CHAPTER 11

Hybrid: Enhanced Interior Gateway Routing Protocol (EIGRP)

As internetworks grew in scale and diversity in the early 1990s, new routing protocols were needed. Cisco developed *Enhanced Interior Gateway Routing Protocol (IGRP)* primarily to address many of the limitations of IGRP and RIP. As WANs were growing, so was the need for a routing protocol that would use efficient address space on WAN links, as well as the LAN networks. OSPF was available, but the CPU-intensive tasks that it had to perform often overloaded the small processors of many edge or remote routers of that time. The configuration was also more complex than that of RIP or IGRP. A routing protocol was needed that could support VLSM and that could scale with large internetworks, yet that was less CPU-intensive than OSPF. In 1994, Cisco answered the call by releasing Enhanced IGRP in Cisco IOS Software Release 9.21. Today, EIGRP is used as the routing protocol on many large government and commercial internetworks. It has proven to be very stable, flexible, and fast. In addition to these characteristics, the ease of EIGRP configuration makes it one of the most popular routing protocols among network engineers.

EIGRP can be referred to as a hybrid protocol. It combines most of the characteristics of traditional distance vector protocols with some characteristics of link-state protocols. Specifically, EIGRP is "enhanced" by using four routing technologies:

- Neighbor discovery/recovery
- Reliable Transport Protocol (RTP)
- DUAL finite-state machine
- Protocol-dependent modules

This chapter covers these technologies, as well as the operation and configuration of EIGRP.

Technical Overview of EIGRP

EIGRP offers many advantages over other routing protocols, including the following:

- **Support for VLSM**—EIGRP is a classless routing protocol and carries the subnet mask of the route in its update.
- **Rapid convergence**—By using the concept of feasible successors, defined by DUAL, EIGRP is capable of preselecting the next best path to a destination. This allows for very fast convergence upon a link failure.
- **Low CPU utilization**—Under normal operation, only hellos and partial updates are sent across a link. Routing updates are not flooded and are processed only periodically.
- **Incremental updates**—EIGRP does not send a full routing update; it sends only information about the changed route.
- **Scalable**—Through the use of VLSM and a complex composite metric, EIGRP networks can be vast in size.
- **Easy configuration**—EIGRP supports hierarchical network design, but it does not require the strict configuration guidelines, such as the ones needed for OSPF.
- **Automatic route summarization**—EIGRP will perform automatic summarization on major bit boundaries.
- **MD5 route authentication**—As of Cisco IOS Software Release 11.3, EIGRP can be configured to perform MD5 password authentication on route updates.

Looking at this list, it becomes evident why EIGRP has become a popular routing protocol. It provides many of the enhancements of OSPF, without the strict configuration guidelines. It could be argued that EIGRP's weakest point is that it is a Cisco-proprietary protocol, but with the aid of redistribution, this point becomes moot.

EIGRP is a classless routing protocol. It directly interfaces to IP as protocol 88. EIGRP uses the multicast address of 224.0.0.10 for hellos and routing updates instead of an all-hosts broadcast like RIP uses. EIGRP also employs a system of hello and hold timers to maintain neighbors. Aside from the initial routing update, partial routing updates are sent only when network topology changes occur. The updates are also bounded, which means that updates are sent only to pertinent routers. Like IGRP, EIGRP uses a composite metric to calculate the best path to a destination. The sections that follow take a closer look at how EIGRP makes use of metrics, neighbors, reliable transport, and DUAL in its operation.

NOTE Early releases of EIGRP had stability issues over low-speed serial links and problems maintaining many neighbors. Cisco significantly enhanced EIGRP with Cisco IOS Software Releases 10.3(11), 11.0(8), and 11.1(3)— early releases of EIGRP are sometimes referred to as EIGRP version 1. Cisco currently ships routers with IOS 12.0 and above.

EIGRP Metrics

EIGRP uses metrics in the same way as IGRP. Each route in the route table has an associated metric. EIGRP uses a composite metric much like IGRP, except that it is modified by a multiplier of 256. Recall from Chapter 10, "Distance Vector Protocols: Interior Gateway Routing Protocol (EIGRP)," that bandwidth, delay, load, reliability, and MTU are the submetrics. Like IGRP, EIGRP chooses a route based primarily on bandwidth and delay, or the composite metric with the lowest numerical value. When EIGRP calculates this metric for a route, it calls it the *feasible distance* to the route. EIGRP calculates a feasible distance to all routes in the network. The following list is a detailed description of the five EIGRP submetrics:

- **Bandwidth**—Bandwidth is expressed in units of kilobits. It must be statically configured to accurately represent the interfaces that EIGRP is running on. For example, the default bandwidth of a 56-kbps interface and a T1 interface is 1544 kbps. To accurately adjust the bandwidth, use the **bandwidth** *kbps* interface subcommand. Table 11-1 highlights some common bandwidth values.
- **Delay**—Delay is expressed in microseconds. It, too, must be statically configured to accurately represent the interface that EIGRP is running on. The delay on an interface can be adjusted with the **delay** *time_in_microseconds* interface subcommand. Common delay values are represented in Table 11-1.
- **Reliability**—Reliability is a dynamic number in the range of 1 to 255, where 255 is a 100 percent reliable link and 1 is an unreliable link.
- **Load**—Load is the number in the range of 1 to 255 that shows the output load of an interface. This value is dynamic and can be viewed using the **show interfaces** command. A value of 1 indicates a minimally loaded link, whereas 255 indicates a 100 percent loaded link.
- **MTU**—The maximum transmission unit (MTU) is the recorded smallest MTU value in the path, usually 1500.

NOTE Whenever you are influencing routing decisions in IGRP or EIGRP, use the metric of delay over bandwidth. Changing bandwidth can affect other routing protocols, such as OSPF. Changing delay affects only IGRP and EIGRP.

Table 11-1 highlights the common metrics used.

Table 11-1 *Common IGRP and EIGRP Metrics*

Medium	Bandwidth	Delay
100-Mbps ATM	100,000 kbps	100 µs
Gigabit Ethernet	100,000 kbps	100 µs
Fast Ethernet	100,000 kbps	100 µs
FDDI	100,000 kbps	100 µs
HSSI	45,045 kbps	20,000 µs
16-Mbps Token Ring	16,000 kbps	630 µs
10-Mbps Ethernet	10,000 kbps	1000 µs
T1	1544 kbps	20,000 µs
DS-0	64 kbps	20,000 µs
56-kbps media	56 kbps	20,000 µs

EIGRP uses a composite metric (CM) that is derived from the five submetrics. When EIGRP computes the composite metric, it uses a formula that involves five constants or "k" values. The constant values have default value such as the following:

$$k1 = k3 = 1 \text{ and } k2 = k4 = k5 = 0$$

By setting k2, k4, and k5 to 0, it essentially nullifies the submetrics of load, reliability, and MTU. This is precisely why you should first use delay and then bandwidth when trying to influence which routes EIGRP prefers. The formula EIGRP uses to calculate the composite metric is as follows:

$$CM = 256 \times ([k1 \times BW_{mim} + (k2 \times BW_{mim}) / (256\text{-}LOAD) + k3 \times DELAY_{sum}] \times X)$$

where the following is true:

$$BW_{mim} = 10^7 / bandwidth\_of\_slowest\_link$$
$$DELAY_{sum} = \Sigma\,(delays\_along\_the\_path)$$
$$X = k5 / (reliability + k4) \text{ if and only if } k1<>1, \text{ if } k1 = 1 \text{ then } X = 1$$

With the k values set at the default value you have

$$k1 = k3 = 1$$
$$k2 = k4 = k5 = 0$$
$$CM = 256 \times (BW_{mim} + DELAY_{sum})$$

NOTE The router calculation of the composite metric will always differ slightly from the result when it is performed by longhand. This is because of the way the router handles floating-point mathematics; there will be slight rounding discrepancies.

Using the default values of constants, k1 = k3 = 1 and k2 = k4 = k5 =0, the formula quickly breaks down to this:

$$(256 \times [BW_{mim} \text{ and } DELAY_{sum}])$$

Substituting the constants, you have the following:

$$CM = 256 \times ([1 \times BW_{mim} + (0 * BW_{mim}) / (256\text{-}LOAD) + 1 \times DELAY_{sum}] \times 1)$$
$$CM = 256 \times ([BW_{mim} + (0) / (256\text{-}LOAD) + DELAY_{sum}] \times 1)$$
$$CM = 256 \times (BW_{mim} + DELAY_{sum})$$

NOTE For reference, the metric is computed the same way for IGRP, except the result of bandwidth and delay is not multiplied by 256, and the $\mathbf{DELAY_{sum}}$ variable is divided by 10.

$CM = (k1 \times BW_{min} + [k2 \times BW_{min}] / [256\text{-}LOAD] + [k3 \times DELAY_{sum}] \times X)$

where the following is true:

$BW_{min} = 10^7$ / bandwidth_of_slowest_link

$DELAY_{sum}$ = S(delays_along_the_path) / 10

X = k5 / (reliability + k4) if and only if k1<>1, if k1=1 then X=1

k1=k3=1

k2=k4=k5=0

With k values set at the default value, you have:

$CM = BW_{min} + DELAY_{sum}$

To demonstrate composite metric calculation, refer to Figure 11-1. In this example, EIGRP calculates a composite metric on the alpha router to 172.16.1.0/24, which resides on the charlie router.

Assuming that the **bandwidth** statements been set by an astute engineer, the lowest bandwidth on the path between alpha and charlie routers would be 56. Therefore, you have

$$BW_{mim} = 10^7 / 56 = 178571$$

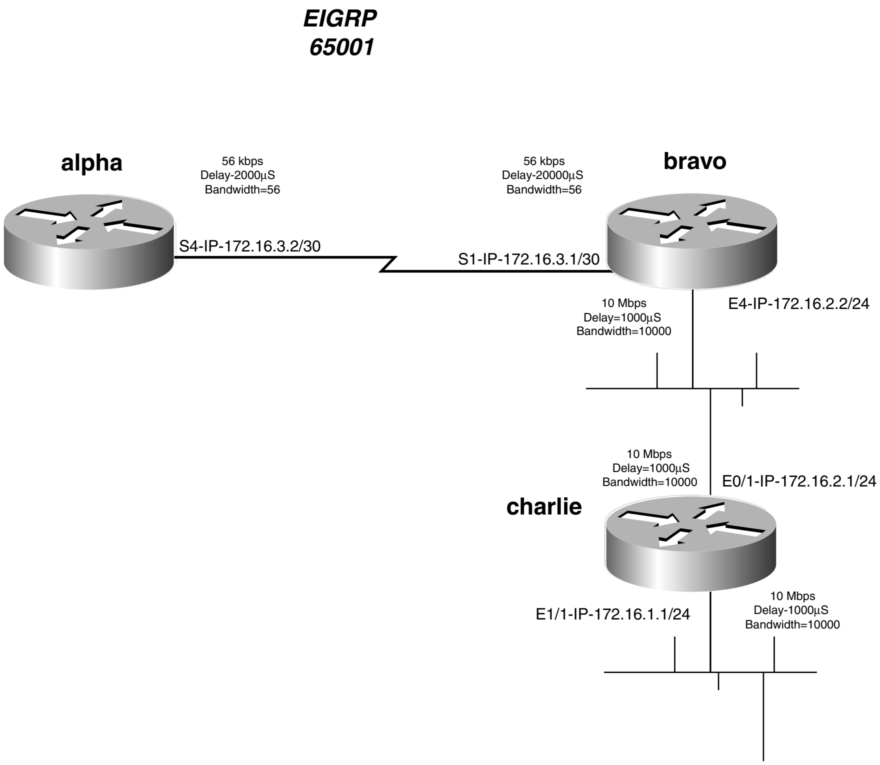
The delay is the summation of the delays on the outbound interfaces only. The summation ends with the delay on the interface in which the final subnet resides. From alpha to bravo, the delay is 20000; from bravo to charlie, it is 1000; this includes the final interface on charlie, which has a delay of 1000. Therefore, you have

$$DELAY_{sum} = 20000 + 1000 + 1000 = 22000$$

The composite metric now yields the following:

$$CM = 256 \times (178571) + 256 \times (22000) = 46277485$$

Figure 11-1 *EIGRP Routing Updates*



The submetrics and the composite metric can be confirmed by performing the **show ip route 172.16.1.0** command on the alpha router, as in Example 11-1. Remember, because of rounding errors, the metric does not match exactly.

Example 11-1 **show ip route** *Command Output Highlighting the EIGRP Metrics*

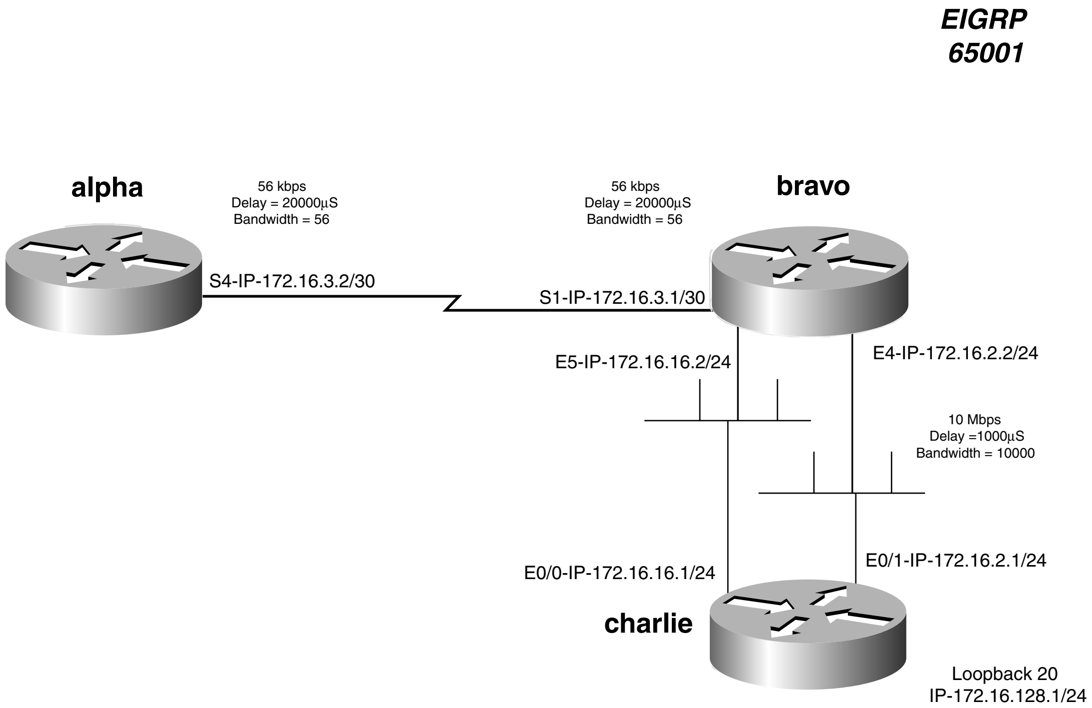
```
alpha#show ip route 172.16.1.0
Routing entry for 172.16.1.0/24
  Known via "eigrp 65001", distance 90, metric 46277376, type internal
  Redistributing via eigrp 65001
  Last update from 172.16.3.1 on Serial7, 00:50:53 ago
  Routing Descriptor Blocks:
  * 172.16.3.1, from 172.16.3.1, 00:50:53 ago, via Serial7
      Route metric is 46277376, traffic share count is 1
      Total delay is 22000 microseconds, minimum bandwidth is 56 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2

alpha#
```

When using metrics to influence routing decisions, use the **delay** *xx* interface command. Be sure to include a delay at each side of the interface if you want symmetrical routing—that is, packets will take the same route back to the source. By default, EIGRP will perform equal-cost load balancing over routes. For example, if you perform a **show ip route** command and see two routes to a destination reported, EIGRP will load-balance over those routes.

To demonstrate the use of the delay metric, we have added another Ethernet segment between the bravo and charlie routers and a loopback interface, 172.16.128.1/24, on the charlie router, as illustrated in Figure 11-2.

Figure 11-2 *EIGRP Load Sharing*



If you perform a **show ip route** command on the bravo router, as shown in Example 11-2, you see two routes to the 172.16.128.0/24 network. The **show ip eigrp topology** command also lists the routes and the composite metric to them.

Example 11-2 *Two Routes Reported to 172.16.128.0/24*

```
bravo#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.128.0/24 [90/409600] via 172.16.2.1, 00:23:50, Ethernet4
                        [90/409600] via 172.16.16.1, 00:23:50, Ethernet5
C       172.16.16.0/24 is directly connected, Ethernet5
C       172.16.2.0/24 is directly connected, Ethernet4
C       172.16.3.0/30 is directly connected, Serial1
bravo#
```

If you want EIGRP to prefer one path to the other, add the **delay** command on each side of the interface. It is important to note that changing the delay of a link will affect only the routing protocol, not the actual throughput of the link.

Continuing with the example, set the delay of the link so that the primary link to 172.16.128.0 will be through 172.16.16.1. This can be accomplished by adding a delay of 1000 to the e4 interface of the bravo router and under the e0/1 interface of the charlie router. Example 11-3 demonstrates the configuration of delay on the bravo router.

Example 11-3 *Addition of the* **delay** *Command*

```
bravo#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
bravo(config)#int e4
bravo(config-if)#delay 1000
bravo(config-if)#^Z
```

Example 11-4 shows the route table of the bravo router after the delay was added to the bravo and charlie routers.

