C H A P T E R   1 2

Using the Border Gateway Protocol for Interdomain Routing

The Border Gateway Protocol (BGP), defined in RFC 1771, provides loop-free interdomain routing between autonomous systems. (An autonomous system [AS] is a set of routers that operate under the same administration.) BGP is often run among the networks of Internet service providers (ISPs). This case study examines how BGP works and how you can use it to participate in routing with other networks that run BGP. The following topics are covered:

- BGP Fundamentals

- BGP Decision Algorithm

- Controlling the Flow of BGP Updates

- Practical Design Example

Note   The version of BGP described in this case study is BGP Version 4.


BGP Fundamentals

This section presents fundamental information about BGP, including the following topics:

- Internal BGP

- External BGP

- BGP and Route Maps

- Advertising Networks

Routers that belong to the same AS and exchange BGP updates are said to be running *internal BGP* (IBGP), and routers that belong to different ASs and exchange BGP updates are said to be running *external BGP* (EBGP). With the exception of the **neighbor ebgp-multihop** router configuration command (described in the section "External BGP" later in this chapter), the commands for configuring EBGP and IBGP are the same. This case study uses the terms EBGP and IBGP as a reminder that, for any particular context, routing updates are being exchanged between ASs (EBGP) or within an AS (IBGP).

Figure 12-1 shows a network that demonstrates the difference between EBGP and IBGP.

Figure 12-1   EBGP, IBGP, and Multiple ASs
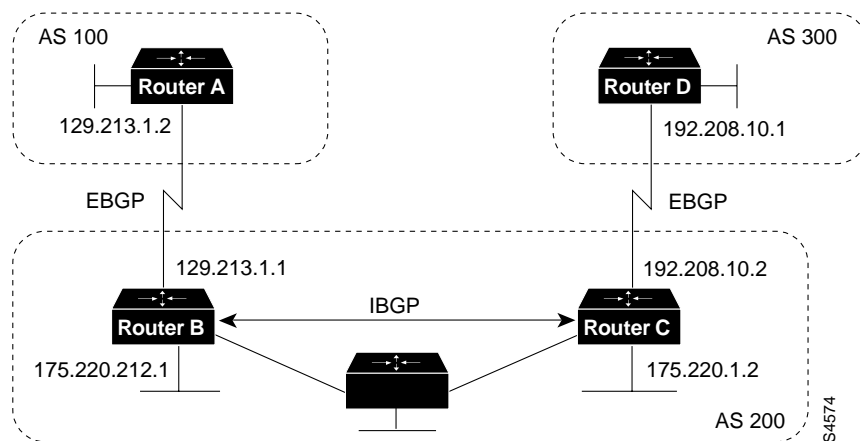
Before it exchanges information with an external AS, BGP ensures that networks within the AS are reachable. This is done by a combination of internal BGP peering among routers within the AS and by redistributing BGP routing information to Interior Gateway Protocols (IGPs) that run within the AS, such as Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), Routing Information Protocol (RIP), and Open Shortest Path First (OSPF).

BGP uses the Transmission Control Protocol (TCP) as its transport protocol (specifically port 179). Any two routers that have opened a TCP connection to each other for the purpose of exchanging routing information are known as *peers* or *neighbors*. In Figure 12-1, Routers A and B are BGP peers, as are Routers B and C, and Routers C and D. The routing information consists of a series of AS numbers that describe the full path to the destination network. BGP uses this information to construct a loop-free map of ASs. Note that within an AS, BGP peers do not have to be directly connected.

BGP peers initially exchange their full BGP routing tables. Thereafter, BGP peers send incremental updates only. BGP peers also exchange keepalive messages (to ensure that the connection is up) and notification messages (in response to errors or special conditions).

In Figure 12-1, the following commands configure BGP on Router A:

```
router bgp 100
neighbor 129.213.1.1 remote-as 200
```

The following commands configure BGP on Router B:

```
router bgp 200
neighbor 129.213.1.2 remote-as 100
neighbor 175.220.1.2 remote-as 200
```

The following commands configure BGP on Router C:

```
router bgp 200
neighbor 175.220.212.1 remote-as 200
neighbor 192.208.10.1 remote-as 300
```

The following commands configure BGP on Router D:

```
router bgp 300
neighbor 192.208.10.2 remote-as 200
```

The **router bgp** global configuration command enables a BGP routing process and assigns to it an AS number.

The **neighbor remote-as** router configuration command adds an entry to the BGP neighbor table specifying that the peer identified by a particular IP address belongs to the specified AS. For routers that run EBGP, neighbors are usually directly connected, and the IP address is usually the IP address of the interface at the other end of the connection. (For the exception to this rule, see the section "EBGP Multihop," later in this chapter.) For routers that run IBGP, the IP address can be the IP address of any of the router's interfaces.

Note the following about the ASs shown in Figure 12-1:

- Routers A and B are running EBGP, and Routers B and C are running IBGP. Note that the EBGP peers are directly connected and that the IBGP peers are not. As long as there is an IGP running that allows the two neighbors to reach one another, IBGP peers do not have to be directly connected.

- All BGP speakers within an AS must establish a peer relationship with each other. That is, the BGP speakers within an AS must be fully meshed logically. BGP4 provides two techniques that alleviate the requirement for a logical full mesh: confederations and route reflectors. For information about these techniques, see the sections "Confederations" and "Route Reflectors," later in this chapter.

- AS 200 is a *transit AS* for AS 100 and AS 300—that is, AS 200 is used to transfer packets between AS 100 and AS 300.

To verify that BGP peers are up, use the **show ip bgp neighbors** EXEC command. Following is the output of this command on Router A:

```
RouterA# show ip bgp neighbors
BGP neighbor is 129.213.1.1, remote AS 200, external link
 BGP version 4, remote router ID 175.220.212.1
 BGP state = established, table version = 3, up for 0:10:59
 Last read 0:00:29, hold time is 180, keepalive interval is 60 seconds
 Minimum time between advertisement runs is 30 seconds
 Received 2828 messages, 0 notifications, 0 in queue
 Sent 2826 messages, 0 notifications, 0 in queue
 Connections established 11; dropped 10
```

Anything other than state = established indicates that the peers are not up. The remote router ID is the highest IP address on that router (or the highest loopback interface, if there is one). Notice the table version number: each time the table is updated by new incoming information, the table version number increments. A table version number that continually increments is an indication that a route is flapping, thereby causing routes to be updated continually.


Note When you make a configuration change with respect to a neighbor for which a peer relationship has been established, be sure to reset the BGP session with that neighbor. To reset the session, at the system prompt, issue the **clear ip bgp** EXEC command specifying the IP address of that neighbor.

Internal BGP

Internal BGP (IBGP) is the form of BGP that exchanges BGP updates within an AS. Instead of IBGP, the routes learned via EBGP could be redistributed into IGP within the AS and then redistributed again into another AS. However, IBGP is more flexible, provides more efficient ways of controlling the exchange of information within the AS, and presents a consistent view of the AS to external neighbors. For example, IBGP provides ways to control the exit point from an AS.

Figure 12-2 shows a topology that demonstrates IBGP.

