RouterA# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort not set

     192.208.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.208.10.0 255.255.255.0
           [110/1000] via 203.250.14.2, 00:41:25, Ethernet0
O    192.208.10 4 255.255.255.252
           [110/138] via 203.250.14.2, 00:41:25, Ethernet0
C    203.250.13.0 is directly connected, Loopback0
     203.250.15.0 is variably subnetted, 3 subnets, 3 masks
O       203.250.15.10 255.255.255.255
           [110/75] via 203.250.14.2, 00:41:25, Ethernet0
O       203.250.15.0 255.255.255.252
           [110/74] via 203.250.14.2, 00:41:25, Ethernet0
B       203.250.15.0 255.255.255.0 [200/0] via 203.250.15.2, 00:41:25
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
B       128.213.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:41:26
C       128.213.63.0 255.255.255.252 is directly connected, Serial0
B*   200.200.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:02:38
```

Following is the output of the **show ip bgp** EXEC command on Router B:

```
RouterB# show ip bgp
table version is 12, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network          Next Hop          Metric LocPrf Weight Path
*>i128.213.0.0        128.213.63.2           0    100      0 200 i
*                     192.208.10.5                         0 300 500 400 200 i
*> 195.208.10.0       192.208.10.5           0             0 300 i
*> 195.211.10.0       192.208.10.5                         0 300 500 i
*>i200.200.10.0       128.213.63.2                100      0 200 400 i
*>                    192.208.10.5                         0 300 500 400 i
*>i203.250.13.0       203.250.13.41          0    100      0 i
*>i203.250.14.0       203.250.13.41          0    100      0 i
*> 203.250.15.0       0.0.0.0                0         32768 i
```

Managing Asymmetry

There are several ways to design the network for AS 100 to communicate with the ISP networks in AS 200 and AS 300. One way is to have a primary ISP and a backup ISP. AS 100 could learn partial routes from one of the ISPs and default routes to both ISPs. In this example, AS 100 receives partial routes from AS 200 and only local routes from AS 300. Both Routers A and B generate default routes into OSPF, with Router B being the more preferred route because of its lower MED attribute. This allows you to balance outgoing traffic between the two ISPs.

Potential asymmetry might occur if traffic going out from Router A comes back via Router B. This might occur if networks are advertised to both of the ISPs. From outside the AS, the networks are reachable via both of the ISPs and either Router A or B could be used to reach them. You might find out that all incoming traffic to your AS is coming via one single point even though you have multiple points to the internetwork.

One other potential reason for asymmetry is the different advertised path length to reach your AS. One ISP might be closer to a certain destination than another. In this example, traffic from AS 400 destined for AS 100 always comes in via Router A because of the shorter path. You might try to affect that decision by using the **set as-path route** map configuration command with the **prepend** keyword to prepend AS numbers to your updates to make the AS_path attribute longer. But, if AS 400 has somehow set its exit point to be via AS 200 based on attributes such as local preference, MED attribute, weight, there is nothing you can do.

Final Configurations

Following is the final configuration for Router A. (New or modified commands are in bold.)

```
!Router A
hostname RouterA
!
interface loopback 0
ip address 203.250.13.41 255.255.255.0
!
interface ethernet 0
ip address 203.250.14.1 255.255.255.0
!
interface serial 0
ip address 128.213.63.1 255.255.255.252
!
router ospf 10
redistribute bgp 100 metric 2000 subnets
passive-interface serial 0
network 203.250.0.0 0.0.255.255 area 0
network 128.213.0.0 0.0.255.255 area 0
default-information originate metric 2000
!
router bgp 100
no synchronization
neighbor 128.213.63.2 remote-as 200
neighbor 128.213.63.2 route-map setlocalpref in
neighbor 203.250.15.2 remote-as 100
neighbor 203.250.15.2 update-source loopback 0
!
ip default-network 200.200.0.0
!
route-map setlocalpref permit 10
set local-preference 200
```

The final configuration for Router A sets the local preference for routes coming from AS 200 to 200. The configuration also uses the **ip default-network** global configuration command to specify network 200.200.0.0 as the candidate default route. The **ip default-information originate** router configuration command is used to inject the default route inside the OSPF domain. For RIP, network 0.0.0.0 is automatically redistributed into RIP without additional configuration. For IGRP and Enhanced IGRP, default information is injected into the IGP domain after BGP is redistributed. Also, with IGRP and Enhanced IGRP, you can redistribute a static route for 0.0.0.0 into the IGP domain.

Following is the final configuration for Router B. (New or modified commands are in bold.)