Example 11-4 *One Route to the 172.16.128.0/24 Route*

```
bravo#show ip route
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.128.0/24 [90/409600] via 172.16.16.1, 00:00:11, Ethernet5
C       172.16.16.0/24 is directly connected, Ethernet5
C       172.16.2.0/24 is directly connected, Ethernet4
C       172.16.3.0/30 is directly connected, Serial1
bravo#
```

Keep in mind that although the second route is removed from the routing table, EIGRP still knows of the route and will keep it as a feasible successor.

The *k* values also can be manipulated to influence routing decisions. This can be accomplished with the **metric weights** tos *k1 k2 k3 k4 k5* command. Manipulating these values directly impacts how EIGRP derives the composite metric for all routes. Change the metric weights only when working with Cisco to solve specific problems.

EIGRP Neighbors

EIGRP does not periodically advertise it routes. Because of this, it needs some way to locate and then exchange routing information with adjacent devices. EIGRP accomplishes this through the use of neighbors. When EIGRP initializes, it sends out a multicast hello on address 224.0.0.10, on broadcast media. On NBMA media, X.25, Frame Relay, and ATM, the hellos are unicast every 60 seconds. EIGRP continues to send out hellos every few seconds, based on the media type. Specifically, EIGRP sends hellos every 5 seconds on the following interfaces:

- LAN broadcast media, such as Ethernet, Token Ring, and FDDI
- High-speed serial link greater than T1 speeds, such as Frame Relay HSSI links
- Point-to-point serial links, such as PPP or HDLC
- ATM and Frame Relay point-to-point subinterfaces

EIGRP sends hellos every 60 seconds on the following interfaces:

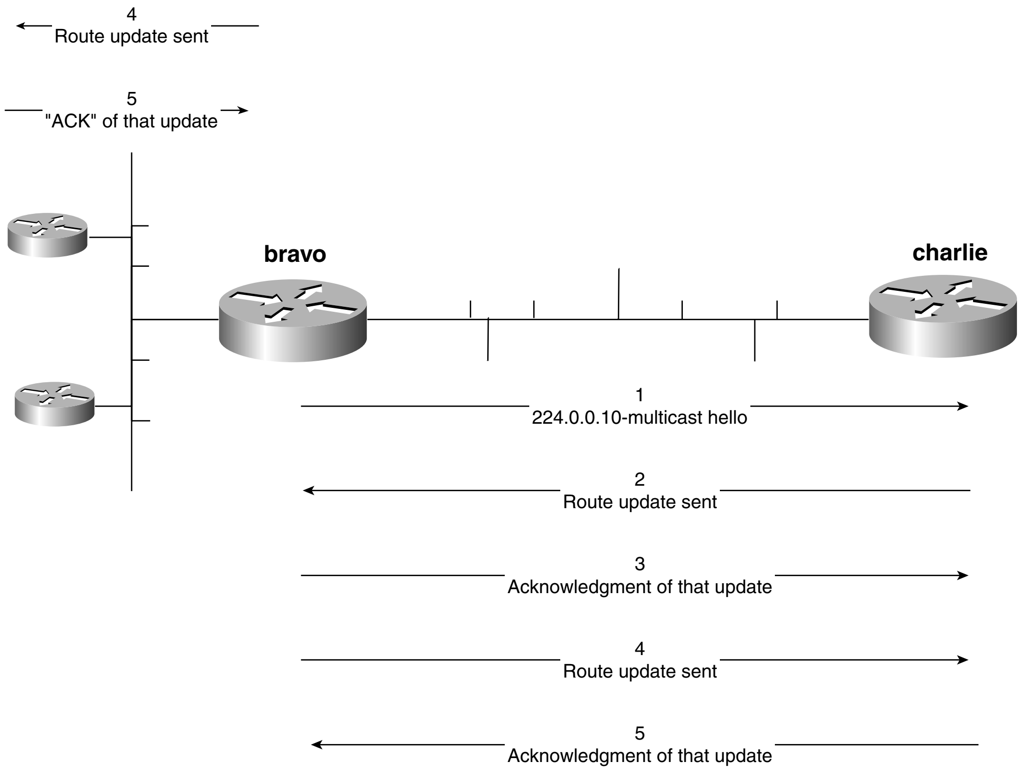
- Low-speed serial links less than T1 speeds, including Frame Relay and multipoint X.25
- ATM and Frame Relay multipoint interfaces, and ATM SVCs
- ISDN BRIs

Routers that reside on the same network receive the multicast hello and respond to form what is called an *adjacency*. Figure 11-3 and the list that follows describe the initial router exchange when forming an adjacency:

1. Hellos are sent out each interface participating in EIGRP, except interfaces quieted by the passive interfaces. All EIGRP hellos and routing updates use the multicast address of 224.0.0.10.
2. Routers on the same IP subnet receive the multicast and respond with a full routing update. This is accomplished by setting the INITialization bit in the EIGRP header; the updates include all networks that EIGRP is aware of and the metric for those routes, except for those suppressed by split horizon. This update packet establishes a neighbor relationship (adjacency). The hello packet also includes a *hold timer*, which tells the router how long it should wait before receiving a hello and declaring the route unreachable and reporting it to the DUAL process. The hold timer is set to three times the value assigned for the hello timer. This usually is 15 or 180 seconds, depending on the media.

3. The bravo router responds to the initialization packet by sending a hello with the ACK bit set. EIGRP sets the ACK bit to acknowledge all messages that it receives that have data. This is one way that EIGRP has reliable transports (discussed further in upcoming sections).
4. The bravo router now inserts the new update into its route table. Because it has a new update, it sends an update to all its neighbors.
5. The neighbors that received the update from the bravo message respond with an acknowledgment packet.
6. The router holds the adjacency by the continuous exchange of hellos. If a hello is not received by the time the hold timer expires, the router marks the route as unreachable.

Figure 11-3 *EIGRP Neighbor Establishment*



When the router forms an adjacency, it treats this as a virtual link to transport routing information.

The router begins to form a neighbor table with the following information:

- The IP address of the router that it received the hello from
- The hold timer
- The SRTT or round-trip time
- The uptime of the neighbor

The status of neighbors can be displayed with the **show ip eigrp neighbors** command, as in Example 11-5. The uptime of the neighbor should be for as long as the adjacency has been established.

Example 11-5 show ip eigrp neighbors *Command Output on the bravo Router*

```
bravo#show ip eigrp neighbors
IP-EIGRP neighbors for process 65001
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
1   172.16.2.1              Et4           12 01:10:36    8   200  0  29
2   172.16.16.1             Et5           13 02:14:15    3   200  0  28
0   172.16.3.2              Se1           11 07:07:44   23  2604  0  23
bravo#
```

Stable EIGRP neighbors are the single most important element in any EIGRP network. Without stable neighbors, an EIGRP network will have difficulty operating properly. Checking the status of EIGRP neighbors should be the first step in verifying the operational status of any EIGRP network.

EIGRP Reliable Transport Protocol (RTP)

RTP ensures that EIGRP packets are received, delivered, ordered, and acknowledged. To guarantee delivery, EIGRP employs the use of a Cisco proprietary reliable multicast message. When each neighbor receives a reliable multicast packet, it is required to respond with a unicast acknowledgment. Updates also have sequence numbers; this is how the router ensures that updates are in the proper order. To facilitate RTP and the other functions of EIGRP, Cisco uses four primary types of packets, even though there are actually five. As previously mentioned, all EIGRP packets directly interface with the IP layer as protocol 88, and the multicast updates use the IP address of 224.0.0.10. The five packet types are as follows:

- **Hello**—Used to discover and maintain neighbors. This packet type uses unreliable delivery.
- **Acknowledgments (ACKs)**—Used to acknowledge updates. They are essentially hellos with no data in them. ACKs also use unreliable delivery.

- **Updates**—Contain routing information. Updates can be either unicast or multicast, depending on how they are generated. Updates use reliable delivery.
- **Queries**—Used by the DUAL process to find feasible successor for routes. The query can be unicast or multicast. Queries always use reliable delivery.
- **Replies**—Used by the DUAL process to aid in finding feasible successor for routes. Replies are always unicast and use reliable delivery.

NOTE Some documentation refers to queries and replies as the four and fifth types of packets. The actual fifth type of packet is a request. The request never was implemented in EIGRP and was intended for route servers. IPX SAPs also use another Opcode in the EIGRP header, making them another packet type.

Diffusing Update Algorithm

The DUAL algorithm is the "brains" of EIGRP, responsible for tracking all routes by all neighbors and ensuring a loop-free topology. It is based on an algorithm first developed by E.W. Dijkstra and C.S. Scholten, and later enhanced by J.J. Garcia-Luna-Aceves.

With the help of DUAL, EIGRP and the processes previously covered, EIGRP keeps the following tables:

- **Neighbor table**—EIGRP tracks every formed adjacency in the neighbor table. A neighbor will be held until an ACK is not received after 16 unicast retransmissions to that neighbor. At this time, the neighbor is dropped. Neighbors can be displayed with the **show ip eigrp neighbors** command.
- **Topology table**—All learned routes reported by neighbors are kept in the topology table. The topology table also tracks the metrics and feasible distances associated with those routes. The topology table can be displayed with the **show ip eigrp topology** *as_number* command.
- **Route table/forwarding table**—Only the routes with the lowest composite metric are entered into the final route or forwarding table. This is the route that the router will forward to.

The process that DUAL uses to perform a loop-free topology is a detailed process. EIGRP has what is called a *feasible successor* and a *successor* to every route in its route table. The *successor* is the primary path for the route, or the path that the router will forward packets to. The *feasible successor* becomes the next-hop address only if the primary route to the destination becomes unreachable. The *feasible successor* is always downstream and, thereby, must have a distance or *feasible distance* that is less than that of the current preferred route. This prevents routing loops because the downstream router must always have a feasible cost lower than that of the current cost of the route to be considered as a feasible successor.

The DUAL process is in control of determining feasible distances, feasible successors, and the successor of the routes in the EIGRP topology table. By having a backup path already defined in the topology table, the router can quickly converge to the new path in case the primary path fails.

Protocol-Dependent Modules

EIGRP is one of the few routing protocols that can work with multiple routed protocols. Cisco implements what it calls *protocol-dependent modules* in the code that handle protocol-specific tasks. For example, IPX EIGRP needs to send and receive SAP updates. IP and IPX form neighbors using different message formats.

EIGRP operates the same way for all routed protocols—that is, it uses DUAL to find the shortest path to forward data toward. Another task of protocol-dependent modules is to pass data into the DUAL process so that a proper topology table, and eventually a route table, can be formed.

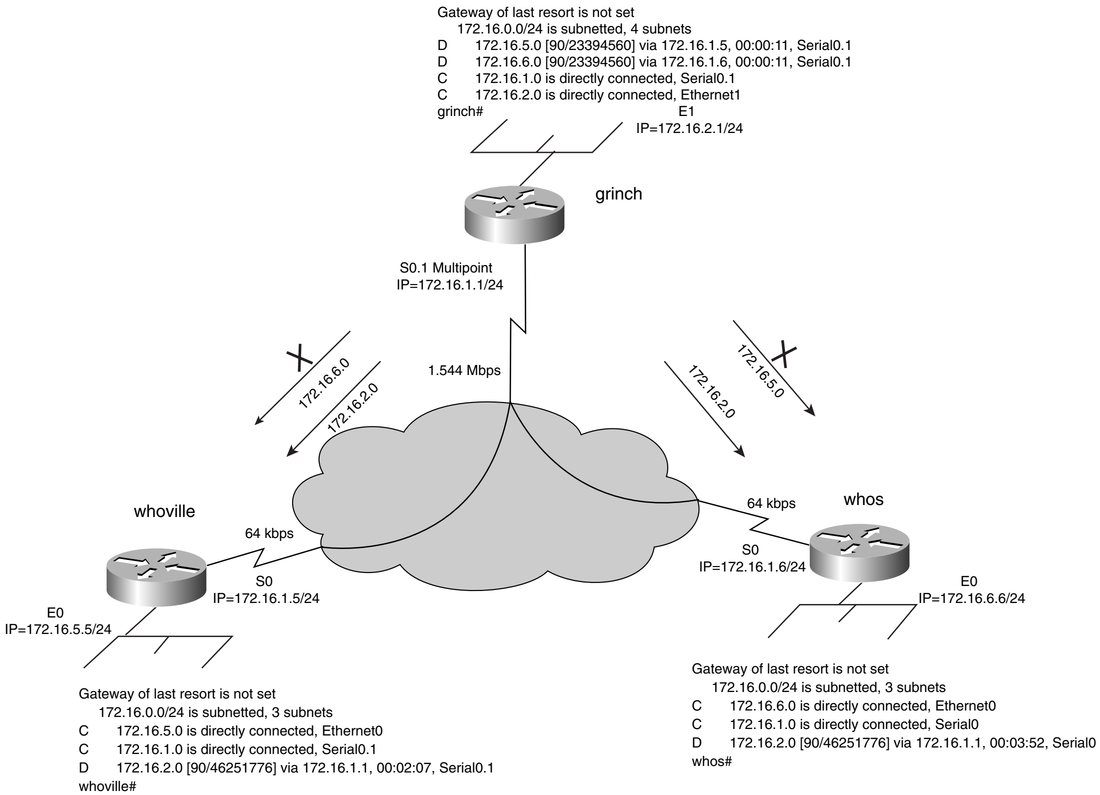
Like IGRP, EIGRP deploys the concepts of split horizon and poison reverse to prevent routing loops.

Split Horizon

Recall from earlier that *split horizon* is a routing technique in which information about routes is prevented from exiting the router interface or subinterface through which that information was received. Split horizon is most prevalent in multipoint networks. Here, routing updates flow into one subinterface but also must be sent out that very same subinterface to reach the other routers on the multipoint network. Split horizon is enabled by default and prevents specific route updates for EIGRP, IGRP, and RIP from being propagated properly in a multipoint configuration. Disable this with the **no ip split-horizon eigrp** *autonomous system* command. This command has similar forms for IPX and AppleTalk.

In Figure 11-4, the grinch router receives updates from the whos and whoville routers, but because of split horizon, the grinch does not advertise 172.16.5.0 and 172.16.6.0 out its serial 0.1 multipoint interface. Because the grinch didn't learn about the 172.16.2.0 network from its 0.1 interface, it advertises that network to the whos and whoville routers.

Figure 11-4 *EIGRP Split Horizons Route Suppression*



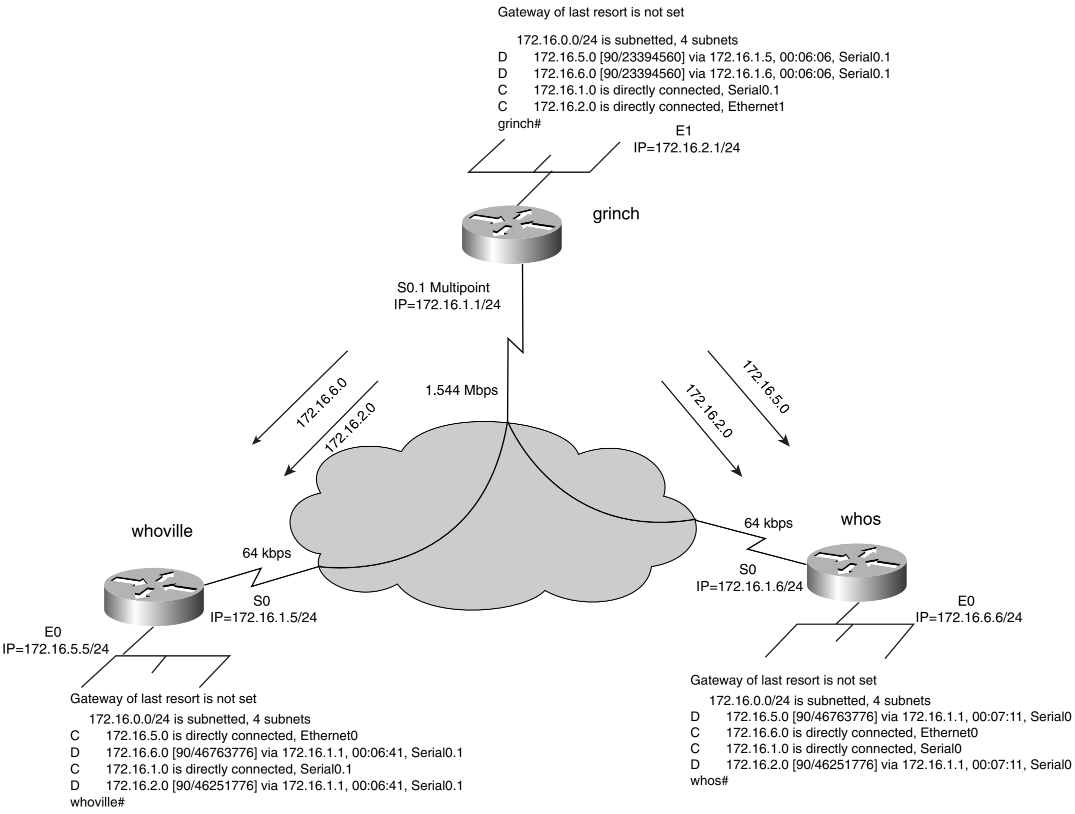
To make the whos and whoville happy again, we need to disable split horizon on the grinch by using the **no ip split-horizon eigrp** command, as demonstrated in Example 11-6.