Figure 12-2   Internal BGP Example
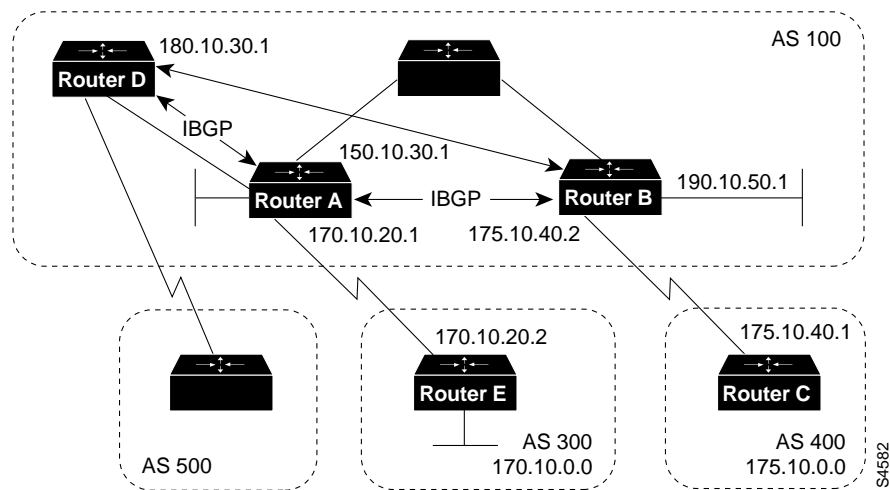
The following commands configure Routers A and B in AS 100, and Router C in AS 400:

```
!Router A
router bgp 100
neighbor 180.10.30.1 remote-as 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0

!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 175.10.40.1 remote-as 400
neighbor 180.10.30.1 remote-as 100
network 190.10.50.0

!Router C
router bgp 400
neighbor 175.10.40.2 remote-as 100
network 175.10.0.0

!Router D
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 190.10.50.1 remote as 100
network 190.10.0.0
```

When a BGP speaker receives an update from other BGP speakers in its own AS (that is, via IBGP), the receiving BGP speaker uses EBGP to forward the update to external BGP speakers only. This behavior of IBGP is why it is necessary for BGP speakers within an AS to be fully meshed.

For example, in Figure 12-2, if there were no IBGP session between Routers B and D, Router A would send updates from Router B to Router E but not to Router D. If you want Router D to receive updates from Router B, Router B must be configured so that Router D is a BGP peer.

Loopback Interfaces

Loopback interfaces are often used by IBGP peers. The advantage of using loopback interfaces is that they eliminate a dependency that would otherwise occur when you use the IP address of a physical interface to configure BGP. Figure 12-3 shows a network in which using the loopback interface is advantageous.

Figure 12-3   Use of Loopback Interfaces
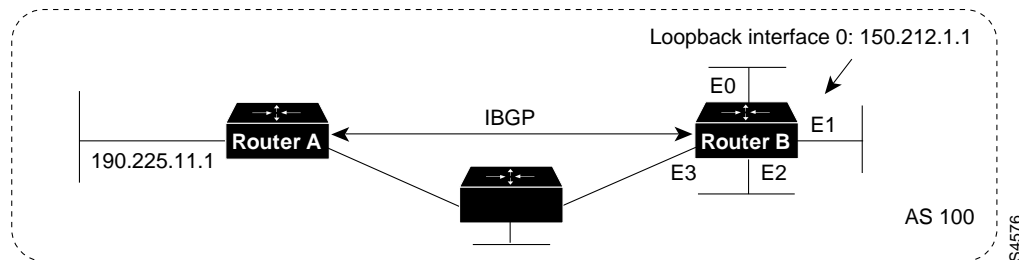
In Figure 12-3, Routers A and B are running IBGP within AS 100. If Router A were to specify the IP address of Ethernet interface 0, 1, 2, or 3 in the **neighbor remote-as** router configuration command, and if the specified interface were to become unavailable, Router A would not be able to establish a TCP connection with Router B. Instead, Router A specifies the IP address of the loopback interface that Router B defines. When the loopback interface is used, BGP does not have to rely on the availability of a particular interface for making TCP connections.

The following commands configure Router A for BGP:

```
!Router A
router bgp 100
neighbor 150.212.1.1 remote-as 100
```

The following commands configure Router B for BGP:

```
!Router B
loopback interface 0
ip address 150.212.1.1 255.255.0.0
!
router bgp 100
neighbor 190.225.11.1 remote-as 100
neighbor 190.225.11.1 update-source loopback 0
```

Router A specifies the IP address of the loopback interface (150.212.1.1) of Router B in the **neighbor remote-as** router configuration command. This use of the loopback interface requires that the configuration of Router B include the **neighbor update-source** router configuration command. When the **neighbor update-source** command is used, the source of BGP TCP connections for the specified neighbor is the IP address of the loopback interface instead of the IP address of a physical interface.

Note   Loopback interfaces are rarely between EBGP peers because EBGP peers are usually directly connected and, therefore, depend on a particular physical interface for connectivity.

External BGP

When two BGP speakers that are not in the same AS run BGP to exchange routing information, they are said to be running EBGP. This section describes commands that solve configuration problems that arise when BGP routing updates are exchanged between different ASs:

- EBGP Multihop
- EBGP Load Balancing
- Synchronization

EBGP Multihop

Usually, the two EBGP speakers are directly connected (for example, over a wide-area network [WAN] connection). Sometimes, however, they cannot be directly connected. In this special case, the **neighbor ebgp-multihop** router configuration command is used.

Note Multihop is used only for EBGP, but not for IBGP.

Figure 12-4 illustrates a topology in which the **neighbor ebgp-multihop** command is useful.

Figure 12-4   EBGP Multihop
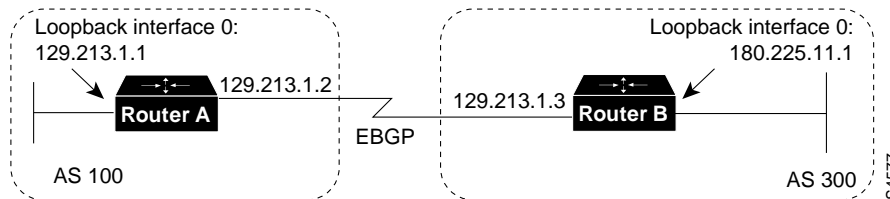
The following commands configure Router A to run EBGP:

```
!Router A
loopback interface 0
ip address 129.213.1.1
!
router bgp 100
neighbor 180.225.11.1 remote-as 300
neighbor 180.225.11.1 ebgp-multihop
neighbor 180.225.11.1 update-source loopback 0
```

The **neighbor remote-as** router configuration command specifies the IP address of an interface that is an extra hop away (180.225.11.1 instead of 129.213.1.3), and the **neighbor ebgp-multihop** router configuration command enables EGBP multihop. Because Router A references an external neighbor by an address that is not directly connected, its configuration must include static routes or must enable an IGP so that the neighbors can reach each other.

The following commands configure Router B:

```
!Router B
loopback interface 0
ip address 180.225.11.1
```

```
router bgp 300
neighbor 129.213.1.1 remote-as 100
neighbor 129.213.1.1 ebgp-multihop
neighbor 129.213.1.1 update-source loopback 0
```

EBGP Load Balancing

The **neighbor ebgp-multihop** router configuration command and loopback interfaces are also useful for configuring load balancing between two ASs over parallel serial lines, as shown in Figure 12-5.