```
!Router B
hostname RouterB
!
interface serial 0
ip address 203.250.15.2 255.255.255.252
!
interface serial 1
ip address 192.208.10.6 255.255.255.252
```

```
router ospf 10
redistribute bgp 100 metric 1000 subnets
passive-interface serial 1
network 203.250.0.0 0.0.255.255 area 0
network 192.208.0.0 0.0.255.255 area 0
default-information originate metric 1000
!
router bgp 100
no synchronization
network 203.250.15.0
neighbor 192.208.10.5 remote-as 300
neighbor 192.208.10.5 route-map LOCALONLY in
neighbor 203.250.13.41 remote-as 100
!
ip default-network 192.208.10.0
ip as-path access-list 1 permit ^300 500$
ip as-path access-list 2 permit ^300$
!
route-map LOCALONLY permit 10
match as-path 1
set local-preference 300
!
route-map LOCALONLY permit 20
match as-path 2
```

The configuration for Router B sets the local preference for updates coming from AS 300 having an AS_path attribute of 300, 500 to 300, which is higher than the IBGP updates coming in from Router A in AS 100. This way, AS 100 will pick Router B for AS 500's local routes. Any other routes on Router B (if there are any) will be sent internally with a local preference of 100, which is lower than the local preference of 200 coming in from Router A. This arrangement causes Router A to be preferred. Further, because of the length of the AS_path attribute, Router B is used to reach routes local to AS 300.

Note that Router B accepts the local routes of AS 300 and AS 500 only. Any updates whose AS_path attribute does not match are dropped. If you want to advertise the local routes and the neighbor routes (customers of the ISP), you can use ^300_[0-9]* as the regular expression. The following is the output of the **show ip bgp** EXEC command for regular expression ^300$:

```
RouterB# show bgp regexp ^300$
BGP table version is 14, local router ID is 203.250.15.2
Status code: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 192.208.10.0     192.28.10.5            0    300      0 300
```

Following is the final configuration for Router C. (New and modified commands are in bold.)

```
!Router C
hostname RouterC
!
interface loopback 0
ip address 128.213.63.130 255.255.255.192
!
interface serial 2/0
ip address 128.213.63.5 255.255.255.252
!
interface serial 2/1
ip address 128.213.63.2 255.255.255.252
!
router bgp 200
network 128.213.0.0
aggregate-address 128.213.0.0 255.255.0.0 summary-only
```

```
neighbor 128.213.63.1 remote-as 100
neighbor 128.213.63.1 distribute-list 1 out
neighbor 128.213.63.6 remote-as 400
!
access-list 1 deny 195.211.0.0 0.0.255.255
access-list 1 permit any
```

The configuration for Router C aggregates network 128.213.0.0/16 and specifies the routes that are to be injected into AS 100. If the ISP refuses to do this task, you have to filter routes coming into AS 100 on Router A.

Following are the final configurations for Routers D and E. (New or modified commands are in bold.)

```
!Router D
hostname RouterD
!
interface loopback 0
ip address 192.208.10.174 255.255.255.192
!
interface serial 0/0
ip address 192.208.10.5 255.255.255.252
!
interface serial 0/1
ip address 192.208.10.2 255.255.255.252
!
router bgp 300
network 192.208.10.0
neighbor 192.208.10.1 remote-as 500
neighbor 192.208.10.6 remote-as 100

!Router E
hostname RouterE
!
interface loopback 0
ip address 200.200.10.1 255.255.255.0
interface serial 0
ip address 195.211.10.2 255.255.255.252
!
interface serial 1
ip address 128.213.63.6 255.255.255.252
!
router bgp 400
network 200.200.10.0
aggregate-address 200.200.0.0 255.255.0.0 summary-only
neighbor 128.213.63.5 remote-as 200
neighbor 195.211.10.1 remote-as 500
```

Router E is aggregating network 200.200.0.0/16.

Following are the final configurations for Routers F and G. (New or modified commands are in bold.)

```
!Router F
hostname RouterF
!
interface ethernet 0
ip address 203.250.14.2 255.255.255.0
!
interface serial 1
ip address 203.250.15.1 255.255.255.252
!
router ospf 10
network 203.250.0.0 0.0.255.255 area 0
!Router G
hostname RouterG
```

```
!
interface loopback 0
ip address 195.211.10.174 255.255.255.192
!
interface serial 0
ip address 192.208.10.1 255.255.255.252
!
interface serial 1
ip address 195.211.10.1 255.255.255.252
!
router bgp 500
network 195.211.10.0
aggregate-address 195.211.0.0 255.255.0.0 summary-only
neighbor 192.208.10.2 remote-as 300
neighbor 192.208.10.2 send-community
neighbor 192.208.10.2 route-map setcommunity out
neighbor 195.211.10.2 remote-as 400
!
access-list 2 permit any
access-list 101 permit ip 195.211.0.0 0.0.255.255 255.255.255.0 0.0.0.255
!
route-map setcommunity permit 10
match ip address 101
set community no-export
!
route-map setcommunity permit 20
match ip address 2
```

The configuration for Router G demonstrates the use of community filtering by adding the no-export community to more specific Class C routes of 195.211.0.0/16 that are sent to Router D. This way, Router D will not export that route to Router B.

Following is the final content of BGP routing table on Router A:

```
RouterA# show ip bgp
table version is 21, local router ID is 203.250.13.41
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network          Next Hop          Metric LocPrf Weight Path
*> 128.213.0.0        128.213.63.2           0    200      0 200 i
*>i192.208.10.0       192.208.10.5           0    300      0 300 i
*> 200.200.0.0/16     128.213.63.2                200      0 200 400 i
*> 203.250.13.0       0.0.0.0                0         32768 i
*> 203.250.14.0       0.0.0.0                0         32768 i
*>i203.250.15.0       203.250.15.2           0    100      0 i
```

Following is the final content of the IP routing table on Router A:

```
RouterA# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 128.213.63.2 to network 200.200.0.0

     192.208.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.208.10.0 255.255.255.0
           [110/1000] via 203.250.14.2, 00:41:25, Ethernet0
O       192.208.10.4 255.255.255.252
           [110/138] via 203.250.14.2, 00:41:25, Ethernet0
C    203.250.13.0 is directly connected, Loopback0
     203.250.15.0 is variably subnetted, 3 subnets, 3 masks
O       203.250.15.10 255.255.255.255
           [110/75] via 203.250.14.2, 00:41:25, Ethernet0
O       203.250.15.0 255.255.255.252
           [110/74] via 203.250.14.2, 00:41:25, Ethernet0
B       203.250.15.0 255.255.255.0 [200/0] via 203.250.15.2, 00:41:25
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
B       128.213.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:41:26
C       128.213.63.0 255.255.255.252 is directly connected, Serial0
B*   200.200.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:02:38
```

Following is the final content of IP routing table on Router F:

```
RouterF# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 203.250.15.2 to network 0.0.0.0

     192.208.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.208.10.0 255.255.255.0
           [110/1000] via 203.250.15.2, 00:48:50, Serial1
O       192.208.10.4 255.255.255.252
           [110/128] via 203.250.15.2, 01:12:09, Serial1
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/11] via 203.250.14.1, 01:12:09, Ethernet0
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.14.1, 01:12:09, Ethernet0
     203.250.15.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.15.10 255.255.255.255
           [110/65] via 203.250.15.2, 01:12:09, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 255.255.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0
           [110/2000] via 203.250.14.1, 00:45:01, Ethernet0
O E2 200.200.0.0 255.255.0.0 [110/1000] via 203.250.14.1, 00:03:47, Ethernet0
O*E2 0.0.0.0 0.0.0.0 [110/1000] via 203.250.15.2, 00:03:33, Serial1
```

Note that on Router F, the routing table indicates that networks local to AS 300, such as 192.208.10.0 are to be reached via Router B. Other known networks, such as 200.200.0.0 are to be reached via Router A. The gateway of last resort is set to Router B. If something happens to the connection between Router B and Router D, the default advertised by Router A will kick in with a MED attribute of 2000.

Following is the final content of BGP routing table on Router B:

```
RouterB# show ip bgp
table version is 14, local router ID is 203.250.15.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network          Next Hop          Metric LocPrf Weight Path
*>i128.213.0.0      128.213.63.2             0    200      0 200 i
*> 192.208.10.0     192.208.10.5             0    300      0 300 i
*>i200.200.0.0/16   128.213.63.2                  200      0 200 400 i
*>i203.250.13.0     203.250.13.41            0    100      0 i
*>i203.250.14.0     203.250.13.41            0    100      0 i
*> 203.250.15.0     0.0.0.0                  0         32768 i
```

Following is the final content of the IP routing table on Router B:

```
RouterF# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 203.250.15.2 to network 192.208.10.0

*    192.208.10.0 is variably subnetted, 2 subnets, 2 masks
B*      192.208.10.0 255.255.255.0 [20/0] via 192.208.10.5, 00:50:46
C       192.208.10.4 255.255.255.252 is directly connected, Serial1
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/75] via 203.250.15.1, 01:20:33, Serial0
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.15.1, 01:15:40, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 2 subnets, 2 masks
O       203.250.15.10 255.255.255.255
           [110/65] via 203.250.15.2, 01:12:09, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 255.255.0.0 is variably subnetted, 2 subnets
C       203.250.15.8 id directly connected, Loopback1
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 01:20:33, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:46:55, Serial0
```

Summary

The primary function of a BGP system is to exchange network reachability information with other BGP systems. This information is used to construct a graph of AS connectivity from which routing loops are pruned and with which AS-level policy decisions are enforced. BGP provides a number of techniques for controlling the flow of BGP updates, such as route, path, and community filtering. It also provides techniques for consolidating routing information, such as CIDR aggregation, confederations, and route reflectors. BGP is a powerful tool for providing loop-free interdomain routing within and between ASs.