Example 11-6 *Disabling Split Horizon on the grinch Router*

```
grinch(config)#int s0.1
grinch(config-subif)#no ip split-horizon eigrp 2001
```

Figure 11-5 illustrates how the routing tables will look after disabling split horizon on the grinch router. Notice that all routes are being propagated.

Figure 11-5 *Fully Functional EIGRP Network*



Configuring EIGRP

Configuring basic EIGRP is, for the most part, identical to configuring IGRP. Configuring EIGRP calls for the definition of an *autonomous system (AS)*. By definition, an AS is a set of routers under a single administrative technical authority. Like IGRP, EIGRP uses the concept of ASs to separate routing processes. Having a registered AS when configuring EIGRP is not required.

This following three-step process can be used to configure EIGRP. The third step is optional to specific environments.

Step 1 Enable EIGRP and define an AS on the router. This is accomplished with the **router eigrp** *autonomous_system_id* global command.

Step 2 Add the networks that you want to run EIGRP on. This is accomplished with the **network a.b.c.d** from the config-router# mode. When you enter the network statements, it is necessary to enter only the major class boundary. In Cisco IOS Software Release 12.0 and later, the **network** command adds an additional wildcard mask, much like OSPF. This is an inverse bit mask—for example, to enable EIGRP on network 172.16.1.0 only, the syntax would be **network 172.16.1.0 0.0.0.255**; however, note that *EIGRP is smart enough to convert a subnet mask to a wildcard mask if you make a mistake*. Now that's user-friendly!

Step 3 (Optional) Fine-tune EIGRP metrics with **bandwidth** statements, or configure IGRP summarization and options. By taking the time to configure bandwidth, EIGRP will have a more accurate picture of the network and also will aid in preventing EIGRP from saturating the link with broadcasts. The bandwidth always should be set on Frame Relay networks. The bandwidth can be changed with the **bandwidth kilobits** interface command. Later sections in the chapter cover bandwidth and summarizing EIGRP in greater detail.

Example 11-7 illustrates the EIGRP configuration from Figure 11-5 on the grinch router.

Example 11-7 *EIGRP Configuration*

```
! hostname grinch
!
interface Ethernet1
 ip address 172.16.2.1 255.255.255.0
 media-type 10BaseT
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 multipoint
 ip address 172.16.1.1 255.255.255.0
 no ip split-horizon eigrp 2001    ←Split Horizons disabled
 bandwidth 112                ←Bandwidth set to the sum of the remote PVCs
 frame-relay map ip 172.16.1.5 110 broadcast
 frame-relay map ip 172.16.1.6 130 broadcast
!
router eigrp 2001            ←EIGRP routing process
 network 172.16.0.0            ←Networks running EIGRP
!
```

Before further discussing these and other EIGRP options in greater detail, lets take a closer look at the **show** commands for EIGRP.

The "Big show" and "Big D" for EIGRP

Cisco offers some useful tools for determining how EIGRP is working. Perhaps one of the best and most overlooked commands is **show ip eigrp neighbors**. EIGRP neighbors remind me of an old Robert Frost poem that said, "Good fences make good neighbors." Well, in EIGRP, "Good networks make good neighbors." The neighbor state is absolutely critical to EIGRP operations. Besides providing the capability to assess neighbor states, Cisco offers tools to look at the EIGRP topology table, as well as providing detailed logging of EIGRP events.

The following is a list of what we find to be the most useful **show**, logging, and **debug** commands for EIGRP:

```
show ip eigrp neighbors [as_number | interface_name]
show ip eigrp topology [as_number | active | pending | summary] [as_number subnet
subnet_mask]
show ip protocols [summary]
show ip route
debug eigrp packets
eigrp log-neighbor-changes
```

show ip eigrp neighbors Command

This can be one of the most useful commands when verifying the operational status of EIGRP. The **show ip eigrp neighbors** command shows the status of all EIGRP neighbors. The neighbor should be "up" for as long as EIGRP has been running on the link. EIGRP forms a neighbor relationship with all routers on the same subnet and in the same AS. EIGRP does not form a neighbor relationship with mismatched k values; however, a neighbor can be formed with mismatched hellos and dead timers. A neighbor with a short uptime is a clear indication of a problem. Another important field is the queue count. This field indicates the number of packets waiting to be transmitted to that neighbor. This value should be 0 or a number under 20. Consistent Q values in the range of 60 or greater are considered high. A high SRTT number can mean that the packet is experiencing some type of delay on the link. Example 11-8 provides some sample output from the **show ip eigrp neighbor** command, which provides the basis for an explanation of the other fields, which follows.

Example 11-8 show ip eigrp neighbor *Command Performed on the grinch Router*

```
grinch#show ip eigrp neighbors
IP-EIGRP neighbors for process 2001
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
1   172.16.1.5              Se0.1        136 05:48:23   36   1302  0  15
0   172.16.1.6              Se0.1        131 05:48:24   40   1302  0  17
grinch#
```

- **Handle (H)**—A Cisco IOS internal number used to identify a neighbor. Do not confuse this with hop count.
- **Neighbor Address**—The adjacent neighbor's IP address. A neighbor should be formed between every router on that subnet running EIGRP in a common AS.
- **Interface**—The interface that is reporting the neighbor.
- **HoldTime**—The amount of time, which counts down, that EIGRP waits for a hello before tearing down the neighbor.
- **Uptime**—Statement of how long the neighbor has been up. This number should be up for as long as the link has been up.
- **Smooth Round Trip Timer (SRRT)**—The number of milliseconds that it takes for an EIGRP packet to be sent to this neighbor and for the local router to receive an acknowledgment—hence, a round-trip timer. If this number equals 0, a packet has never made a successful round trip.
- **Retransmission TimeOut (RTO)**—The amount of time, in milliseconds, that the EIGRP waits before retransmitting a packet from the retransmission queue to a neighbor.
- **Queue count (Q)**—The number of packets waiting in the queue to be sent out to this neighbor. This value should be 0 or a very low number. A high queue count indicates that data is having trouble getting through.
- **Sequence Number (Seq-Num)**—Sequence number of the last update, query, or reply that was received from this neighbor. If this number equals 0, it indicates that no reliable packets have ever been received from the neighbor, another clear indication of a problem.

NOTE Just because a network appears in the route table does not necessarily mean that "routing" is working properly. In some instances, such as timer mismatches, networks can "phase" in and out of the route table. It is important to look at other things, such as neighbors and databases, to get a clearer view of whether "routing" is actually working properly.

show ip eigrp topology Command

This command lists the EIGRP topology table discussed earlier. The table lists all routes that EIGRP is aware of and shows whether EIGRP is actively processing information on that route. Under most normal conditions, the routes should all be in a passive state and no EIGRP process are running for that route. If the routes are active, this could indicate the dreaded *stuck in active*, or SIA, state, which is discussed in more detail in an upcoming section. The **show ip eigrp topology** command also can be extended to show information about an individual route or subnet. This information includes all relevant information about the route, including all its metrics and successors, as well as how the route was learned. Example 11-9 illustrates the use of **show ip eigrp topology**, followed by the extended version of the command.

Example 11-9 *EIGRP Topology Table of the grinch Router*

```
grinch#show ip eigrp topology
IP-EIGRP Topology Table for process 2001

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 172.16.5.0/24, 1 successors, FD is 23394560
         via 172.16.1.5 (23394560/281600), Serial0.1
P 172.16.6.0/24, 1 successors, FD is 23394560
         via 172.16.1.6 (23394560/281600), Serial0.1
P 172.16.1.0/24, 1 successors, FD is 23368960
         via Connected, Serial0.1
P 172.16.2.0/24, 1 successors, FD is 281600
         via Connected, Ethernet1
grinch#

grinch#show ip eigrp topology 2001 172.16.5.0 255.255.255.0
IP-EIGRP topology entry for 172.16.5.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 23394560
  Routing Descriptor Blocks:
  172.16.1.5 (Serial0.1), from 172.16.1.5, Send flag is 0x0
      Composite metric is (23394560/281600), Route is Internal
      Vector metric:
        Minimum bandwidth is 112 Kbit
        Total delay is 21000 microseconds
        Reliability is 254/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
grinch#
```

The fields to note in this output are as follows:

- **P**—Passive; no EIGRP computation is being performed. This is the ideal state.
- **A**—Active; EIGRP computations are “actively” being performed for this destination. Routes constantly appearing in an active state indicate a neighbor or query problem. Both are symptoms of the SIA problem.
- **U**—Update; an update packet was sent to this destination.
- **Q**—Query; a query packet was sent to this destination.
- **R**—Reply; a reply packet was sent to this destination.

- **Route information**—IP address of the route or network, its subnet mask, and the successor, or next hop to that network, or the feasible successor.
- **FD**—Feasible distance to the destination network.
- Send Flag—The type of packets that need to be sent for the entry.
  - 0x1 The router has received a query for this network and needs to send a unicast reply.
  - 0x2 The route is active, and a multicast query should be sent.
  - 0x3 The route has changed, and a multicast update should be sent.

show ip protocols Command

This command displays all routing protocols, detailed timer and metric information, as well as routing update information. Example 11-10 lists the output of the **show ip protocols** command.

Example 11-10 show ip protocols *Command Output*

```
grinch#show ip protocols
Routing Protocol is "eigrp 2001"        ←AS system ID
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0 ←'K' values
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 2001
  Automatic network summarization is in effect    ←Auto-summary in effect
  Routing for Networks:
    172.16.0.0                          ←Networks running EIGRP
  Routing Information Sources:
    Gateway         Distance      Last Update
   172.16.1.5          90     00:08:48      ←Routes reported, and administrative
    172.16.1.6          90      00:08:52         distance of the route.
  Distance: internal 90 external 170           ←Default admin distance

grinch#
```

show ip route Command

This command lists the router's current route or forwarding table. The output lists what routing protocol the route is from—in this case, D for EIGRP internal routes and D EX for routes redistributed into EIGRP. The number behind the route is the administrative distance of the route, followed by the composite metric of the route. The via field explains where the route is from, how long ago an update was received, and by what interface it was received. Example 11-11 lists the output of this command.

Example 11-11 show ip route *Command Output*

```
grinch#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 4 subnets
D       172.16.5.0 [90/23394560] via 172.16.1.5, 00:17:51, Serial0.1
D       172.16.6.0 [90/23394560] via 172.16.1.6, 00:29:06, Serial0.1
C       172.16.1.0 is directly connected, Serial0.1
C       172.16.2.0 is directly connected, Ethernet1
grinch#
```

debug eigrp packets Command

The "Big D" command for EIGRP, is just that: big. As discussed earlier, debugs always should be used in conjunction with logging. However, some EIGRP debugs can be so big that additional debugs are needed to control the output of the original **debug** command. One such case is the **debug eigrp packets** command.

Use the **debug eigrp packets** command to verify that EIGRP hellos are being exchanged and that adjacencies are being established. Each EIGRP packet sent and received is listed in this output. The output of this command can be controlled with further debugs, such as **debug ip eigrp [neighbor as_number IP_address_of_neighbor]**. Use the **debug ip eigrp** command. Use this command with caution and only to look further into a problem. Do not start troubleshooting EIGRP with this command. Example 11-12 lists the output of the **debug eigrp packets** command.

Example 11-12 debug eigrp *packets Command Output*

```
grinch#debug eigrp packets
06:22:29: EIGRP: Received HELLO on Serial0.1 nbr 172.16.1.5
06:22:29:   AS 2001, Flags 0x0, Seq 0/0 idbQ 0/0
06:22:29: EIGRP: Enqueueing UPDATE on Serial0.1 nbr 172.16.1.5 iidbQ un/rely 0/1
 peerQ un/rely 0/0 serno 2-10
06:22:29: EIGRP:  Requeued unicast on Serial0.1
06:22:29: EIGRP: Sending UPDATE on Serial0.1 nbr 172.16.1.5
06:22:29:   AS 2001, Flags 0x1, Seq 7/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely
 0/1 serno 2-10
```

eigrp log-neighbor-changes Command

EIGRP also offers a unique logging command that can be useful when trying to isolate problems on your network. Use the router command **eigrp log-neighbor-changes** to verify any loss of EIGRP neighbors. Example 11-13 lists the log after an EIGRP hold time has expired.

Example 11-13 *EIGRP Log After a Neighbor Change*

```
grinch(config-router)#eigrp log-neighbor-changes
06:42:12: %DUAL-5-NBRCHANGE: IP-EIGRP 2001: Neighbor 172.16.1.6 (Serial0.1) is d
own: holding time expired
```

Tuning EIGRP Updates

Like IGRP, EIGRP has several parameters for tuning timers, controlling broadcasts, load sharing, and controlling routes. The following is a list of parameters adjustable for EIGRP:

- Router(config-if)**ip hello-interval eigrp** *as_number interval_in_seconds*—Use this interface command to change the hello timer for EIGRP. The default value of this command is interface-dependant. By default, hello packets are sent every 5 seconds. The exception to this is low-speed, *nonbroadcast multiaccess media (NBMA)*, where it is 60 seconds. Low-speed is defined as rates of T1 (1.544 Mbps) or slower. All neighbors residing on a network should have equal hello timers.
- Router(config-if)**ip hold-time eigrp** *as_number holdown_timer_in_seconds*—Use this command to change the EIGRP hold timer for routes received by this interface. The timer has a default vault of 180 seconds for low-speed NBMA networks and 15 seconds for all other networks. All neighbors residing on a network should have an equal hold timer.

EIGRP Redistribution and Route Control

To filter routing updates in EIGRP, use a distribute list. A distribute list calls a standard or extended access list and filters routing updates accordingly. When redistributing one protocol into another, use the **redistribute** command along with a default metric. A route map should be used in place of a distribute list when controlling specific routes during the redistribution process. Redistribution happens automatically between IGRP and EIGRP when they are in the same autonomous systems.

- Router(config-router)**distribute-list** [**1-199**] [**in | out**] [*interface*]—Use this command to call a standard or extended access list to filter inbound or outbound routing updates. The **in** and **out** options always are applied from the view of the interface—in other words, to prevent a routing update from being advertised out an interface, use the **out** option. To prohibit route updates from entering an interface, use the **in** option.

- Router(config-router)**redistribute** [**connected** | **static** | **bgp** | **rip** | **igrp** | **ospf** | **isis**] *{metric} {route-map}*—Use this command to redistribute other routing protocols into EIGRP. A route map may be added for additional route control. An optional metric also can be supplied for routes originating from the routing protocol being redistributed that are different from the default metric. Whenever redistributing routes, remember that IP needs a route to and from a destination. Many times, mutual redistribution might be required to give IP a path to *and* from a destination.

  Router(config-router)**default-metric** [*bandwidth_kbps 1-4214748364*] [*delay_ ms 1-4214748364*] [**reliability** *1-255*] [**load** *1-244*] [**mtu** *1-4214748364*]—Use this command to set the default metric of all routes redistributed into EIGRP. You must supply a default metric whenever redistributing. A common metric to use is **default-metric 1544 100 254 1 1500**. This metric tells the router to derive the composite metric from the values of bandwidth of 1544 and delay of 100; with a link that is 254 reliable, where 255 is 100 percent reliable; with a load of 1, or no load; and, finally, an MTU of 1500. Perhaps more important than the actual value of the default metric is the practice of using the same metric throughout the EIGRP domain so that all redistributed routes have the same weight.

NOTE Whenever you are redistributing one routing protocol into another, you must use a default metric or supply a metric on the redistribution command.

The following subsets of commands are used to influence routing decisions made by EIGRP. Individual metrics can be modified in addition to the administrative distance of the EIGRP. Whenever you are influencing a specific link's metric, use the **delay** command over the **bandwidth** command. Both may be used; however, recall that OSPF also is affected by **bandwidth**, whereas **delay** affects only IGRP and EIGRP.

- Router(config-router)**metric weights 0** *k1 k2 k3 k4 k5*—This command allows you to set the weight of the EIGRP metric in terms of bandwidth, load, delay, and reliability. Change these values with extreme caution; EIGRP will not form neighbors with mismatched K values.
- Router(config-router)**distance** [*1-255*] *adjacent_neighbors_ip_address wildcard_ mask [access_list_0-99]*—Use this command to change the administrative distance of routes received from a neighbor. If the IP address and wildcard mask are omitted, all routes for that protocol will be set to the distance value. For a specific example and more practice with the **distance** command, see Chapter 10, "Distance Vector Protocols: Interior Gateway Routing Protocol (IGRP)."

- **Router(config-if)delay** [*1-4214748364*]—Specifies the delay of an interface in tens of microseconds. This command is used only by routing protocols and does not affect traffic on the link.
- Router(config-if)**bandwidth** [*bandwidth_kbps 1-4214748364*]—Specifies the bandwidth of an interface in kilobits per second. This command is used only by routing protocols and does not affect traffic on the link.
- Router(config-router)**passive-interface** *interface_name*—Prevents the sending of EIGRP hellos on the link. This command operates differently on EIGRP than on IGRP. Because hellos are suppressed, no neighbors are formed; therefore, no routing updates are sent or received.
- Router(config-router)**offset-list** [**access_list_0-99** {**in** | **out**} **offset** [*metric_offset_ 1-214748364*] [interface]—Used to increase the value of the routing metrics. The metric offset cannot exceed 214748364. The offset list is applied in the same way as it is in RIP, using the EIGRP metric. For an example of the application of the offset list, see Chapter 9, "Distance Vector Protocols: Routing Information Protocol Versions 1 and 2 (RIP-1 and RIP-2).