Figure 12-5   Load Balancing over Parallel Serial Lines
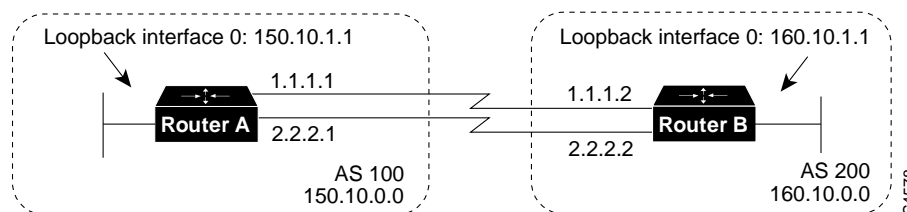
Without the **neighbor ebgp-multihop** command on each router, BGP would not perform load balancing in Figure 12-5, but with the **neighbor ebgp-multihop** command on each router, BGP uses both serial lines. The following commands configure load balancing for Router A:

```
!Router A
interface loopback 0
ip address 150.10.1.1 255.255.255.0
!
router bgp 100
neighbor 160.10.1.1 remote-as 200
neighbor 160.10.1.1 ebgp-multihop
neighbor 160.10.1.1 update-source loopback 0
network 150.10.0.0
!
ip route 160.10.0.0 255.255.0.0 1.1.1.2
ip route 160.10.0.0 255.255.0.0 2.2.2.2
```

The following commands configure load balancing for Router B:

```
!Router B
interface loopback 0
ip address 160.10.1.1 255.255.255.0
!
router bgp 200
neighbor 150.10.1.1 remote-as 100
neighbor 150.10.1.1 ebgp-multihop
neighbor 150.10.1.1 update-source loopback 0
network 160.10.0.0
!
ip route 150.10.0.0 255.255.0.0 1.1.1.1
ip route 150.10.0.0 255.255.0.0 2.2.2.1
```

The **neighbor ebgp-multihop** and **neighbor update-source** router configuration commands have the effect of making the loopback interface the next hop for EBGP, which allows load balancing to occur. Static routes are used to introduce two equal-cost paths to the destination. (The same effect

could also be accomplished by using an IGP.) Router A can reach the next hop of 160.10.1.1 in two ways: via 1.1.1.2 and via 2.2.2.2. Likewise, Router B can reach the next hop of 150.10.1.1 in two ways: via 1.1.1.1 and via 2.2.2.1.

Synchronization

When an AS provides transit service to other ASs and if there are non-BGP routers in the AS, transit traffic might be dropped if the intermediate non-BGP routers have not learned routes for that traffic via an IGP. The BGP synchronization rule states that if an AS provides transit service to another AS, BGP should not advertise a route until all of the routers within the AS have learned about the route via an IGP. The topology shown in Figure 12-6 demonstrates the synchronization rule.

Figure 12-6   Synchronization
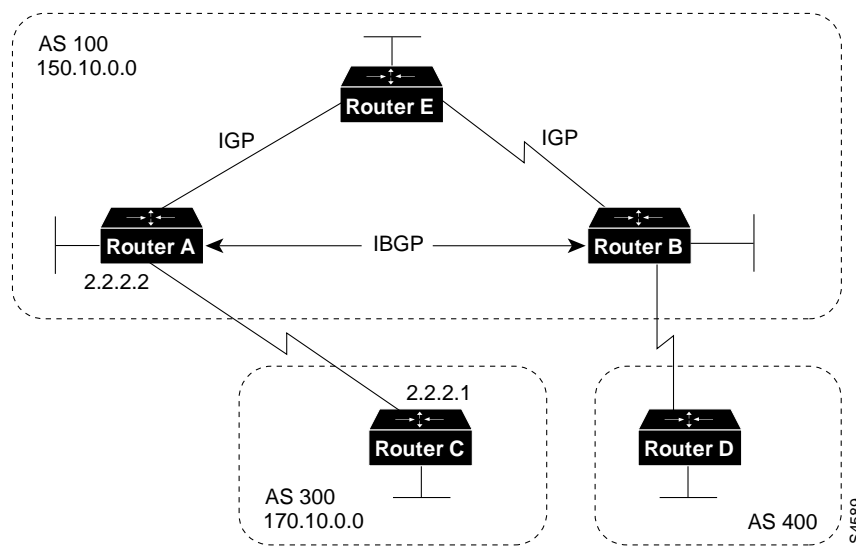
In Figure 12-6, Router C sends updates about network 170.10.0.0 to Router A. Routers A and B are running IBGP, so Router B receives updates about network 170.10.0.0 via IBGP. If Router B wants to reach network 170.10.0.0, it sends traffic to Router E. If Router A does not redistribute network 170.10.0.0 into an IGP, Router E has no way of knowing that network 170.10.0.0 exists and will drop the packets.

If Router B advertises to AS 400 that it can reach 170.10.0.0 before Router E learns about the network via IGP, traffic coming from Router D to Router B with a destination of 170.10.0.0 will flow to Router E and be dropped.

This situation is handled by the synchronization rule of BGP, which states that if an AS (such as AS 100 in Figure 12-6) passes traffic from one AS to another AS, BGP does not advertise a route before all routers within the AS (in this case, AS 100) have learned about the route via an IGP. In this case, Router B waits to hear about network 170.10.0.0 via an IGP before it sends an update to Router D. In some cases, you might want to disable synchronization. Disabling synchronization allows BGP to converge more quickly, but it might result in dropped transit packets.

You can disable synchronization if one of the following conditions is true:

- Your AS does not pass traffic from one AS to another AS.

- All the transit routers in your AS run BGP.

Figure 12-7 shows a topology in which it is desirable to disable synchronization.

Figure 12-7   Disabled Synchronization
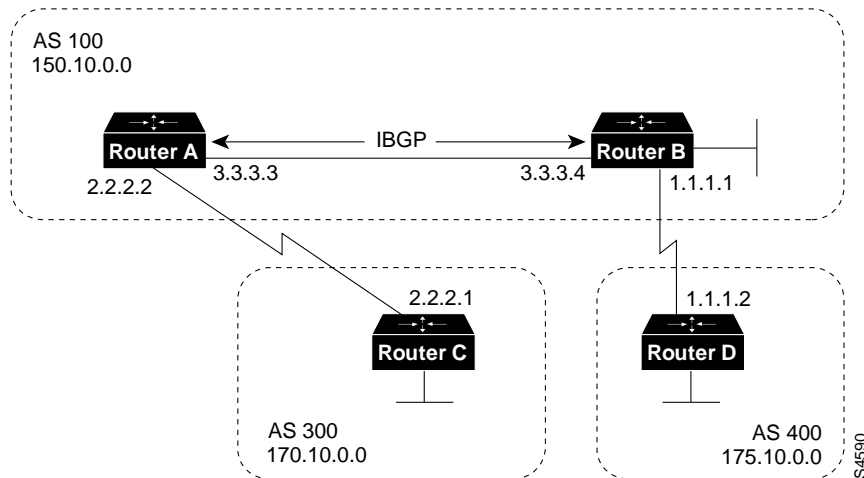
The following commands configure Routers A, B, and C:

```
!Router A
network 150.10.0.0
neighbor 3.3.3.4 remote-as 100
neighbor 2.2.2.1 remote-as 300
no synchronization

!Router B
router bgp 100
network 150.10.0.0
neighbor 1.1.1.2 remote-as 400
neighbor 3.3.3.3 remote-as 100
no synchronization

!Router D
router bgp 400
neighbor 1.1.1.1 remote-as 100
network 175.10.0.0
```

The **no synchronization** router configuration command causes Router B to put 170.10.0.0 in its IP routing table and advertise it to Router D without learning network 170.10.0.0 via an IGP.