Practical Example: Applying EIGRP Redistribution

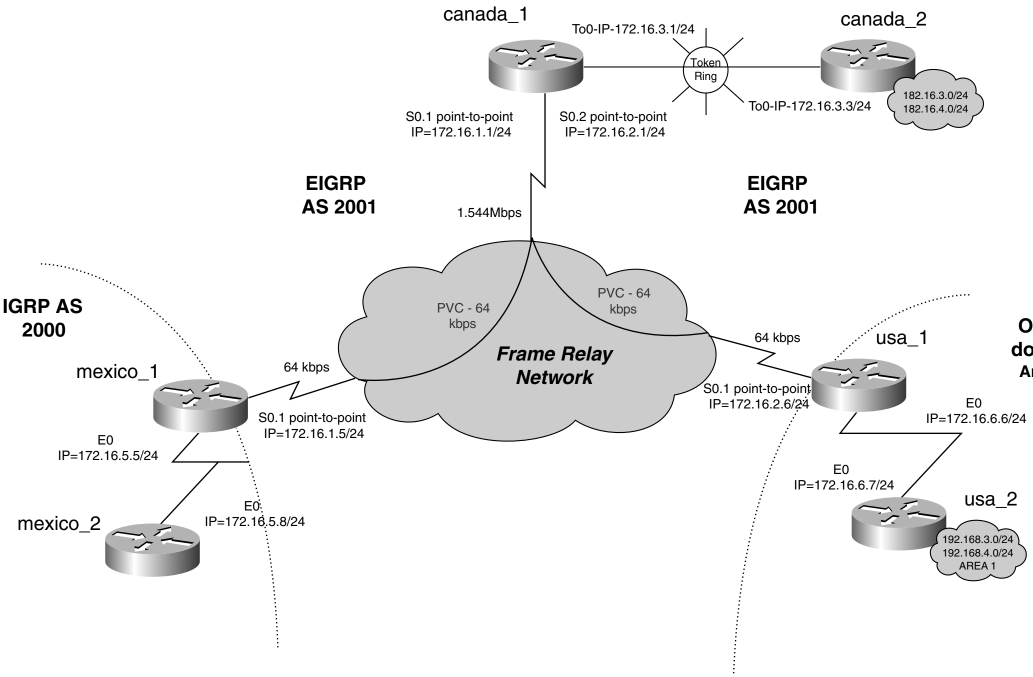
Let's apply some of these concepts to a practical model in route redistribution and control. The model in Figure 11-6 shows three routing domains. The canada routers and the Frame Relay network reside in the EIGRP domain. Across the Frame Relay network reside two other routing domains; the mexico routers are in an IGRP domain, while the usa routers reside in an OSPF domain.

You must verify two things within the routing domains to allow IP end-to-end connectivity:

- Notice that the IGRP domain is on a 24-bit boundary. This means that when the IGRP domain receives a route, it must exist on a major bit boundary or a 24-bit boundary for the interface to accept that route.
- Mutual redistribution must occur between EIGRP and IGRP, and EIGRP and OSPF.

Beginning with the configuration for the canada_1 router, you can follow the three-step process for configuring EIGRP as listed earlier in this chapter. First, all EIGRP routers are in the autonomous system 2001; therefore, you will use this as the Autonomous System ID. Second, the networks that you are running EIGRP on reside in the major network of 172.16.0.0, which you will use in the **network** command. The third step is optional; in this case, however, you are configuring EIGRP over Frame Relay, so it's a good idea to add the **bandwidth** commands under the serial subinterfaces. In this model, you will set the bandwidth equal to the port speed of the remote routers Frame Relay interface. Example 11-14 lists the configuration of the canada_1 router.

Figure 11-6 *EIGRP Network For EIGRP Redistribution and Route Control Examples*



Example 11-14 *Configuration of the canada_1 Router*

```
hostname canada_1
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 point-to-point
 ip address 172.16.1.1 255.255.255.0
 bandwidth 64                       ←EIGRP bandwidth set
 frame-relay interface-dlci 110
!
interface Serial0.2 point-to-point
 ip address 172.16.2.1 255.255.255.0
 bandwidth 64                       ←EIGRP bandwidth set
 frame-relay interface-dlci 130
!
interface TokenRing0
 ip address 172.16.3.1 255.255.255.0
```

continues

Example 11-14 *Configuration of the canada_1 Router (Continued)*

```
 ring-speed 16
!
router eigrp 2001                  ←EIGRP routing enabled
 network 172.16.0.0                  ←Networks running EIGRP
```

You can follow the same process to configure EIGRP for the mexico_1 and usa_1 routers, with a couple of minor differences. In both instances, you do not want to risk having any EIGRP neighbors automatically created on the Ethernet segments of these routers. To accomplish this, add the **passive-interface ethernet 0** command under EIGRP for the mexico_1 and usa_1 routers. Example 11-15 lists the configuration thus far for the mexico_1 and usa_1 routers. For more information on the IGRP and OSPF configuration portions of the configuration, see Chapter 10 and Chapter 12, "Link-State Protocols: Open Shortest Path First (OSPF)."

Example 11-15 *EIGRP Configuration of mexico_1 and usa_1 Routers*

```
hostname mexico_1
!
interface Ethernet0
 ip address 172.16.5.5 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 point-to-point
 ip address 172.16.1.5 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 111
!
<<<Text omitted>>>
!
router eigrp 2001
 passive-interface Ethernet0
 network 172.16.0.0
!
router igrp 2000
 passive-interface Serial0.1
 network 172.16.0.0
<<<Text omitted>>>
```

```
hostname usa_1
!
interface Ethernet0
 ip address 172.16.6.6 255.255.255.0
!
```

Example 11-15 *EIGRP Configuration of mexico_1 and usa_1 Routers (Continued)*

```
interface Serial0
 no ip address
 encapsulation frame-relay
!
interface Serial0.1 point-to-point
 ip address 172.16.2.6 255.255.255.0
 frame-relay interface-dlci 131
!
<<<Text omitted>>>
!
router eigrp 2001
 passive-interface Ethernet0
 network 172.16.0.0
!
router ospf 69
 network 172.16.6.6 0.0.0.0 area 0
<<<Text omitted>>>
```

At this point, you should have full IP connectivity within the EIGRP routing domain. To verify this, perform a **show ip route** combined with the **show ip eigrp neighbors** command on the canada_1 router, as demonstrated in Example 11-16.

Example 11-16 *Verifying EIGRP Routing*

```
canada_1#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 5 subnets
D       172.16.5.0 [90/40537600] via 172.16.1.5, 00:45:58, Serial0.1
D       172.16.6.0 [90/40537600] via 172.16.2.6, 00:45:58, Serial0.2
C       172.16.1.0 is directly connected, Serial0.1
C       172.16.2.0 is directly connected, Serial0.2
C       172.16.3.0 is directly connected, TokenRing0
D    182.16.0.0/16 [90/304128] via 172.16.3.3, 00:43:27, TokenRing0
canada_1#
canada_1#show ip eigrp neighbors
IP-EIGRP neighbors for process 2001
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
2   172.16.3.3              To0           11 00:43:36  685  4110  0  3
1   172.16.2.6              Se0.2         14 1d06h      48  2280  0  28
0   172.16.1.5              Se0.1         12 1d06h      29  2280  0  23
canada_1#
```

The important elements of the output that you are looking for are that route 172.16.5.0/24 is reported through 172.16.1.5, route 172.16.6.0/24 is reported through 172.16.2.6, and 182.16.3.0/24 and 182.16.4.0/24 are reported through 172.16.3.3. Because of EIGRP auto-summarization, 182.16.3.0/24 and 182.16.4.0 will be summarized at its natural 16 bit-boundary when these routes are advertised out the canada_2 Token Ring interface. The **show ip eigrp neighbors** command verifies that EIGRP adjacencies have been formed between canada_1 and the other two routers.

To allow EIGRP connectivity to the OSPF routing domain, you must mutually redistribute between EIGRP and OSPF on the usa_1 router. There is only one redistribution point for EIGRP and OSPF, so you do not have to take into account "route feedback" or redistribution loops. Example 11-17 shows the configuration of the usa_1 router.

Example 11-17 *Redistribution Configuration Portion of usa_1*

```
 !
router eigrp 2001
 redistribute ospf 69
 passive-interface Ethernet0
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
!
router ospf 69
 redistribute eigrp 2001 subnets
 network 172.16.6.6 0.0.0.0 area 0
 default-metric 100
!
```

The OSPF routes 192.168.3.0/24 and 192.168.4.0/24 now appear as external EIGRP routes on the canada_1 router. Likewise, all EIGRP routes appear as OSPF external Type 2 routes on the usa_2 router.

Mutual redistribution also must be performed between the EIGRP and IGRP routing domains on the mexico_1 router. If the IGRP routing domain was in the same autonomous system as EIGRP, redistribution would not be necessary because it would occur automatically. Example 11-18 shows the configuration of the mexico_1 router.

Example 11-18 *Redistribution Configuration portion of mexico_1*

```
!
router eigrp 2001
 redistribute igrp 2000
 passive-interface Ethernet0
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
!
router igrp 2000
 redistribute eigrp 2001
```

Example 11-18 *Redistribution Configuration portion of mexico_1 (Continued)*

```
 passive-interface Serial0.1
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
!
```

The route table for the mexico_2 router now shows all the appropriate routes for every network in the model. Example 11-19 shows the route table of mexico_2.

Example 11-19 *Route Table of the mexico_2 Router After Redistribution*

```
mexico_2#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 5 subnets
C       172.16.5.0 is directly connected, Ethernet0/0
I       172.16.6.0 [100/10676] via 172.16.5.5, 00:00:16, Ethernet0/0
I       172.16.1.0 [100/8576] via 172.16.5.5, 00:00:16, Ethernet0/0
I       172.16.2.0 [100/10576] via 172.16.5.5, 00:00:16, Ethernet0/0
I       172.16.3.0 [100/8639] via 172.16.5.5, 00:00:16, Ethernet0/0
I    192.168.4.0/24 [100/10676] via 172.16.5.5, 00:00:16, Ethernet0/0
I    182.16.0.0/16 [100/9139] via 172.16.5.5, 00:00:16, Ethernet0/0
I    192.168.3.0/24 [100/10676] via 172.16.5.5, 00:00:16, Ethernet0/0
mexico_2#
```

The redistribution in this model was relatively straightforward because all the networks in the model either are on a 24-bit boundary or are automatically summarized on a 24-bit boundary. EIGRP automatically summarizes at a major bit boundary when advertising or redistributing. During redistribution into IGRP, EIGRP automatically summarized the network 192.168.4.0/24 because it is on a 24-bit boundary along with 192.168.3.0/24. The network 182.16.0.0 was summarized when it was advertised out the s0.1 and s0.2 interfaces.

It is important to note that if the IP address of the advertising interface is in the same major class boundary as the route being advertised, automatic summarization will not occur. For example, if the router were advertising 172.16.100.0/30 out an interface with an IP address of 172.16.10.1/24, EIGRP would not summarize the route at its natural bit boundary. If the same network, 172.16.100.0/24, was advertised out an interface with the IP address of 172.17.10.1/24, EIGRP would advertise only the summary route 172.16.0.0/16, as seen in the previous model.

As you will see in the upcoming section on EIGRP summarization, we will make some subtle changes to the IP address structure, which will force the use of manual summarization before redistribution will work correctly.

Practical Example: Applying EIGRP Route Control

Now that you have a working IP network, let's examine route control through the application of route maps and distribution lists using the network in Figure 11-6 as the model again. On the usa_1 router, you will apply a distribution list preventing the route 192.168.3.0/24 from being advertised by EIGRP to the entire EIGRP domain. To carry out this task, use the **distribute-list** router command; apply an access list denying 192.168.3.0/24, while allowing other routes to be advertised. Example 11-20 highlights the configuration of the usa_1 router, allowing EIGRP to advertise only the 192.168.3.0/24 route.

Example 11-20 *Application of Distribution List*

```
router eigrp 2001
 redistribute ospf 69
 passive-interface Ethernet0
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
 distribute-list 10 out Serial0.1    ←Apply access list 10 to interface s0.1
!
router ospf 69
 redistribute eigrp 2001 subnets
 network 172.16.6.6 0.0.0.0 area 0
 default-metric 100
!
ip classless
access-list 10 deny   192.168.3.0 0.0.0.255    ←deny route 192.168.3.0/24
access-list 10 permit any           ←allow all other routes to pass
!
```

Whenever you are controlling routing updates from one routing protocol to another, use a route map. In this model, a route map is used to prohibit the OSPF route of 172.16.6.0/24 from being redistributed from EIGRP into IGRP. The route map is called from the **redistribution** command in IGRP; the route map then calls and permits routes that match access list 11. Example 11-21 lists the configuration of the mexico_1 router using a route map to filter the route 172.16.6.0/24.

Example 11-21 *Calling a Route Map During Redistribution on mexico_1*

```
router eigrp 2001
 redistribute igrp 2000
 passive-interface Ethernet0
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
```

Example 11-21 *Calling a Route Map During Redistribution on mexico_1 (Continued)*

```
!
router igrp 2000
 redistribute eigrp 2001 route-map noospf    ←call route map named noospf
 passive-interface Serial0.1
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
!
ip classless
!
access-list 11 deny   172.16.6.0 0.0.0.255    ←deny 172.16.6.0/24
access-list 11 permit any
route-map noospf permit 10
 match ip address 11                ←allow routes that pass access list 11
!
```

The route table on the mexico_2 router now shows only one route from the OSPF domain 192.168.4.0. Compare the output in Example 11-22 with that of Example 11-19 to see the application of the route map and distribution list.

Example 11-22 *The Route Table of mexico_2 After Route Filtering*

```
mexico_2#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 4 subnets
C       172.16.5.0 is directly connected, Ethernet0/0
I       172.16.1.0 [100/8576] via 172.16.5.5, 00:00:51, Ethernet0/0
I       172.16.2.0 [100/10576] via 172.16.5.5, 00:00:51, Ethernet0/0
I       172.16.3.0 [100/8639] via 172.16.5.5, 00:00:51, Ethernet0/0
I    192.168.4.0/24 [100/10676] via 172.16.5.5, 00:00:51, Ethernet0/0
I    182.16.0.0/16 [100/9139] via 172.16.5.5, 00:00:51, Ethernet0/0
mexico_2#
```

NOTE A route map also may be used to set an EIGRP tag, using the syntax **set tag** xx under the **route-map** command. Setting tags can be useful for looking at how routes entered a route table. The tag can be viewed in EIGRP by the **show ip eigrp topology** command, and in OSPF by **show ip ospf database**. The OSPF tag also can be entered directly on the **redistribution** command.

EIGRP Summarization

Understanding EIGRP summarization and knowing how to effectively use it are absolutely vital to the design of large EIGRP networks. EIGRP scales very well, but when the number of routes starts to climb into the hundreds, extra care should be taken to control route propagation and the query range. As much as EIGRP is plug-and-play on small networks, it is not on large networks. The larger the network is, the more care should be taken to control how routes propagation.

Summarization provides two powerful enhancements to EIGRP. First, by lowering the number of routes in the route table, it lessens the number and size of the EIGRP advertisements. Second, and more importantly, it can limit the EIGRP query range.

Controlling the Query Range Through Summarization, Addressing Stuck in Active (SIA) Route Issues

Arguably one of the most common and complex problems facing large EIGRP networks is *stuck in active (SIA)* routes. A route becomes SIA when EIGRP is "actively" running computations for the route, and it doesn't stop. EIGRP will log multiple messages similar to the following:

```
%DUAL-3-SIA: Route 192.168.1.16 Stuck-in-Active
```

Most of the time, the route shows active because it is waiting for query to return from neighbor. There can be many reasons for this:

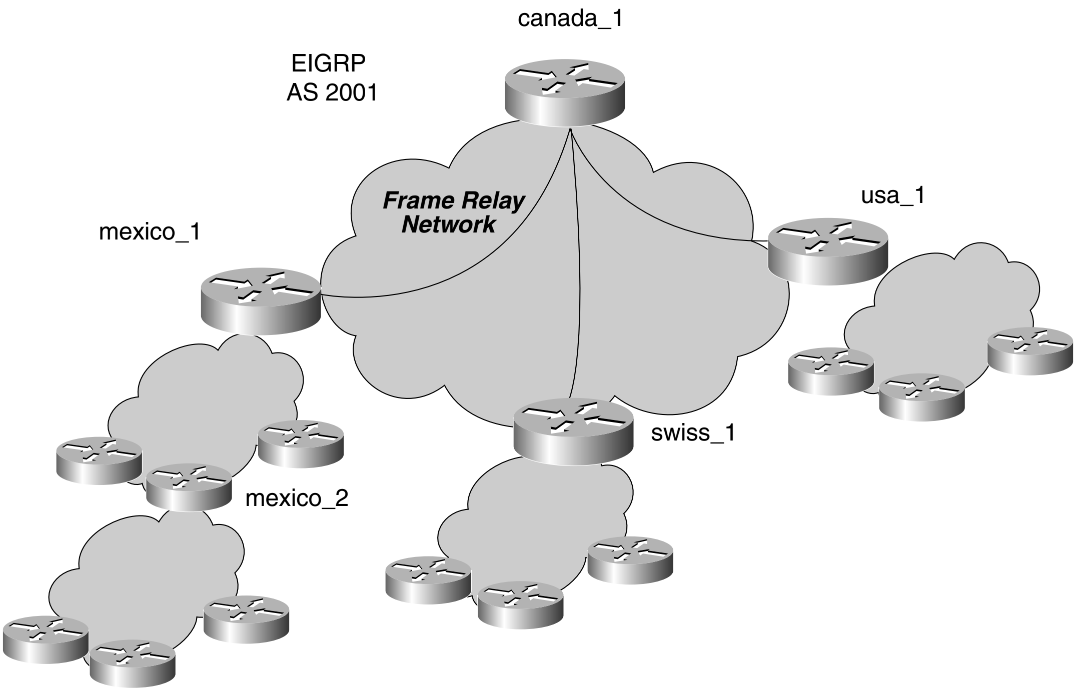
- **The router is overutilized**—Edge routes can get flooded with query request and often don't have the CPU power to keep up with the request. Queries receive irregular replies, if any, and the route stays active.
- **The router is having memory issues**—This can be compounded from a slow processor and the router having many items in its queues.
- **Overutilized circuit**—EIGRP hellos might not be getting through, causing the neighbor to drop.

Two types of configurations can manifest the SIA situation:

- Most EIGRP networks have autosummary disabled. This is primarily because the IP addressing scheme has discontinuous subnets, so the query range is not bounded.
- Large Frame Relay networks have many remote sites coming into the same router, therefore, there are many EIGRP neighbors.

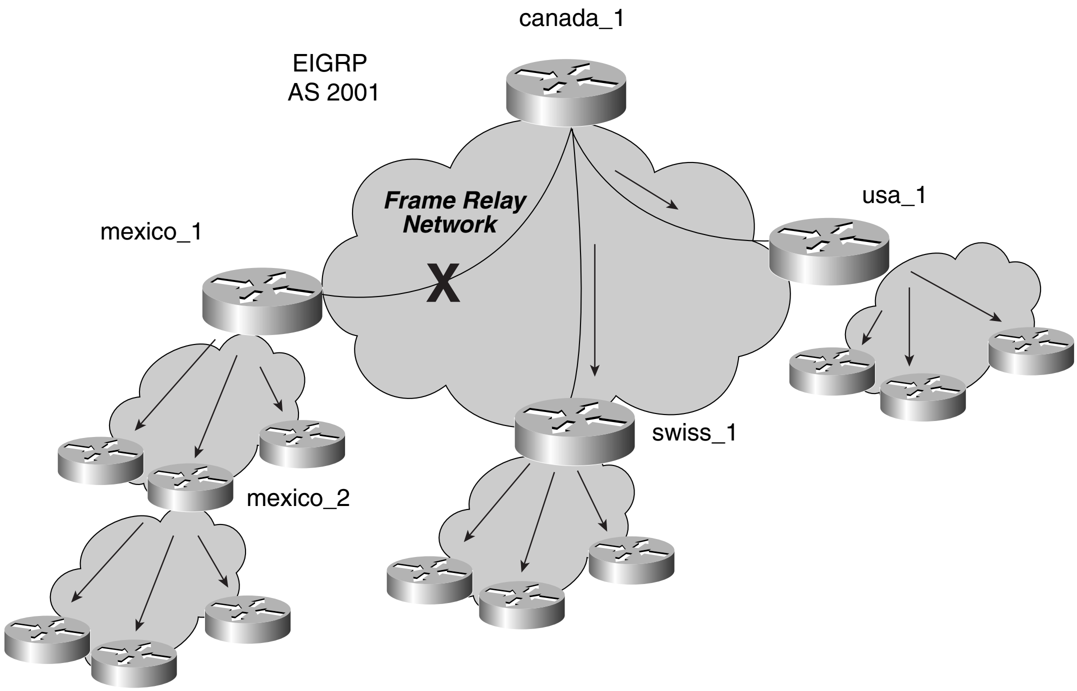
For example, in a Frame Relay network, if a single PVC goes inactive or a route starts flapping, it can cause a small *EIGRP query storm*. Figures 11-7 and 11-8 illustrate a common Frame Relay network and the query process.

Figure 11-7 *EIGRP Network*



If the PVC is lost from canada_1 to mexico_1, the canada_1 router sends an EIGRP query message to all of its neighbors regarding the routes that it lost from the mexico_1 router. It is looking for a new feasible successor to the routes. In this case, the message goes to the swiss_1 and usa_1 routers. The mexico_1 router also sends a query message to all of its neighbors looking for a new feasible successor for the routes that it lost from canada_1. All routers in the EIGRP domain continue to issue queries to neighbors. The routes stay in an "active" state until EIGRP receives "replys" to the queries that it sent. If you scale this network to a router with an HSSI or T3 interface, it would be possible to have hundreds of PVCs on a single interface, and the loss of just a single PVC could generate hundreds or thousands of queries. Fortunately, summarization bounds the query process and is one of the most effective ways to control EIGRP query storms. A large EIGRP network without summarization is an SIA problem looking for an owner.

Figure 11-8 *EIGRP Query Storm*



EIGRP Autosummarization

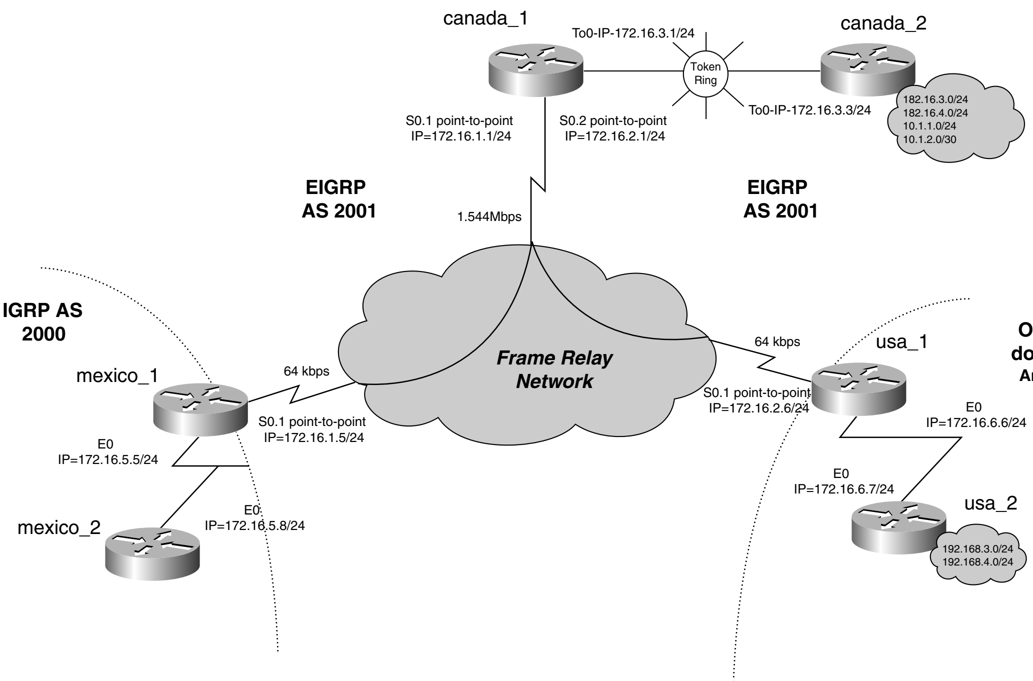
By default, EIGRP performs autosummarization in two situations:

- Autosummarization will occur at the major class boundary during redistribution from EIGRP into a classful routing protocol, such as IGRP or RIP. This type of summarization cannot be disabled.
- Autosummarization will occur at the major class boundary when the route is advertised out an interface that is on a different major class boundary. This summarization can be disabled with the command **no auto-summary** from the router(config-router) prompt.

EIGRP will not automatically summarize EIGRP external routes.

EIGRP routes that are summarized have an administrative distance of 90. In Figure 11-9, the internetwork has been modified from the previous example, adding two additional networks to the canada_2 router.

Figure 11-9 *EIGRP Autosummarization*



With autosummarization enabled, EIGRP on the canada_2 router advertises two summary routes to canada_1. The routes 182.16.3.0/24 and 182.16.4.0/24 are advertised as 182.16.0.0/16. The routes 10.1.1.0/24 and 10.1.2.0/30 are advertised at their natural class boundary with a route of 10.0.0.0/8. It is important to note that EIGRP summarizes the route only when advertising out an interface that is in a different class. For example, if the network between canada_1 and canada_2 is 10.1.3.0/24, the 10's network would not be summarized; only the 182.16.x.x networks would. Example 11-23 lists the route table of canada_1, highlighting the summarized routes.

Example 11-23 *Route Table of canada_1 with Summarized Routes*

```
canada_1#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR
```

continues

Example 11-23 *Route Table of canada_1 with Summarized Routes (Continued)*

```
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 5 subnets
D       172.16.5.0 [90/2195456] via 172.16.1.5, 01:06:41, Serial0.1
D       172.16.6.0 [90/2195456] via 172.16.2.6, 01:08:01, Serial0.2
C       172.16.1.0 is directly connected, Serial0.1
C       172.16.2.0 is directly connected, Serial0.2
C       172.16.3.0 is directly connected, TokenRing0
     192.168.4.0/32 is subnetted, 1 subnets
D EX    192.168.4.1 [170/2195456] via 172.16.2.6, 01:07:46, Serial0.2
D    10.0.0.0/8 [90/304128] via 172.16.3.3, 00:51:27, TokenRing0
D    182.16.0.0/16 [90/304128] via 172.16.3.3, 00:51:27, TokenRing0
     192.168.3.0/32 is subnetted, 1 subnets
D EX    192.168.3.1 [170/2195456] via 172.16.2.6, 01:07:46, Serial0.2
canada_1#
```

As helpful as EIGRP summarization might appear on the surface, it has serious drawbacks on most modern networks. It essentially makes a classless routing protocol enforce the discontinuous subnets rule at the major bit boundaries. When EIGRP forms a summary route to advertise, it also forms a route to null for all the networks in that summary. For example, the canada_2 router will form three routes to null for each major class network that the router has an interface in. The route to null will discard any packets that this router has an explicit route to. Example 11-24 demonstrates autosummary null routes displayed by the **show ip route** command.

Example 11-24 *EIGRP Autosummary Null Routes*

```
canada_2#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D       172.16.5.0/24 [90/2211584] via 172.16.3.1, 00:12:02, TokenRing0
D       172.16.6.0/24 [90/2211584] via 172.16.3.1, 00:12:02, TokenRing0
D       172.16.0.0/16 is a summary, 00:12:06, Null0
D       172.16.1.0/24 [90/2185984] via 172.16.3.1, 00:12:02, TokenRing0
D       172.16.2.0/24 [90/2185984] via 172.16.3.1, 00:12:02, TokenRing0
C       172.16.3.0/24 is directly connected, TokenRing0
     192.168.4.0/32 is subnetted, 1 subnets
D EX    192.168.4.1 [170/2211584] via 172.16.3.1, 00:12:02, TokenRing0
     10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C       10.1.2.0/30 is directly connected, Loopback32
D       10.0.0.0/8 is a summary, 00:12:06, Null0
```

Example 11-24 *EIGRP Autosummary Null Routes (Continued)*

```
C       10.1.1.0/24 is directly connected, Loopback31
     182.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C       182.16.4.0/24 is directly connected, Loopback21
C       182.16.3.0/24 is directly connected, Loopback20
D       182.16.0.0/16 is a summary, 00:12:07, Null0
     192.168.3.0/32 is subnetted, 1 subnets
D EX    192.168.3.1 [170/2211584] via 172.16.3.1, 00:12:03, TokenRing0
canada_2#
```

- For autosummarization to work properly, discontinuous subnets at the major bit boundaries must be avoided at all costs. Unfortunately, on modern networks, something causes subnets to be deployed in places they shouldn't be, and EIGRP has forwarding problems. By disabling autosummarization with the **no auto-summary** command, the routes to null are not created and automatic summary routes are not forwarded. In place of autosummary, use manual summarization. Most engineers that we have worked with disable autosummarization when using EIGRP to prevent routes to the null interface.

EIGRP Manual Summarization or Route Aggregation

EIGRP manual summarization is critical to large EIGRP networks. It limits the EIGRP query and can significantly reduce the size of the routing table. There are essentially two ways to deploy manual summarization:

- Advertise an summary address or aggregate address with the following interface command:

  ip summary-address eigrp *as_number summary_address address_mask*

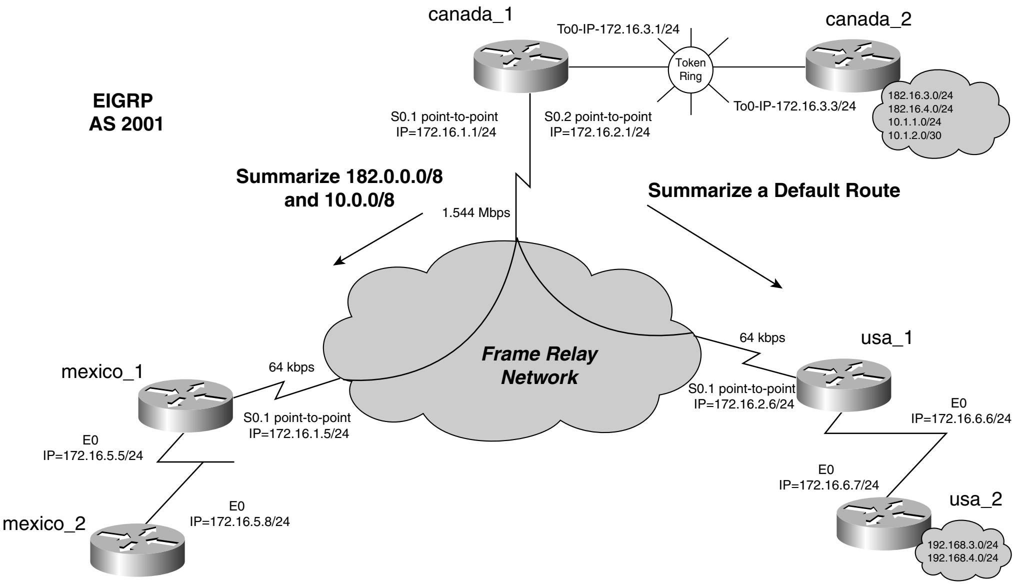
- Advertise a default route with the following interface command:

  ip summary-address eigrp *as_number 0.0.0.0 0.0.0.0.*

This command causes only the default route to be advertised; all other routing updates are suppressed.

One of the powerful functions of EIGRP is the capability to advertised multiple summary routes and default routes on different interfaces. The EIGRP network in Figure 11-10 is under the same autonomous system, and autosummary has been disabled on all the routers. In this model, EIGRP is configured to advertise a default route out the s0.2 port on the canada_1 router to the usa_1 router. The canada_1 router also advertises two summary routes, 182.0.0.0/8 and 10.0.0.0/8, out the s0.1 interface to the mexico_1 router.

Figure 11-10 *EIGRP Manual Summarization*



Example 11-25 lists the configuration of canada_1 router performing manual summarization.

Example 11-25 *Manual Summarization on canada_1 Serial Interfaces*

```
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 point-to-point
 ip address 172.16.1.1 255.255.255.0
 ip summary-address eigrp 2001 182.0.0.0 255.0.0.0    ←Manual summarization
 ip summary-address eigrp 2001 10.0.0.0 255.0.0.0
 frame-relay interface-dlci 110
!
interface Serial0.2 point-to-point
 ip address 172.16.2.1 255.255.255.0
 ip summary-address eigrp 2001 0.0.0.0 0.0.0.0    ←Advertise a default route only
 frame-relay interface-dlci 130
!
interface TokenRing0
 ip address 172.16.3.1 255.255.255.0
 ring-speed 16
!
```

Example 11-25 *Manual Summarization on canada_1 Serial Interfaces (Continued)*

```
router eigrp 2001
 network 172.16.0.0
 no auto-summary
!
```

Example 11-26 lists the route table of the mexico_1 router and the usa_1 router. Notice how the routes are summarized. The usa_1 router receives only a default route 0.0.0.0, and the gateway of last resort is set.

Example 11-26 *Route Tables of mexico_1 and usa_1 with Summarization Applied*

```
mexico_1#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 5 subnets
C       172.16.5.0 is directly connected, Ethernet0
D       172.16.6.0 [90/2707456] via 172.16.1.1, 00:34:11, Serial0.1
C       172.16.1.0 is directly connected, Serial0.1
D       172.16.2.0 [90/2681856] via 172.16.1.1, 00:34:11, Serial0.1
D       172.16.3.0 [90/2185984] via 172.16.1.1, 00:34:11, Serial0.1
D    192.168.4.0/24 [90/2835456] via 172.16.1.1, 00:34:11, Serial0.1
D    10.0.0.0/8 [90/2313984] via 172.16.1.1, 00:34:11, Serial0.1   ←Summary Route
D    192.168.3.0/24 [90/2835456] via 172.16.1.1, 00:34:11, Serial0.1
D    182.0.0.0/8 [90/2313984] via 172.16.1.1, 00:34:11, Serial0.1   ←Summary Route

mexico_1#
```

```
usa_1#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR

Gateway of last resort is 172.16.2.1 to network 0.0.0.0

D    192.168.3.0/24 [90/409600] via 172.16.6.7, 00:45:52, Ethernet0
D    192.168.4.0/24 [90/409600] via 172.16.6.7, 00:45:52, Ethernet0
     172.16.0.0/24 is subnetted, 2 subnets
```

continues

Example 11-26 *Route Tables of mexico_1 and usa_1 with Summarization Applied (Continued)*

```
C       172.16.6.0 is directly connected, Ethernet0
C       172.16.2.0 is directly connected, Serial0.1
D*   0.0.0.0/0 [90/2185984] via 172.16.2.1, 00:44:54, Serial0.1
usa_1#
```

NOTE In Cisco IOS Software 12.0(4)T, an administrative distance can be added to the summary address to alter the default admin distance of 90.