BGP and Route Maps

Route maps are used with BGP to control and modify routing information and to define the conditions by which routes are redistributed between routing domains. The format of a route map is as follows:

```
route-map map-tag [[permit | deny] | [sequence-number]]
```

The map tag is a name that identifies the route map, and the sequence number indicates the position that an instance of the route map is to have in relation to other instances of the same route map. (Instances are ordered sequentially.)

For example, you might use the following commands to define a route map named MYMAP:

```
route-map MYMAP permit 10
! First set of conditions goes here.
route-map MYMAP permit 20
! Second set of conditions goes here.
```

When BGP applies MYMAP to routing updates, it applies the lowest instance first (in this case, instance 10). If the first set of conditions is not met, the second instance is applied, and so on, until either a set of conditions has been met, or there are no more sets of conditions to apply.

The **match** and **set** route map configuration commands are used to define the condition portion of a route map. The **match** command specifies a criteria that must be matched, and the **set** command specifies an action that is to be taken if the routing update meets the condition defined by the **match** command.

Following is an example of a simple route map:

```
route-map MYMAP permit 10
match ip address 1.1.1.1
set metric 5
```

When an update matches IP address 1.1.1.1, BGP sets the metric for the update to 5, sends the update (because of the **permit** keyword), and breaks out of the list of route-map instances.

When an update does not meet the criteria of an instance, BGP applies the next instance of the route map to the update, and so on, until an action is taken, or there are no more route map instances to apply. If the update does not meet any criteria, the update is not redistributed or controlled.

When an update meets the match criteria, and the route map specifies the **deny** keyword, BGP breaks out of the list of instances, and the update is not redistributed or controlled.

Note Route maps cannot be used to filter incoming BGP updates based on IP address. You can, however, use route maps to filter outgoing BGP updates based on IP address.

Figure 12-8 shows a topology that demonstrates the use of route maps.

Figure 12-8   Route Map Example
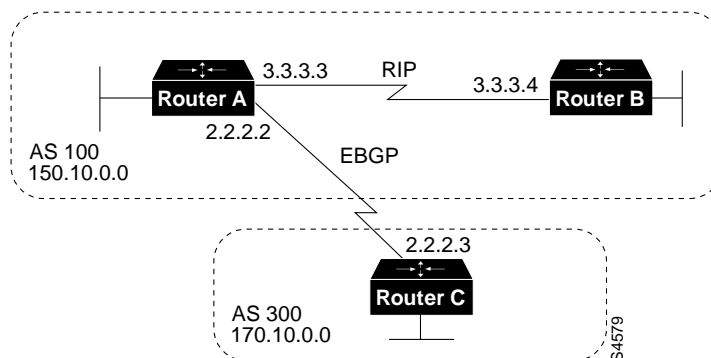
In Figure 12-8, Routers A and B run RIP with each other, and Routers A and C run BGP with each other. If you want Router A to redistribute routes from 170.10.0.0 with a metric of 2 and to redistribute all other routes with a metric of 5, use the following commands for Router A:

```
!Router A
router rip
network 3.0.0.0
network 2.0.0.0
network 150.10.0.0
passive-interface serial 0
redistribute bgp 100 route-map SETMETRIC
!
router bgp 100
neighbor 2.2.2.3 remote-as 300
network 150.10.0.0
!
route-map SETMETRIC permit 10
match ip-address 1
set metric 2
!
route-map SETMETRIC permit 20
set metric 5
!
access-list 1 permit 170.10.0.0 0.0.255.255
```

When a route matches the IP address 170.10.0.0, it is redistributed with a metric of 2. When a route does not match the IP address 170.10.0.0, its metric is set to 5, and the route is redistributed.

Assume that on Router C you want to set to 300 the community attribute of outgoing updates for network 170.10.0.0. The following commands apply a route map to outgoing updates on Router C:

```
!Router C
router bgp 300
network 170.10.0.0
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map SETCOMMUNITY out
!
route-map SETCOMMUNITY permit 10
match ip address 1
set community 300
!
access-list 1 permit 0.0.0.0 255.255.255.255
```

Access list 1 denies any update for network 170.10.0.0 and permits updates for any other network.
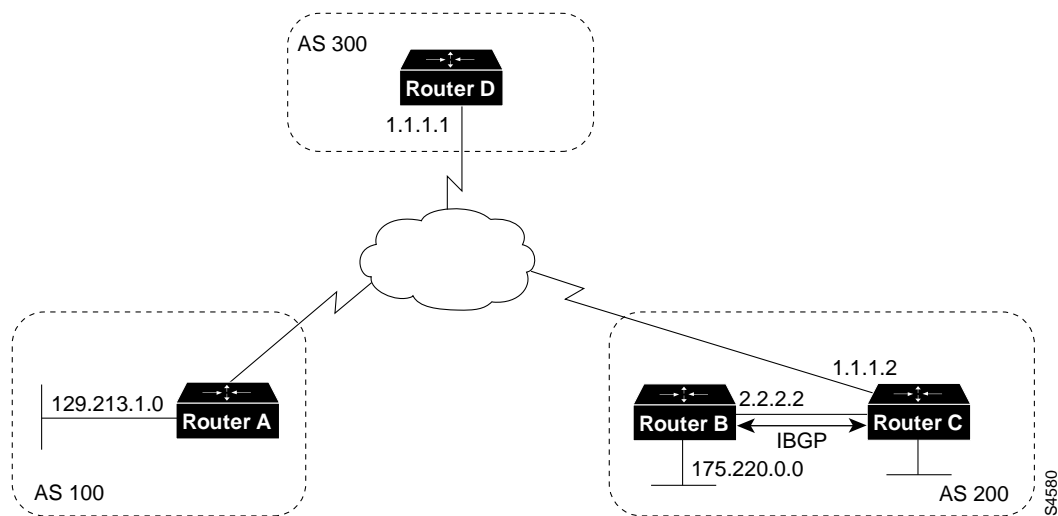
Advertising Networks

A network that resides within an AS is said to originate from that network. To inform other ASs about its networks, the AS advertises them. BGP provides three ways for an AS to advertise the networks that it originates:

- Redistributing Static Routes

- Redistributing Dynamic Routes

- Using the network Command

> **Note** It is important to remember that routes advertised by the techniques described in this section are advertised *in addition* to other BGP routes that a BGP-configured router learns from its internal and external neighbors. BGP always passes on information that it learns from one peer to other peers. The difference is that routes generated by the **network** and **redistribute** router configuration commands specify the AS of the router as the originating AS for the network.

This section uses the topology shown in Figure 12-9 to demonstrate how networks that originate from an AS can be advertised.

Figure 12-9   Network Advertisement Example 1

Redistributing Static Routes

One way to advertise that a network or a subnet originates from an AS is to redistribute static routes into BGP. The only difference between advertising a static route and advertising a dynamic route is that when you redistribute a static route, BGP sets the origin attribute of updates for the route to Incomplete. (For a discussion of other values that can be assigned to the origin attribute, see the section "Origin Attribute," later in this chapter.)

To configure Router C in Figure 12-9 to originate network 175.220.0.0 into BGP, use these commands:

```
!Router C
router bgp 200
neighbor 1.1.1.1 remote-as 300
redistribute static
!
ip route 175.220.0.0 0.0.255.255 null 0
```

The **redistribute** router configuration command and the **static** keyword cause all static routes to be redistributed into BGP.

The **ip route** global configuration command establishes a static route for network 175.220.0.0. In theory, the specification of the null 0 interface would cause a packet destined for network 175.220.0.0 to be discarded. In practice, there will be a more specific match for the packet than 175.220.0.0, and the router will send it out the appropriate interface. Redistributing a static route is the best way to advertise a supernet because it prevents the route from flapping.

Note   Regardless of route type (static or dynamic), the **redistribute** router configuration command is the only way to inject BGP routes into an IGP.

Redistributing Dynamic Routes

Another way to advertise networks is to redistribute dynamic routes. Typically, you redistribute IGP routes (such as Enhanced IGRP, IGRP, IS-IS, OSPF, and RIP routes) into BGP. Some of your IGP routes might have been learned from BGP, so you need to use access lists to prevent the redistribution of routes back into BGP.

Assume that in Figure 12-9 Routers B and C are running IBGP, that Router C is learning 129.213.1.0 via BGP, and that Router B is redistributing 129.213.1.0 back into Enhanced IGRP. The following commands configure Router C:

```
!Router C
router eigrp 10
network 175.220.0.0
redistribute bgp 200
redistributed connected
default-metric 1000 100 250 100 1500
!
router bgp 200
neighbor 1.1.1.1 remote-as 300
neighbor 2.2.2.2 remote-as 200
neighbor 1.1.1.1 distribute-list 1 out
redistribute eigrp 10
!
access-list 1 permit 175.220.0.0 0.0.255.255
```

The **redistribute** router configuration command with the **eigrp** keyword redistributes Enhanced IGRP routes for process ID 10 into BGP. (Normally, distributing BGP into IGP should be avoided because too many routes would be injected into the AS.) The **neighbor distribute-list** router configuration command applies access list 1 to outgoing advertisements to the neighbor whose IP address is 1.1.1.1 (that is, Router D). Access list 1 specifies that network 175.220.0.0 is to be advertised. All other networks, such as network 129.213.1.0, are implicitly prevented from being advertised. The access list prevents network 129.213.1.0 from being injected back into BGP as if it originated from AS 200, and allows BGP to advertise network 175.220.0.0 as originating from AS 200.

Note   Redistribution of dynamic routes requires careful use of access lists to prevent updates from being injected back into BGP. If possible, you should use the **network** command (described in the section "Using the network Command," later in this chapter) or redistribute static routes instead of redistributing dynamic routes.

Using the network Command

Another way to advertise networks is to use the **network** router configuration command. When used with BGP, the **network** command specifies the networks that the AS originates. (By way of contrast, when used with an IGP such as RIP, the **network** command identifies the interfaces on which the IGP is to run.) The **network** command works for networks that the router learns dynamically or that are configured as static routes. The origin attribute of routes that are injected into BGP by means of the **network** command is set to IGP.
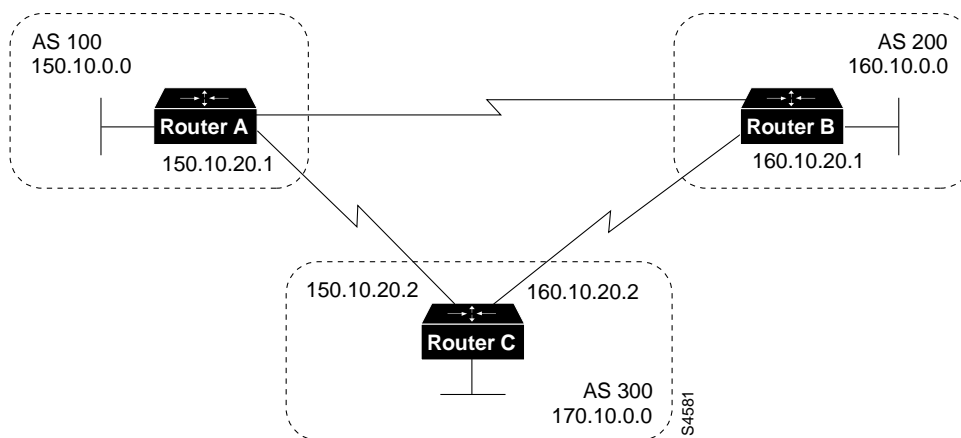
The following commands configure Router C to advertise network 175.220.0.0:

```
!Router C
router bgp 200
neighbor 1.1.1.1 remote-as 300
network 175.220.0.0
```

The **network** router configuration command causes Router C to generate an entry in the BGP routing table for network 175.220.0.0.

Figure 12-10 shows another topology that demonstrates the effects of the **network** command.

Figure 12-10   Network Advertisement Example 2

The following configurations use the **network** command to configure the routers shown in Figure 12-10:

```
!Router A
router bgp 100
neighbor 150.10.20.2 remote-as 300
network 150.10.0.0

!Router B
router bgp 200
neighbor 160.10.20.2 remote-as 300
network 160.10.0.0

!Router C
router bgp 300
neighbor 150.10.20.1 remote-as 100
neighbor 160.10.20.1 remote-as 200
network 170.10.0.0
```

To ensure a loop-free interdomain topology, BGP does not accept updates that originated from its own AS. For example, in Figure 12-10, if Router A generates an update for network 150.10.0.0 with the origin set to AS 100 and sends it to Router C, Router C will pass the update to Router B with the origin still set to AS 100. Router B will send the update (with the origin still set to AS 100) to Router A, which will recognize that the update originated from its own AS and will ignore it.

BGP Decision Algorithm

When a BGP speaker receives updates from multiple ASs that describe different paths to the same destination, it must choose the single best path for reaching that destination. Once chosen, BGP propagates the best path to its neighbors. The decision is based on the value of attributes (such as next hop, administrative weights, local preference, the origin of the route, and path length) that the update contains and other BGP-configurable factors. This section describes the following attributes and factors that BGP uses in the decision-making process:

- AS_path Attribute
- Origin Attribute
- Next Hop Attribute
- Weight Attribute
- Local Preference Attribute
- Multi-Exit Discriminator Attribute
- Community Attribute

AS_path Attribute

Whenever an update passes through an AS, BGP prepends its AS number to the update. The AS_path attribute is the list of AS numbers that an update has traversed in order to reach a destination. An AS-SET is a mathematical set of all the ASs that have been traversed.

Consider the network shown in Figure 12-11.