Default Routing with EIGRP

A default route can be injected into EIGRP in two primary ways:

- Redistribute a default static route into EIGRP so that EIGRP recognizes the route of 0.0.0.0 to be the default route. A default static route is created with the global router entry **ip route 0.0.0.0 0.0.0.0** *next_hop_IP_address*. This route then must be redistributed into EIGRP with the **redistribute static** command. If the network 0.0.0.0 is not used, you can still mark the route as a default route by using the **ip default-network** *a.b.c.d* command.
- Summarize a default route of 0.0.0.0 with the interface command **ip summary-address eigrp** *as_number* **0.0.0.0 0.0.0.0**. The example in the previous section demonstrated how to propagate a default route with this command.

With both ways, the router needs the **ip classless** global command enabled. With IP classless, the router forwards any packets toward the default route that it does not have a more specific route toward. **ip classless** is enabled by default in Cisco IOS Software Release 12.0 and later.

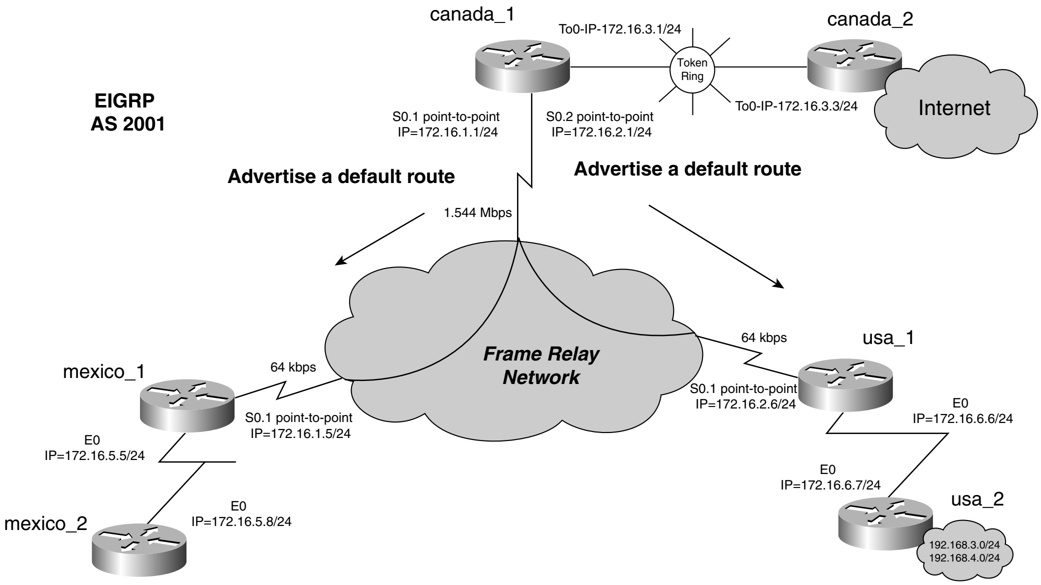
In Figure 11-11, the canada_1 router advertises a default route to usa_1 and mexico_1. canada_1 does this by creating a static route pointing at a next-hop address of 172.16.3.3 or the canada_2 router.

Example 11-27 lists the configuration of the canada_1 router advertising a default static route.

Example 11-27 *Advertising a Default Static Route with EIGRP on canada_1*

```
router eigrp 2001
 redistribute static        ←redistribute the static routes
 network 172.16.0.0
 default-metric 16000 630 254 1 1500    ←Don't forget the default-metric
 no auto-summary
!
ip classless                ←IP classless must be enabled for default routing
ip route 0.0.0.0 0.0.0.0 172.16.3.3   ←The default route points at Canada_1 router
!
```

Figure 11-11 *EIGRP Default Routing*



Example 11-28 lists the route table of the mexico_1 router demonstrating how the default route is received. Notice that when the route is advertised, it is an external router because it is redistributed, and it has the * denoting it is the default route. The gateway of last resort also is set.

Example 11-28 *Route Table of the mexico_1 Router*

```
mexico_1#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is 172.16.1.1 to network 0.0.0.0

     172.16.0.0/24 is subnetted, 5 subnets
C       172.16.5.0 is directly connected, Ethernet0
D       172.16.6.0 [90/2707456] via 172.16.1.1, 00:04:15, Serial0.1
C       172.16.1.0 is directly connected, Serial0.1
D       172.16.2.0 [90/2681856] via 172.16.1.1, 00:04:51, Serial0.1
D       172.16.3.0 [90/2185984] via 172.16.1.1, 00:04:51, Serial0.1
```

continues

Example 11-28 *Route Table of the mexico_1 Router (Continued)*

```
D    192.168.4.0/24 [90/2835456] via 172.16.1.1, 00:04:15, Serial0.1
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.1.2.0/30 [90/2313984] via 172.16.1.1, 00:04:51, Serial0.1
D       10.1.1.0/24 [90/2313984] via 172.16.1.1, 00:04:52, Serial0.1
     182.16.0.0/24 is subnetted, 2 subnets
D       182.16.4.0 [90/2313984] via 172.16.1.1, 00:04:52, Serial0.1
D       182.16.3.0 [90/2313984] via 172.16.1.1, 00:04:52, Serial0.1
D    192.168.3.0/24 [90/2835456] via 172.16.1.1, 00:04:15, Serial0.1
D*EX 0.0.0.0/0 [170/2331136] via 172.16.1.1, 00:00:53, Serial0.1
mexico_1#
```

EIGRP Stub Routing

In Cisco IOS Software Release 12.0(7)T, Cisco introduced EIGRP stub routing to further control stability and reduce resource utilization. This feature was fully integrated into Release 12.0(15)S. EIGRP stub routing functions very much like that of an OSPF stub area. The stub router has one exit path from the routing domain and forwards all traffic to a central or distribution router. Another way to say this is that the stub network cannot be a transit router for EIGRP, and it can have only one EIGRP neighbor.

When configuring EIGRP stub routing, only the remote or the spoke router needs to be configured as a stub. This router responds to queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message "inaccessible." This process greatly reduces the overhead associated with responding to queries by the remote routers. The stub router also sends special peer information to its neighbor informing its neighbor that it is a stub router.

To configure EIGRP stub routing, use the following router command under EIGRP:

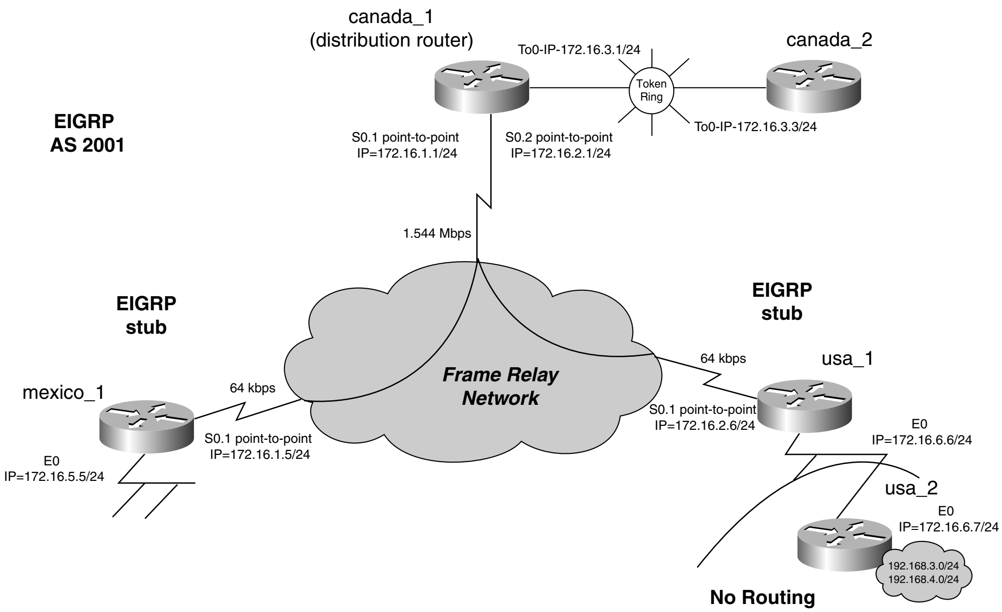
```
Router(config-router)#eigrp stub [receive-only | connected | static | summary]
```

The options are described as follows:

- **receive-only**—This causes the router to not send any routes.
- **connected**—The router advertises all connected routes to the single neighbor. No redistribution is necessary.
- **static**—The router advertises all static routes to a single neighbor. The static routes still need to be redistributed into EIGRP to be advertised.
- **summary**—The router advertises summary routes.

A stub router can be configured to advertise connected and static routes at the same time, which is the case in most stub domains. Figure 11-12 shows two EIGRP stub networks configured. The mexico_1 router is configured as a stub router that advertises only its local Ethernet network. The usa_1 router advertises its local Ethernet network along with two static routes to the 192.168.3.0/24 and 192.168.4.0/24 networks. The usa_2 router has a default gateway pointing to 172.16.6.6 and has no routing enabled. The distribution router is canada_1. No additional EIGRP configuration is necessary on the distribution router.

Figure 11-12 *EIGRP Stub Routing*



Example 11-29 lists the EIGRP configuration of the mexico_1 and the usa_1 routers.

Example 11-29 *EIGRP Stub Configuration*

```
!
hostname mexico_1
!
router eigrp 2001
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
 no auto-summary
 eigrp stub connected        ←Set EIGRP stub, and advertise connected routes
!
```

```
!
hostname usa_1
!
router eigrp 2001
 redistribute static        ←Redistribute static
```

continues

Example 11-29 *EIGRP Stub Configuration (Continued)*

```
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
 no auto-summary
eigrp stub connected static    ←Set EIGRP stub, and advertise connected and static
                                routes
!
ip classless
ip route 192.168.3.0 255.255.255.0 172.16.6.7
ip route 192.168.4.0 255.255.255.0 172.16.6.7
```

Finally, by viewing the canada_1 route table, all routes are being reported by EIGRP in the correct manner. Example 11-30 lists the route table of the canada_1 router.

Example 11-30 *Route Table of canada_1 with Two EIGRP Stub Domains*

```
canada_1#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 5 subnets
D       172.16.5.0 [90/2195456] via 172.16.1.5, 01:03:47, Serial0.1
D       172.16.6.0 [90/2195456] via 172.16.2.6, 00:48:40, Serial0.2
C       172.16.1.0 is directly connected, Serial0.1
C       172.16.2.0 is directly connected, Serial0.2
C       172.16.3.0 is directly connected, TokenRing0
D EX 192.168.4.0/24 [170/2195456] via 172.16.2.6, 00:43:27, Serial0.2
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.1.2.0/30 [90/304128] via 172.16.3.3, 07:31:49, TokenRing0
D       10.1.1.0/24 [90/304128] via 172.16.3.3, 07:31:49, TokenRing0
     182.16.0.0/24 is subnetted, 2 subnets
D       182.16.4.0 [90/304128] via 172.16.3.3, 07:31:49, TokenRing0
D       182.16.3.0 [90/304128] via 172.16.3.3, 07:31:49, TokenRing0
D EX 192.168.3.0/24 [170/2195456] via 172.16.2.6, 00:43:28, Serial0.2
canada_1#
```

To verify that the router is configured as an EIGRP stub router, use the **show ip eigrp neighbor detail** command. The last line of the output will show whether stub routing is enabled and what the stub router can advertise. This output can be viewed in Example 11-41. The show **eigrp packet stub** command shows debug information about the stub status of the peer routers.

EIGRP Equal- and Unequal-Cost Load Balancing

By default, EIGRP load-shares over four equal-cost paths. For load sharing to happen, the routes to load-share over must show up in the IP forwarding table or with the **show ip route** command. Only when a route shows up in the forwarding table with multiple paths to it will load sharing occur. Use the **bandwidth** interface command on serial links to ensure that EIGRP has a consistent perspective of the metrics of the network. This also might aid in making the route show up in the IP forwarding table.

EIGRP also has the capability to use unequal-cost load balancing in the same manner as IGRP. The router uses variance as a multiplier in choosing the upper boundary of the path with the greatest metric.

Configuring EIGRP unequal-cost load balancing is a three-step process:

Step 1 Configure the bandwidth on both sides of all the interfaces involved in the load-sharing group. Use the **bandwidth** *xx_kbps* command to accomplish this.

Step 2 Define the lowest-cost metric and the highest-cost metric. From these values, compute the variance multiplier and add it to the EIGRP routing process. The composite metric that EIGRP is using can be viewed with the **show ip eigrp topology** command, as discussed in previous sections.

Step 3 (Optional) Set the *maximum-paths* or the *traffic-share* variables.

The following example walks through the calculation of a fictional variance. EIGRP has a route whose metric is 100. The router also has two more routes to that same destination whose metrics are 200 and 300. To allow EIGRP to use all three paths in sharing data, you would set the variance to 3:

$$3 \times 100 = 300$$

Another way to view it is as variance × lowest_metric = largest metric of path to load-share over—in this case, 300. To properly set the variance in a real network, use the following formula:

Variance = 1 + ([*metric of highest cost route / metric of the lowest cost route*], rounded up to the nearest 1s decimal place)

The metric of the lowest-cost and highest-cost routes can be discovered with the **show ip eigrp topology** command. Be sure to change variance and any other variables, such as bandwidth, on both ends of the link. The bandwidth should be set on all serial links. The following is the syntax for the commands used in configuring load balancing:

```
Router(config-router)variance [metric_multiplier 1-128]
Router(config-router)maximum-paths [1-6]
Router(config-router)traffic-share {balanced | min across-interface]
Router(config-router)bandwidth xx kbps
```

The **variance** command defines the metric multiplier of which routes to use in unequal-cost load balancing. The default variance is 1, which is equal-cost load balancing.

With the **maximum-paths** command, the router uses up to six paths to share traffic across; to limit this number, use the **maximum-paths** command. The multiple paths that make up a single-hop transport to a common destination are called a *load-sharing group*. The default value is 4.

With the **traffic-share** command, if there are multiple minimum-cost paths and **traffic-share-min** is configured, EIGRP will use equal-cost load balancing. By default, the command is set to **balanced**, where traffic will be distributed proportionally to the ratio of the metrics. For example, if variance is set to 3 and traffic-share is set to balanced, the best route will transport traffic three times that of the worst route.

For a route to be included in unequal-cost load sharing, three other conditions must be met:

- The maximum-paths limit must not be exceeded as a result of adding this route to the load-sharing group.
- The downstream router must be metrically closer to the destination.
- The metric of the lowest-cost route, multiplied by the variance, must be greater than the metric of the route to be added to the load-sharing group.

Chapter 10 provides a detailed example of load sharing over IGRP, which is syntactically identical to configuring EIGRP traffic load balancing.

Lab 22: EIGRP Route Redistribution, Summarization, and Stub Routing—Part I

Practical Scenario

As EIGRP networks continue to grow in popularity, it becomes increasingly important to be able to control the query range and integrate it with other routing protocols, such as RIP and OSPF. It is also important to understand the default setting of EIGRP, such as split horizon and autosummarization.

This lab gives you practice in controlling the query range, integrating EIGRP with other routing protocols, and performing summarization.

Lab Exercise

Cisco Training Partners provide custom-tailored Cisco courses around the United States and are in the process of integrating their training facilities over a common network. Your task is to configure an EIGRP network using the following parameters as design guidelines:

- Configure an IP network, as depicted in Figure 11-13, using EIGRP as the routing protocol and 65001 as the Autonomous System ID.
- Configure the Frame Relay network as a multipoint network between the wisconsin, georgia, and ohio routers. Configure the Frame Relay network as a point-to-point network between the wisconsin and minnesota routers.
- Ensure full IP reachability with the RIP domain without using static routes or advertising a default route.
- (Optional) Configure the georgia and ohio routers as EIGRP stub routers. The routers should advertise their local LAN networks in EIGRP.

Lab Objectives

- Configure the Cisco Training Partners Network as depicted in Figure 11-13. Configure IP as denoted in the diagram. The LAN topology type is not important in this lab.
- Use Frame Relay data link protocol on the WAN.
- Configure redistribution between RIP and EIGRP.
- Ensure full IP connectivity to all IP interfaces—that is, be sure that you can **ping** all Frame Relay and LAN interfaces from the RIP domain. Also ensure that the georgia and ohio routers can **ping** each others' Frame Relay and LAN interfaces. You cannot configure any static routes or default routes on the network.

- (Optional) Configure the georgia and ohio routers as EIGRP stub routers. You will need Cisco IOS Software Release 12.0(7)T or Cisco IOS Release 12.0(15)S or 12.1 and later, the T and S trains are required.

Equipment Needed

- Six Cisco routers. Four will be connected through V.35 back-to-back cables or in a similar manner to a Frame Relay switch.
- Four LAN segments, provided through hubs or switches. The LAN topology is not significant to this lab.

Physical Layout and Prestaging

- Connect the hubs and serial cables to the routers, as shown in Figure 11-13.
- The stillwater router will run RIP only on network 172.16.0.0. Configure this router at this time. See Chapter 9 if you need assistance on this.
- A Frame Relay switch with three PVCs also is required. Example 11-31 lists the Frame Relay configuration used in this lab.