Figure 12-11   AS_path Attribute
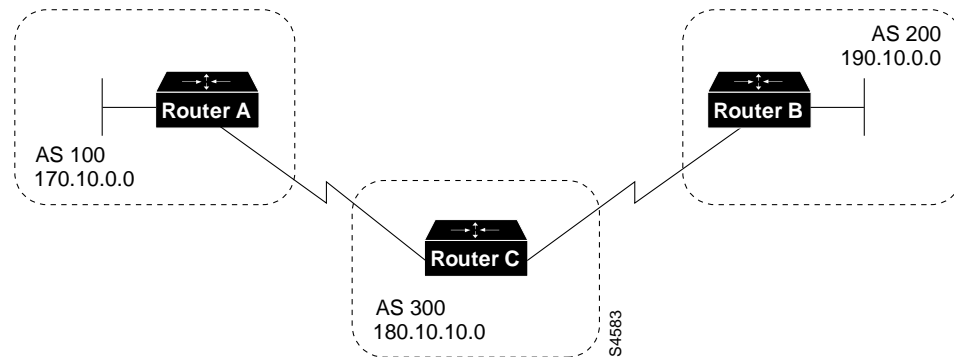
In Figure 12-11, Router B advertises network 190.10.0.0 in AS 200 with an AS_path of 200. When the update for 190.10.0.0 traverses AS 300, Router C prepends its own AS number to it, so when the update reaches Router A, two AS numbers have been attached to it: 200 and then 300. That is, the AS_path attribute for reaching network 190.10.0.0 from Router A is 300, 200. Likewise, the AS_path attribute for reaching network 170.10.0.0 from Router B is 300, 100.

Origin Attribute

The origin attribute provides information about the origin of the route. The origin of a route can be one of three values:

- *IGP*—The route is interior to the originating AS. This value is set when the **network** router configuration command is used to inject the route into BGP. The IGP origin type is represented by the letter i in the output of the **show ip bgp** EXEC command.

- *EGP*—The route is learned via the Exterior Gateway Protocol (EGP). The EGP origin type is represented by the letter e in the output of the **show ip bgp** EXEC command.

- *Incomplete*—The origin of the route is unknown or learned in some other way. An origin of Incomplete occurs when a route is redistributed into BGP. The Incomplete origin type is represented by the ? symbol in the output of the **show ip bgp** EXEC command.

Figure 12-12 shows a network that demonstrates the value of the origin attribute.

Figure 12-12   Origin Attribute
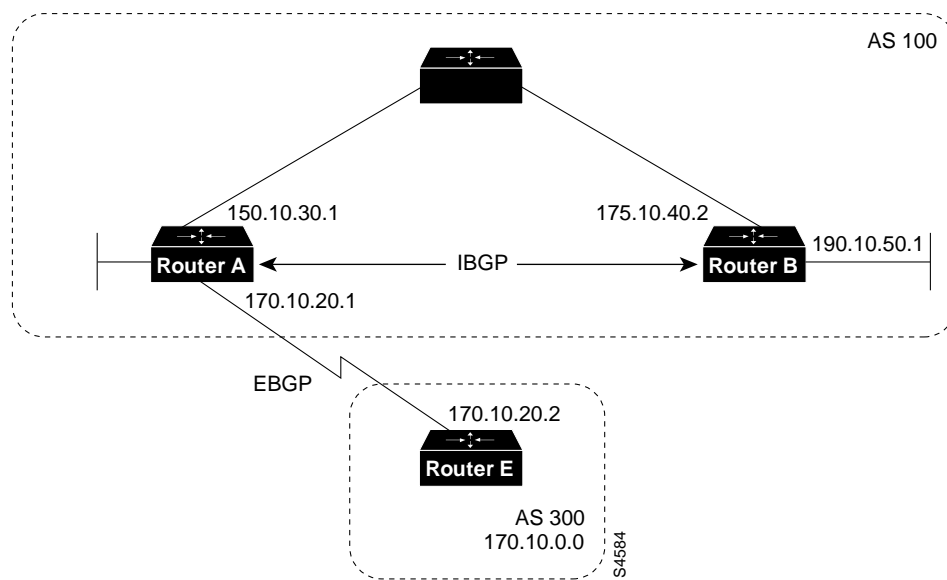
The following commands configure the routers shown in Figure 12-12:

```
!Router A
router bgp 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0
redistribute static
!
ip route 190.10.0.0 255.255.0.0 null 0
```

```
!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100
network 190.10.50.0

!Router E
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

Given these configurations, the following is true:

- From Router A, the route for reaching 170.10.0.0 has an AS_path of 300 and an origin attribute of IGP.

- From Router A, the route for reaching 190.10.50.0 has an empty AS_path (the route is in the same AS as Router A) and an origin attribute of IGP.

- From Router E, the route for reaching 150.10.0.0 has an AS_path of 100 and an origin attribute of IGP.

- From Router E, the route for reaching 190.10.0.0 has an AS_path of 100 and an origin attribute of Incomplete (because 190.10.0.0 is a redistributed route).

Next Hop Attribute

The BGP next hop attribute is the IP address of the next hop that is going to be used to reach a certain destination.

For EBGP, the next hop is usually the IP address of the neighbor specified by the **neighbor remote-as** router configuration command. (The exception is when the next hop is on a multiaccess media, in which case, the next hop could be the IP address of the router in the same subnet.) Consider the network shown in Figure 12-13.

Figure 12-13   Next Hop Attribute
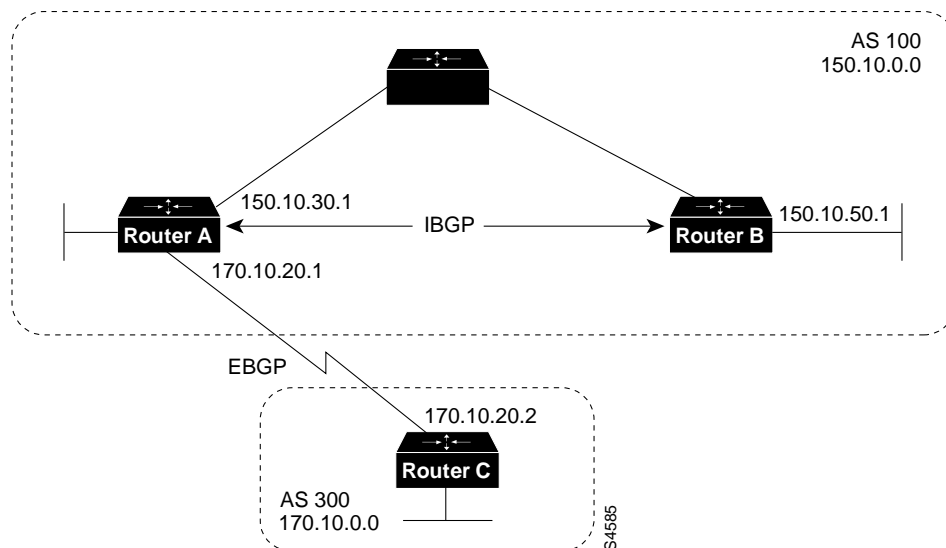
In Figure 12-13, Router C advertises network 170.10.0.0 to Router A with a next hop attribute of 170.10.20.2, and Router A advertises network 150.10.0.0 to Router C with a next hop attribute of 170.10.20.1.

BGP specifies that the next hop of EBGP-learned routes should be carried without modification into IBGP. Because of that rule, Router A advertises 170.10.0.0 to its IBGP peer (Router B) with a next hop attribute of 170.10.20.2. As a result, according to Router B, the next hop to reach 170.10.0.0 is 170.10.20.2, instead of 150.10.30.1. For that reason, the configuration must ensure that Router B can reach 170.10.20.2 via an IGP. Otherwise, Router B will drop packets destined for 170.10.0.0 because the next hop address is inaccessible.

For example, if Router B runs IGRP, Router A should run IGRP on network 170.10.0.0. You might want to make IGRP passive on the link to Router C so that only BGP updates are exchanged.

The following commands configure the routers shown in Figure 12-13:

```
!Router A
router bgp 100
neighbor 170.10.20.2 remote-as 300
neighbor 150.10.50.1 remote-as 100
network 150.10.0.0

!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100

!Router C
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

Note   Router C advertises 170.10.0.0 to Router A with a next hop attribute of 170.10.20.2, and Router A advertises 170.10.0.0 to Router B with a next hop attribute of 170.10.20.2. The next hop of EBGP-learned routes is passed to the IBGP neighbor.

Next Hop Attribute and Multiaccess Media

BGP might set the value of the next hop attribute differently on multiaccess media, such as Ethernet. Consider the network shown in Figure 12-14.

Figure 12-14   Next Hop Attribute and Multiaccess Media
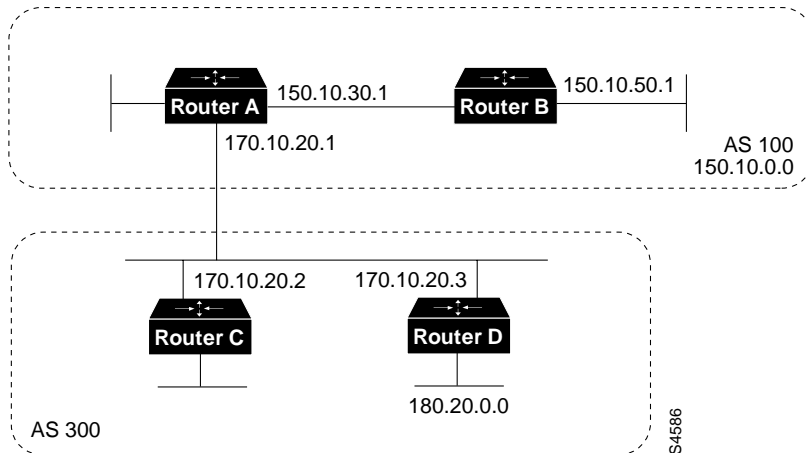
In Figure 12-14, Routers C and D in AS 300 are running OSPF. Router C is running BGP with Router A. Router C can reach network 180.20.0.0 via 170.10.20.3. When Router C sends a BGP update to Router A regarding 180.20.0.0, it sets the next hop attribute to 170.10.20.3, instead of its own IP address (170.10.20.2). This is because Routers A, B, and C are in the same subnet, and it makes more sense for Router A to use Router D as the next hop rather than taking an extra hop via Router C.

Next Hop Attribute and Nonbroadcast Media Access

In Figure 12-15, three networks are connected by a nonbroadcast media access (NBMA) cloud, such as Frame Relay.

Figure 12-15   Next Hop Attribute and Nonbroadcast Media Access
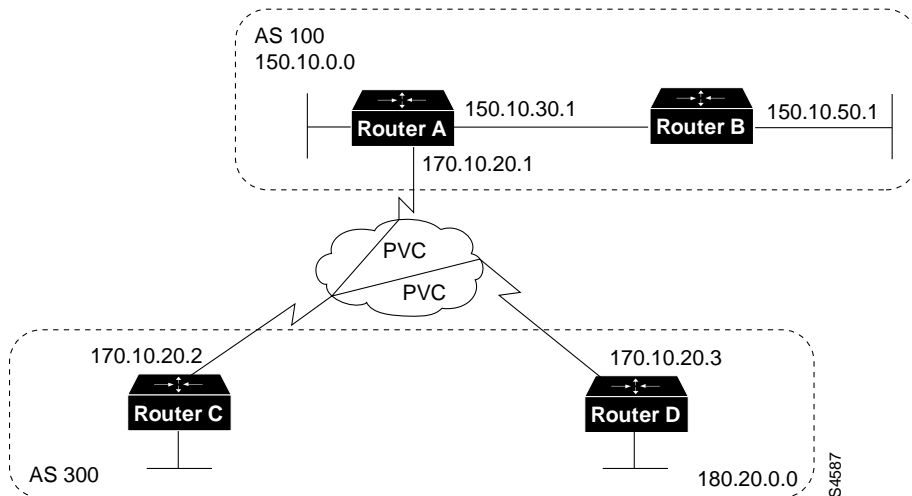
If Routers A, C, and D, use a common media such as Frame Relay (or any NBMA cloud), Router C advertises 180.20.0.0 to Router A with a next hop of 170.10.20.3, just as it would do if the common media were Ethernet. The problem is that Router A does not have a direct permanent virtual

connection (PVC) to Router D and cannot reach the next hop, so routing will fail. To remedy this situation, use the **neighbor next-hop-self** router configuration command, as shown in the following configuration for Router C:

```
!Router C
router bgp 300
neighbor 170.10.20.1 remote-as 100
neighbor 170.10.20.1 next-hop-self
```

The **neighbor next-hop-self** command causes Router C to advertise 180.20.0.0 with the next hop attribute set to 170.10.20.2.

Weight Attribute

The weight attribute is a special Cisco attribute that is used in the path selection process when there is more than one route to the same destination. The weight attribute is local to the router on which it is assigned, and it is not propagated in routing updates. By default, the weight attribute is 32768 for paths that the router originates and zero for other paths. Routes with a higher weight are preferred when there are multiple routes to the same destination.

Consider the network shown in Figure 12-16.

Figure 12-16   Weight Example
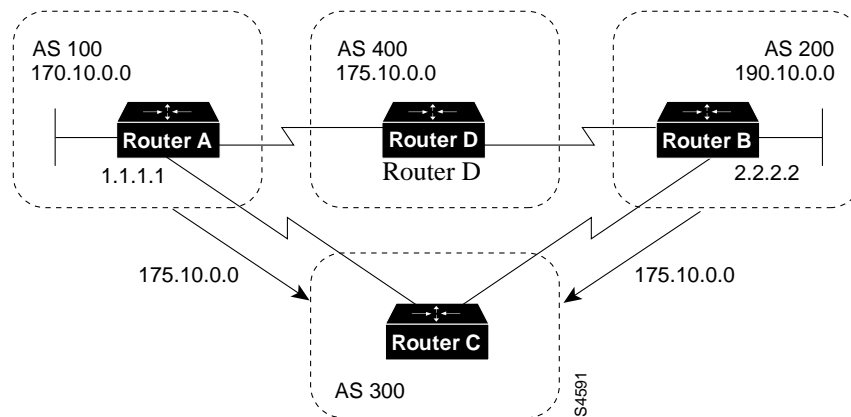
In Figure 12-16, Routers A and B learn about network 175.10.0.0 from AS 400, and each propagates the update to Router C. Router C has two routes for reaching 175.10.0.0 and has to decide which route to use. If, on Router C, you set the weight of the updates coming in from Router A to be higher than the updates coming in from Router B, Router C will use Router A as the next hop to reach network 175.10.0.0.