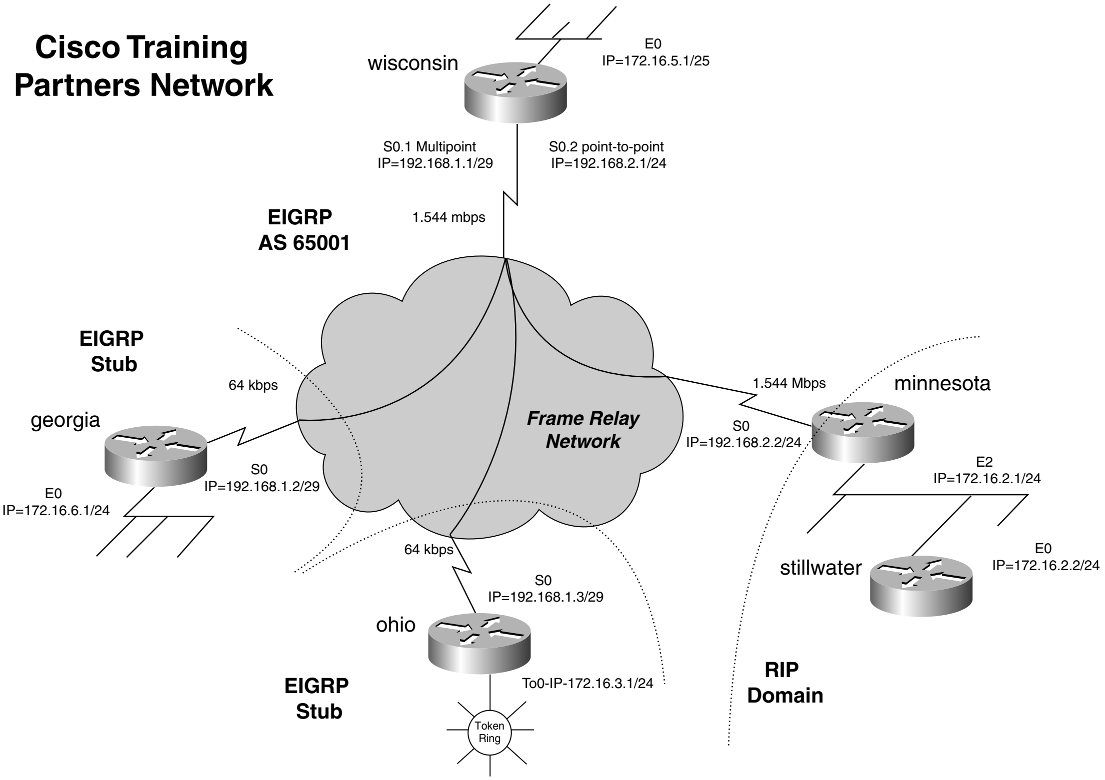
Example 11-31 *Frame Relay Switch Configuration*

```
hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 148000
 frame-relay intf-type dce
 frame-relay route 111 interface Serial1 110
 frame-relay route 121 interface Serial3 102
 frame-relay route 150 interface Serial5 151
!
interface Serial1
 no ip address
 encapsulation frame-relay
 clockrate 148000
 frame-relay intf-type dce
 frame-relay route 110 interface Serial0 111
!
<<<text omitted>>>
!
interface Serial3
```

Example 11-31 *Frame Relay Switch Configuration (Continued)*

```
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 102 interface Serial0 121
!
<<<text omitted>>>
!
interface Serial5
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 151 interface Serial0 150
!
```

Figure 11-13 *Cisco Training Partners Network*



Lab 22: EIGRP Route Redistribution, Summarization, and Stub Routing—Part II

Lab Walkthrough

Configure the Frame Relay switch and attach the four routers in a back-to-back manner to the Frame Relay switch. Use V.35 cables or CSU/DSUs with crossover cables, to connect the routers. Create the four LANs by using switches or hubs/MAUs, as illustrated in Figure 11-13.

When the physical connections are complete, assign IP addresses to all LAN and WAN interfaces, as depicted in Figure 11-13. Be sure that you can **ping** each router's local LAN and WAN interface before moving on. The wisconsin router will need subinterfaces; one will be a multipoint interface and one will be a point-to-point interface. You will use **frame-relay map** statements on the multipoint interface. You will use **frame-relay interface-dlci** commands on the point-to-point interface between the wisconsin and minnesota routers. For full IP connectivity, you will need an additional **frame-relay map** statement on ohio and georgia pointing toward each other. Example 11-32 lists the Frame Relay configuration, to this point, on all routers involved.

Example 11-32 *Frame Relay Configuration on wisconsin, georgia, ohio, and minnesota Routers*

```
!
hostname wisconsin
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 ip address 192.168.1.1 255.255.255.248
 no ip directed-broadcast
 frame-relay map ip 192.168.1.2 121 broadcast
 frame-relay map ip 192.168.1.3 150 broadcast
!
interface Serial0.2 point-to-point
 ip address 192.168.2.1 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 111
```

Example 11-32 *Frame Relay Configuration on wisconsin, georgia, ohio, and minnesota Routers (Continued)*

```
!
hostname georgia
!
<<<text omitted>>>
!
interface Serial0
 ip address 192.168.1.2 255.255.255.248
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 192.168.1.1 102 broadcast
 frame-relay map ip 192.168.1.3 102 broadcast
 frame-relay lmi-type cisco
!
```

```
hostname ohio
!
enable password cisco
!
<<<text omitted>>>
!
interface Serial0
 ip address 192.168.1.3 255.255.255.248
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 192.168.1.1 151 broadcast
 frame-relay map ip 192.168.1.2 151 broadcast
 frame-relay lmi-type cisco
!
```

```
hostname minnesota
!
<<<text omitted>>>
!
interface Serial0
 ip address 192.168.2.2 255.255.255.0
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay interface-dlci 110
```

After local WAN and LAN connectivity has been established, the network configuration will be divided into two parts. First, you will configure the EIGRP domain and then you will integrate RIP.

The basic EIGRP configuration will be similar on all the routers. Following the three-step process, begin by enabling EIGRP on all the routers using the **router eigrp 65001** command. The second step is to define the networks to run EIGRP on. The wisconsin, georgia, and ohio routers will route EIGRP on the major networks of 172.16.0.0 and 192.168.1.0.

Therefore, use these networks for your **network** statements. The wisconsin and minnesota routers will run EIGRP on 192.168.2.0 in addition to 172.16.0.0. Because this is a Frame Relay network, it's a good idea to set the bandwidth statements. Set the **bandwidth** to 128 kbps on the wisconsin interface s0.1 to accommodate the two 64-kbps PVCs. The georgia and ohio routers should have the **bandwidth** set to 64 kbps on the Frame Relay interfaces. The default bandwidth is 1.544 Mbps (T1 speed), so there is no need to modify it on the S0.2 interface on the wisconsin router. Example 11-33 lists the configuration of the wisconsin router to this point.

Example 11-33 *Configuration of the wisconsin Router*

```
hostname wisconsin
!
interface Ethernet0
 ip address 172.16.5.1 255.255.255.128
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 bandwidth 128
 ip address 192.168.1.1 255.255.255.248
 no ip directed-broadcast
 frame-relay map ip 192.168.1.2 121 broadcast
 frame-relay map ip 192.168.1.3 150 broadcast
!
interface Serial0.2 point-to-point
 ip address 192.168.2.1 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 111
!
router eigrp 65001
 network 172.16.0.0
 network 192.168.1.0
 network 192.168.2.0
!
```

At first glance, it might appear that routing is working. After all, you have a route table and three EIGRP neighbors on the wisconsin router. But some EIGRP defaults must be disabled to make the network route properly. Notice in Example 11-34 that EIGRP has three neighbors in the route table. Unfortunately, the router also has injected a couple routes to null in its forwarding table.

Example 11-34 show ip route *and* **show ip eigrp neighbors** *Command Output on the wisconsin Router*

```
wisconsin#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.5.0/25 is directly connected, Ethernet0
D       172.16.0.0/16 is a summary, 00:10:58, Null0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
D       192.168.1.0/24 is a summary, 00:11:33, Null0
C       192.168.1.0/29 is directly connected, Serial0.1
C    192.168.2.0/24 is directly connected, Serial0.2
wisconsin#
wisconsin#show ip eigrp neighbors
IP-EIGRP neighbors for process 65001
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq Type
                                        (sec)         (ms)       Cnt Num
2   192.168.1.2             Se0.1        171 00:14:00  768  4608  0  4
1   192.168.1.3             Se0.1        152 00:14:11 1544  5000  0  4
0   192.168.2.2             Se0.2        157 00:14:22    0  3000  0  11
wisconsin#
```

If you try to **ping** any routers in the 172.16.0.0 domain, it will fail. This is because the router is forwarding those packets to its null interface.

You need to correct two problems:

- The network has discontinuous subnets at the major bit boundaries. The major network 172.16.0.0/16 is divided by the networks 192.168.1.0/29 and 192.168.2.0/24. To correct this problem, disable EIGRP autosummarization on all routers in the internetwork with the **no auto-summary** EIGRP router command.
- Split horizon must be corrected, although this problem won't manifest itself until autosummarization is disabled.

Example 11-35 lists the route table of the wisconsin router now that autosummarization is disabled throughout the network. Notice that the 172.16.2.0/24, 172.16.3.0/24, and 172.16.5.0/24 routes are now in the forwarding table.

Example 11-35 show ip route *Command Output on the wisconsin Router*

```
wisconsin#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.5.0/25 is directly connected, Ethernet0
D       172.16.6.0/24 [90/20537600] via 192.168.1.2, 00:01:47, Serial0.1
D       172.16.2.0/24 [90/40537600] via 192.168.2.2, 00:54:38, Serial0.2
D       172.16.3.0/24 [90/20528128] via 192.168.1.3, 00:08:32, Serial0.1
     192.168.1.0/29 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0.1
C    192.168.2.0/24 is directly connected, Serial0.2
wisconsin#
```

As mentioned previously, the other problem that you need to remedy is split horizon. If you test IP connectivity from strictly the wisconsin router, everything would appear normal. However, upon examining the forwarding table of the georgia and ohio routers, you will see that the georgia router does not have the 172.16.3.0/24 subnet. The ohio router also does not have the 172.16.6.0/24 subnet, as shown in Example 11-36.

Example 11-36 show ip route *Command Output on the ohio and georgia Routers*

```
ohio#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
D       172.16.5.0/25 [90/40537600] via 192.168.1.1, 00:00:52, Serial0
D       172.16.2.0/24 [90/41049600] via 192.168.1.1, 00:00:52, Serial0
C       172.16.3.0/24 is directly connected, TokenRing0
     192.168.1.0/29 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0
D    192.168.2.0/24 [90/41024000] via 192.168.1.1, 00:00:53, Serial0
```

Example 11-36 **show ip route** *Command Output on the ohio and georgia Routers (Continued)*

```
ohio#
```

```
georgia#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
D       172.16.5.0/25 [90/40537600] via 192.168.1.1, 00:01:21, Serial0
C       172.16.6.0/24 is directly connected, Ethernet0
D       172.16.2.0/24 [90/41049600] via 192.168.1.1, 00:01:21, Serial0
     192.168.1.0/29 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0
D    192.168.2.0/24 [90/41024000] via 192.168.1.1, 00:01:21, Serial0
georgia#
```

These routes are not being propagated because of EIGRP split horizon. This can be verified with the **debug ip eigrp packets** command. To allow updates to flow properly across a multipoint network, disable split horizon on that interface with the **no ip split-horizon eigrp** command on the wisconsin s0.1 interface:

```
wisconsin(config)#int s0.1
wisconsin(config-subif)#no ip split-horizon eigrp 65001
```

Example 11-37 lists the forwarding tables of the ohio and georgia routers after split horizon has been disabled on the wisconsin router.

Example 11-37 **show ip route** *Command Output on the ohio and georgia Routers*

```
ohio#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.5.0/25 [90/40537600] via 192.168.1.1, 00:00:04, Serial0
D       172.16.6.0/24 [90/41049600] via 192.168.1.1, 00:00:04, Serial0
D       172.16.2.0/24 [90/41049600] via 192.168.1.1, 00:00:04, Serial0
```

continues

Example 11-37 **show ip route** *Command Output on the ohio and georgia Routers (Continued)*

```
C       172.16.3.0/24 is directly connected, TokenRing0
     192.168.1.0/29 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0
D    192.168.2.0/24 [90/41024000] via 192.168.1.1, 00:00:04, Serial0
ohio#
```

```
georgia#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.5.0/25 [90/40537600] via 192.168.1.1, 00:01:41, Serial0
C       172.16.6.0/24 is directly connected, Ethernet0
D       172.16.2.0/24 [90/41049600] via 192.168.1.1, 00:01:41, Serial0
D       172.16.3.0/24 [90/41040128] via 192.168.1.1, 00:00:49, Serial0
     192.168.1.0/29 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0
D    192.168.2.0/24 [90/41024000] via 192.168.1.1, 00:01:41, Serial0
georgia#
```

At this point, you have full IP connectivity to all routers except the stillwater router, which resides in the RIP domain.

To fully integrate the RIP domain into EGIRP, you must ensure that the configuration has two elements:

- Mutual redistribution between RIP and EIGRP on the minnesota router
- All EIGRP routes summarized on a 24-bit boundary, the bit boundary the RIP network is on

To enable mutual redistribution on the minnesota router, use the **redistribution** and **default-metric** commands. Example 11-38 lists the configuration of the minnesota router.

Example 11-38 *EIGRP and RIP Configuration of the minnesota Router*

```
!
router eigrp 65001
 redistribute rip
 network 172.16.0.0
 network 192.168.2.0
 default-metric 1544 100 254 1 1500
```

Example 11-38 *EIGRP and RIP Configuration of the minnesota Router (Continued)*

```
 no auto-summary
!
router rip
 redistribute eigrp 65001
 network 172.16.0.0
 default-metric 4
!
```

The stillwater router now starts to receive routes from the minnesota router; however, it can receive only routes that have a 24-bit mask. The stillwater router will not have routes to the Frame Relay multipoint network, 192.168.1.0/29, or the Ethernet network, 182.16.5.0/25 on the wisconsin router. For the stillwater router to receive these routes, you must configure two summary addresses on a 24-bit boundary, on the point-to-point subnet between the wisconsin and minnesota router. Example 11-39 lists the configuration needed on the wisconsin router.

Example 11-39 *EIGRP Summarization on the wisconsin Router*

```
!
interface Serial0.2 point-to-point
 bandwidth 64
 ip address 192.168.2.1 255.255.255.0
 no ip directed-broadcast
 ip summary-address eigrp 65001 192.168.1.0 255.255.255.0 5
 ip summary-address eigrp 65001 172.16.5.0 255.255.255.0 5
 frame-relay interface-dlci 111
!
```

Example 11-40 lists the IP forwarding table of the stillwater router, followed by three **ping**s. To test complete IP connectivity, **ping**s have been issued from the stillwater router to the networks that were not originally on a 24-bit boundary.

Example 11-40 *The* **show ip route** *Command Followed by a* **ping** *on the stillwater Router*

```
stillwater#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR

Gateway of last resort is not set

R    192.168.1.0/24 [120/4] via 172.16.2.1, 00:00:01, Ethernet0
R    192.168.2.0/24 [120/4] via 172.16.2.1, 00:00:01, Ethernet0
     172.16.0.0/24 is subnetted, 4 subnets
```

continues

Example 11-40 *The* **show ip route** *Command Followed by a* **ping** *on the stillwater Router (Continued)*

```
R       172.16.5.0 [120/4] via 172.16.2.1, 00:00:01, Ethernet0
R       172.16.6.0 [120/4] via 172.16.2.1, 00:00:01, Ethernet0
C       172.16.2.0 is directly connected, Ethernet0
R       172.16.3.0 [120/4] via 172.16.2.1, 00:00:01, Ethernet0
stillwater#ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/70/72 ms
stillwater#ping 192.168.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/70/72 ms
stillwater#ping 172.16.5.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.5.1, timeout is 2 seconds:
!!!!!
```

The final part of this lab is optional and involves configuring the georgia and ohio routers as EIGRP stub routers. Both routers still must advertise their connected networks; therefore, they need to use the **connected** keyword with the **eigrp stub** command. The only routers that need to be configured as stub routers are georgia and ohio; no configuration is necessary on the wisconsin router. The syntax needed on both routers resembles the following:

```
georgia(config-router)#eigrp stub connected
```

To verify that a stub router is working, use the **show ip eigrp neighbors detail** command, as in Example 11-41. The last line of the output shows whether stub routing is enabled and what the stub router can advertise. **ping**s also should be issued from the RIP domain to the newly configured stub areas to verify IP routing.

Example 11-41 *Verifying Stub Routing*

```
wisconsin#show ip eigrp neighbors detail 65001
IP-EIGRP neighbors for process 65001
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq Type
                                        (sec)         (ms)       Cnt Num
2   192.168.1.3             Se0.1        178 00:00:53   52  1140  0  25
   Version 12.0/1.1, Retrans: 1, Retries: 0
   Stub Peer Advertising ( CONNECTED ) Routes
1   192.168.1.2             Se0.1        156 00:03:11  209  1254  0  28
   Version 12.0/1.1, Retrans: 0, Retries: 0
   Stub Peer Advertising ( CONNECTED ) Routes
0   192.168.2.2             Se0.2        130 01:01:01   26  2280  0  33
   Version 11.3/1.0, Retrans: 1, Retries: 0
wisconsin#
```

The last example (11-42) lists the complete configuration of the georgia, wisconsin, and minnesota routers.

Example 11-42 *Configuration listings of georgia, wisconsin, and minnesota Routers*

```
hostname georgia
!
<<<text omitted>>>
!
interface Ethernet0
 ip address 172.16.6.1 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 bandwidth 64
 ip address 192.168.1.2 255.255.255.248
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 fair-queue 64 256 0
 frame-relay map ip 192.168.1.1 102 broadcast
 frame-relay map ip 192.168.1.3 102 broadcast
 frame-relay lmi-type cisco
!
router eigrp 65001
 network 172.16.0.0
 network 192.168.1.0
 no auto-summary
 eigrp stub connected
!
```

```
hostname wisconsin
!
<<<text omitted>>>
!
interface Ethernet0
 ip address 172.16.5.1 255.255.255.128
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 bandwidth 128
 ip address 192.168.1.1 255.255.255.248
 no ip directed-broadcast
 no ip split-horizon eigrp 65001
 frame-relay map ip 192.168.1.2 121 broadcast
 frame-relay map ip 192.168.1.3 150 broadcast
!
```

continues

Example 11-42 *Configuration listings of georgia, wisconsin, and minnesota Routers (Continued)*

```
interface Serial0.2 point-to-point
 ip address 192.168.2.1 255.255.255.0
 no ip directed-broadcast
 ip summary-address eigrp 65001 192.168.1.0 255.255.255.0 5
 ip summary-address eigrp 65001 172.16.5.0 255.255.255.0 5
 frame-relay interface-dlci 111
!
interface Serial1
 no ip address
 no ip directed-broadcast
 shutdown
!
interface BRI0
 no ip address
 no ip directed-broadcast
 shutdown
 isdn guard-timer 0 on-expiry accept
!
router eigrp 65001
 network 172.16.0.0
 network 192.168.1.0
 network 192.168.2.0
 no auto-summary
```

```
hostname minnesota
!
<<<text omitted>>>
!
interface Ethernet2
 ip address 172.16.2.1 255.255.255.0
 media-type 10BaseT
!
<<<text omitted>>>
!
interface Serial0
 ip address 192.168.2.2 255.255.255.0
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay interface-dlci 110
!
router eigrp 65001
 redistribute rip
 network 172.16.0.0
 network 192.168.2.0
 default-metric 1544 100 254 1 1500
 no auto-summary
!
router rip
 redistribute eigrp 65001
 network 172.16.0.0
 default-metric 4
!
```

Lab 23: Default Routing, Route manipulation, and Filtering in EIGRP Networks—Part I

Practical Scenario

Most networks today are connected to the Internet in some form. Connecting to the Internet usually requires a default route to be propagated throughout the network. The following lab gives you practice in controlling routes and propagating a default route throughout EIGRP.

Lab Exercise

Small groups of Internet coffee shops and their suppliers have pooled to leverage a common connection to the Internet. Solar Bucks Inc., G & S INC of Sweden, and Barneys have decided to share common networks while providing new services to their customers. Some shops also have private networks and do not want them propagated to other coffee shops. Your task is to configure an EIGRP network using the following parameters as design guidelines:

- Configure an IP network as depicted in Figure 11-14, using EIGRP as the routing protocol and 2001 as the Autonomous System ID.
- Configure the Frame Relay network as a point-to-point network among all the routers. Do not create a multipoint network.
- Do not allow any other shops to see the subnet 172.16.3.0/24 on the barneys router.
- Inject a default route into the solar_bucks router pointing all traffic to the internet_router.
- The direct Frame link between solar_bucks and g_and_s router is very expensive. Configure EIGRP so that traffic from g_and_s will go first to barneys and then to solar_bucks. If the PVC between barneys and g_and_s drops, traffic will flow directly from g_and_s to solar_bucks.

Lab Objectives

- Configure the Internet Coffee Shop Network as depicted in Figure 11-14. Configure IP as denoted in the diagram. The LAN topology type is not important in this lab.
- Use the Frame Relay data link protocol on the WAN. Use only point-to-point networks on the Frame Relay network.
- Ensure full IP connectivity to all IP interfaces—that is, be sure that you can **ping** all Frame Relay and LAN interfaces except those that are filtered.
- Filter the network 172.16.3.0/24 from g_and_s and solar_bucks routers.

- Inject a default route into the solar_bucks router pointing all traffic to the internet_router.
- Control routes so that the traffic from the g_and_s router passes through barneys before it hits the Internet. Traffic to 172.16.50.0/0 also should go through the barneys router. Do not use policy routing.

Equipment Needed

- Five Cisco routers. Three will be connected through V.35 back-to-back cables or in a similar manner to a Frame Relay switch.
- Four LAN segments, provided through hubs or switches. The LAN topology is not significant to this lab. The Internet connection can be real or not; it does not affect the configuration of the router.

Physical Layout and Prestaging

- Connect the hubs and serial cables to the routers, as shown in Figure 11-14.
- Configure an additional router to serve as the connection to the Internet. Use EIGRP for the routing protocol.
- A Frame Relay switch with three PVCs also is required. Example 11-43 lists the Frame Relay configuration used in this lab.

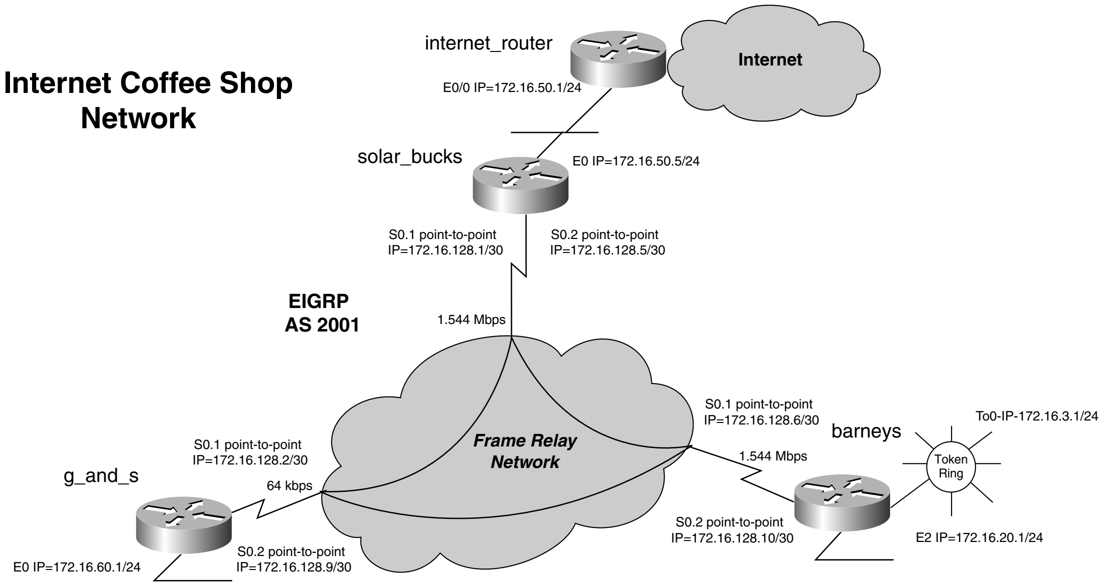
Example 11-43 *Frame Relay Switch Configuration*

```
hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 148000
 frame-relay intf-type dce
 frame-relay route 111 interface Serial1 110
 frame-relay route 121 interface Serial3 102
!
interface Serial1
 no ip address
 encapsulation frame-relay
 clockrate 148000
 frame-relay intf-type dce
 frame-relay route 110 interface Serial0 111
 frame-relay route 130 interface Serial3 131
```

Example 11-43 *Frame Relay Switch Configuration (Continued)*

```
!
interface Serial2
 no ip address
 shutdown
!
interface Serial3
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 102 interface Serial0 121
 frame-relay route 131 interface Serial1 130
!
```

Figure 11-14 *Internet Coffee Shop Network*



Lab 23: Default Routing, Route Manipulation, and Filtering in EIGRP Networks—Part II

Lab Walkthrough

Configure the Frame Relay switch and attach the three routers in a back-to-back manner to the Frame switch. Use V.35 cables or CSU/DSUs with crossover cables to connect the routers. Create the four LANs by the use of switches or hubs/MAUs, as illustrated in Figure 11-14.

When the physical connections are complete, assign IP addresses to all LAN and WAN interfaces, as depicted in Figure 11-14. Be sure that you can **ping** each routers' local LAN and WAN interface before moving on. You will use **frame-relay interface-dlci** commands on the point-to-point interfaces among all the routers. Example 11-44 lists the Frame Relay configuration, to this point, on all routers involved.

Example 11-44 *Frame Relay Configurations*

```
hostname solar_bucks
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
 ip address 172.16.128.1 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 121
!
interface Serial0.2 point-to-point
 ip address 172.16.128.5 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 111
!
```

```
hostname g_and_s
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 no ip directed-broadcast
```

Example 11-44 *Frame Relay Configurations (Continued)*

```
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
 ip address 172.16.128.2 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 102
!
interface Serial0.2 point-to-point
 ip address 172.16.128.9 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 131
!
```

```
hostname barneys
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 point-to-point
 ip address 172.16.128.6 255.255.255.252
 frame-relay interface-dlci 110
!
interface Serial0.2 point-to-point
 ip address 172.16.128.10 255.255.255.252
 frame-relay interface-dlci 130
!
```

The basic EIGRP configuration for this lab is far simpler than that of the previous lab. There are no discontinuous subnets; therefore, you do not have to disable EIGRP autosummarization. The Frame Relay network is a point-to-point network, thereby making split horizon a nonissue, as well. Following the three-step process for configuring EIGRP, you simply need to enable EIGRP routing and assign the AS number of 2001. You will use the **network** statement of 172.16.0.0 on each router. This is all that you need to configure for basic EIGRP routing. Because the PVCs to the g_and_s router are only 64 kbps, set the bandwidth to 64 on all the Frame Relay links to the g_and_s router. The EIGRP portion of the solar_bucks router, which resembles all the EIGRP configurations to this point, is presented in Example 11-45.

Example 11-45 *EIGRP Configuration of All Routers to This Point*

```
!
router eigrp 2001
 network 172.16.0.0
!
```

At this time, you can verify routing by performing source **ping**s and examining the route table. When basic routing is working, you can proceed to the next portion of the lab, which requires that barneys not propagate the subnet 172.16.3.0 throughout the EIGRP domain. There are many ways to accomplish this, but for this lab, you will use a distribution list. The list will be applied to EIGRP updates leaving the s0.1 and s0.2 interfaces on the barneys router. Example 11-46 demonstrates the configuration of an access list denying the network 172.16.3.0/24 only. Access list 10 then is called by the distribution list in EIGRP. The distribution list must be applied to serial interfaces s0.1 and s0.2 to prevent the route from leaking back into the network.

Example 11-46 *Configuration of a Distribution List*

```
barneys(config)#access-list 10 deny 172.16.3.0 0.0.0.255
barneys(config)#access-list 10 permit any
barneys(config)#router eigrp 2001
barneys(config-router)#distribute-list 10 out serial 0.1
barneys(config-router)#distribute-list 10 out serial 0.2
barneys(config-router)#^z
```

By observing the forwarding table on g_and_s in Example 11-47, you can see that the route 172.16.3.0/24 is now missing. You still can **ping** the 172.16.20.0/24 subnet, so you know that the filter was a success.

Example 11-47 *Testing a Route Filter*

```
g_and_s#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 7 subnets, 3 masks
C       172.16.128.8/30 is directly connected, Serial0.2
D       172.16.128.4/30 [90/41024000] via 172.16.128.1, 00:05:14, Serial0.1
                        [90/41024000] via 172.16.128.10, 00:05:14, Serial0.2
C       172.16.128.0/30 is directly connected, Serial0.1
```

Example 11-47 *Testing a Route Filter (Continued)*

```
C       172.16.60.0/24 is directly connected, Ethernet0
D       172.16.50.0/24 [90/40537600] via 172.16.128.1, 00:05:14, Serial0.1
D       172.16.20.0/24 [90/40537600] via 172.16.128.10, 00:05:13, Serial0.2
D       172.16.0.0/16 is a summary, 01:10:38, Null0
g_and_s#ping 172.16.20.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/41/44 ms
g_and_s#
```

The next portion of the lab requires solar_bucks to inject a default route into the EIGRP domain. To accomplish this task, configure a default static route pointing all traffic to the internet_routers Ethernet port, 172.16.50.1. For the routers to use the default network, ensure that IP classless is enabled. The static route is redistributed into EIGRP. Example 11-48 demonstrates the configuration of the default route on the solar_bucks router.

Example 11-48 *Configuring a Default Route for EIGRP*

```
solar_bucks(config)#ip route 0.0.0.0 0.0.0.0 172.16.50.1
solar_bucks(config)#router eigrp 2001
solar_bucks(config-router)#redistribute static
solar_bucks(config-router)#default-metric 1544 100 254 1 1500
solar_bucks(config-router)#^Z
solar_bucks#
```

By viewing the route or forwarding table on g_and_s or barneys, you can see that the default route is being propagated and is marked as an external, default candidate route, as shown in Example 11-49.

Example 11-49 *Viewing the Default Route on Barneys*

```
g_and_s#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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
       U - per-user static route, o - ODR

Gateway of last resort is 172.16.128.1 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 7 subnets, 3 masks
C       172.16.128.8/30 is directly connected, Serial0.2
```

continues

Example 11-49 *Viewing the Default Route on Barneys (Continued)*

```
D       172.16.128.4/30 [90/41024000] via 172.16.128.1, 00:20:43, Serial0.1
                        [90/41024000] via 172.16.128.10, 00:20:43, Serial0.2
C       172.16.128.0/30 is directly connected, Serial0.1
C       172.16.60.0/24 is directly connected, Ethernet0
D       172.16.50.0/24 [90/40537600] via 172.16.128.1, 00:20:43, Serial0.1
D       172.16.20.0/24 [90/40537600] via 172.16.128.10, 00:20:42, Serial0.2
D       172.16.0.0/16 is a summary, 01:26:07, Null0
D*EX 0.0.0.0/0 [170/40537600] via 172.16.128.1, 00:09:12, Serial0.1
g_and_s#
```

The final phase of the lab involves influencing EIGRP routing decisions. In the previous example, g_and_s is using solar_bucks as the preferred route to the Internet. By changing the delay on this link, you can affect the route table so that the barneys router is the preferred path to the Internet. To accomplish this, use the **delay 1000** command on each side of the PVC going between the g_and_s router and solar_bucks. Example 11-50 lists the route table of g_and_s, showing all routes now going through barneys first. A source trace can be performed to further test the configuration.

Example 11-50 *Route Table of g_and_s After the Delay Was Implemented*

```
g_and_s#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
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 172.16.128.10 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 7 subnets, 3 masks
C       172.16.128.8/30 is directly connected, Serial0.2
D       172.16.128.4/30 [90/41024000] via 172.16.128.10, 00:00:01, Serial0.2
C       172.16.128.0/30 is directly connected, Serial0.1
C       172.16.60.0/24 is directly connected, Ethernet0
D       172.16.50.0/24 [90/41049600] via 172.16.128.10, 00:00:01, Serial0.2
D       172.16.20.0/24 [90/40537600] via 172.16.128.10, 00:00:11, Serial0.2
D       172.16.0.0/16 is a summary, 01:28:54, Null0
D*EX 0.0.0.0/0 [170/41049600] via 172.16.128.10, 00:00:02, Serial0.2
g_and_s#
```

Example 11-51 lists the final configurations.

Example 11-51 *Final Router Configurations for the Internet Coffee Shop Network*

```
hostname solar_bucks
!
<<<text omitted>>>
!
interface Ethernet0
 ip address 172.16.50.5 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
 bandwidth 64
 ip address 172.16.128.1 255.255.255.252
 no ip directed-broadcast
 delay 1000
 frame-relay interface-dlci 121
!
interface Serial0.2 point-to-point
 ip address 172.16.128.5 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 111
!
<<<text omitted>>>
!
router eigrp 2001
 redistribute static
 network 172.16.50.0 0.0.0.255        ←Optional 12.0 way, listed for example only
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.16.50.1
```

```
hostname g_and_s
!
<<<text omitted>>>
!
interface Ethernet0
 ip address 172.16.60.1 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
```

continues

Example 11-51 *Final Router Configurations for the Internet Coffee Shop Network (Continued)*

```
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
 bandwidth 64
 ip address 172.16.128.2 255.255.255.252
 no ip directed-broadcast
 delay 1000
 frame-relay interface-dlci 102
!
interface Serial0.2 point-to-point
 bandwidth 64
 ip address 172.16.128.9 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 131
!
router eigrp 2001
 network 172.16.0.0
!
ip classless
```

```
hostname barneys
!
<<<text omitted>>>
!
interface Ethernet2
 ip address 172.16.20.1 255.255.255.0
 media-type 10BaseT
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 point-to-point
 ip address 172.16.128.6 255.255.255.252
 frame-relay interface-dlci 110
!
interface Serial0.2 point-to-point
 ip address 172.16.128.10 255.255.255.252
 bandwidth 64
 frame-relay interface-dlci 130
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.16.0.0
```

Example 11-51 *Final Router Configurations for the Internet Coffee Shop Network (Continued)*

```
 distribute-list 10 out Serial0.1
 distribute-list 10 out Serial0.2
!
ip classless
!
access-list 10 deny   172.16.3.0 0.0.0.255
access-list 10 permit any
```