There are three ways to set the weight for updates coming in from Router A:

- Using an Access List to Set the Weight Attribute
- Using a Route Map to Set the Weight Attribute
- Using the neighbor weight Command to Set the Weight Attribute

Using an Access List to Set the Weight Attribute

The following commands on Router C use access lists and the value of the AS_path attribute to assign a weight to route updates:

```
!Router C
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 filter-list 5 weight 2000
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 filter-list 6 weight 1000
!
ip as-path access-list 5 permit ^100$
ip as-path access-list 6 permit ^200$
```

In this example, 2000 is assigned to the weight attribute of updates from the neighbor at IP address 1.1.1.1 that are permitted by access list 5. Access list 5 permits updates whose AS_path attribute starts with 100 (as specified by ^) and ends with 100 (as specified by $). (The ^ and $ symbols are used to form regular expressions. For a complete explanation of regular expressions, see the appendix on regular expressions in the Cisco Internetwork Operating System (Cisco IOS) software configuration guides and command references.

This example also assigns 1000 to the weight attribute of updates from the neighbor at IP address 2.2.2.2 that are permitted by access list 6. Access list 6 permits updates whose AS_path attribute starts with 200 and ends with 200.

In effect, this configuration assigns 2000 to the weight attribute of all route updates received from AS 100 and assigns 1000 to the weight attribute of all route updates from AS 200.

Using a Route Map to Set the Weight Attribute

The following commands on Router C use a route map to assign a weight to route updates:

```
!Router C
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-map SETWEIGHTIN in
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 route-map SETWEIGHTIN in
!
ip as-path access-list 5 permit ^100$
!
route-map SETWEIGHTIN permit 10
match as-path 5
set weight 2000
route-map SETWEIGHTIN permit 20
set weight 1000
```

This first instance of the SETWEIGHTIN route map assigns 2000 to any route update from AS 100, and the second instance of the SETWEIGHTIN route map assigns 1000 to route updates from any other AS.

Using the neighbor weight Command to Set the Weight Attribute

The following configuration for Router C uses the **neighbor weight** router configuration command:

```
!Router C
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 weight 2000
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 weight 1000
```

This configuration sets the weight of all route updates from AS 100 to 2000, and the weight of all route updates coming from AS 200 to 1000. The higher weight assigned to route updates from AS 100 causes Router C to send traffic through Router A.

Local Preference Attribute

When there are multiple paths to the same destination, the local preference attribute indicates the preferred path. The path with the higher preference is preferred (the default value of the local preference attribute is 100). Unlike the weight attribute, which is only relevant to the local router, the local preference attribute is part of the routing update and is exchanged among routers in the same AS.

The network shown in Figure 12-17 demonstrates the local preference attribute.

Figure 12-17   Local Preference
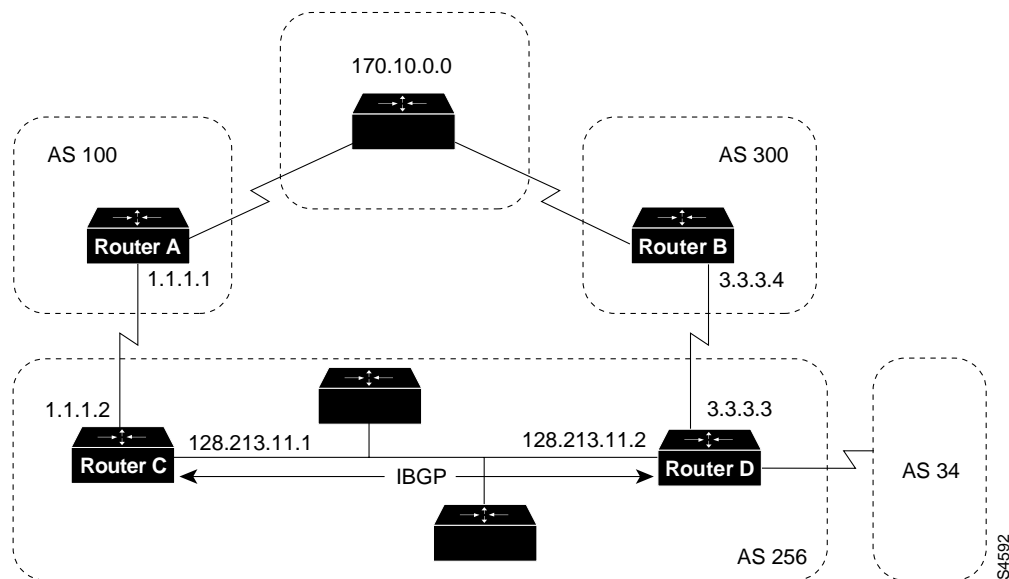
In Figure 12-17, AS 256 receives route updates for network 170.10.0.0 from AS 100 and AS 300. There are two ways to set local preference:

- Using the bgp default local-preference Command

- Using a Route Map to Set Local Preference

Using the bgp default local-preference Command

The following configurations use the **bgp default local-preference** router configuration command to set the local preference attribute on Routers C and D:

```
!Router C
router bgp 256
neighbor 1.1.1.1 remote-as 100
neighbor 128.213.11.2 remote-as 256
bgp default local-preference 150
```

```
!Router D
router bgp 256
neighbor 3.3.3.4 remote-as 300
neighbor 128.213.11.1 remote-as 256
bgp default local-preference 200
```

The configuration for Router C causes it to set the local preference of all updates from AS 300 to 150, and the configuration for Router D causes it to set the local preference for all updates from AS 100 to 200. Because local preference is exchanged within the AS, both Routers C and D determine that updates regarding network 170.10.0.0 have a higher local preference when they come from AS 300 than when they come from AS 100. As a result, all traffic in AS 256 destined for network 170.10.0.0 is sent to Router D as the exit point.

Using a Route Map to Set Local Preference

Route maps provide more flexibility than the **bgp default local-preference** router configuration command. When the **bgp default local-preference** command is used on Router D in Figure 12-17, the local preference attribute of all updates received by Router D will be set to 200, including updates from AS 34.

The following configuration uses a route map to set the local preference attribute on Router D specifically for updates regarding AS 300:

```
!Router D
router bgp 256
neighbor 3.3.3.4 remote-as 300
route-map SETLOCALIN in
neighbor 128.213.11.1 remote-as 256
!
ip as-path 7 permit ^300$
route-map SETLOCALIN permit 10
match as-path 7
set local-preference 200
!
route-map SETLOCALIN permit 20
```

With this configuration, the local preference attribute of any update coming from AS 300 is set to 200. Instance 20 of the SETLOCALIN route map accepts all other routes.

Multi-Exit Discriminator Attribute

The multi-exit discriminator (MED) attribute is a hint to external neighbors about the preferred path into an AS when there are multiple entry points into the AS. A lower MED value is preferred over a higher MED value. The default value of the MED attribute is 0.

Note In BGP Version 3, MED is known as Inter-AS_Metric.

Unlike local preference, the MED attribute is exchanged between ASs, but a MED attribute that comes into an AS does not leave the AS. When an update enters the AS with a certain MED value, that value is used for decision making within the AS. When BGP sends that update to another AS, the MED is reset to 0.

Unless otherwise specified, the router compares MED attributes for paths from external neighbors that are in the same AS. If you want MED attributes from neighbors in other ASs to be compared, you must configure the **bgp always-compare-med** command.

The network shown in Figure 12-18 demonstrates the use of the MED attribute.

Figure 12-18   MED Example
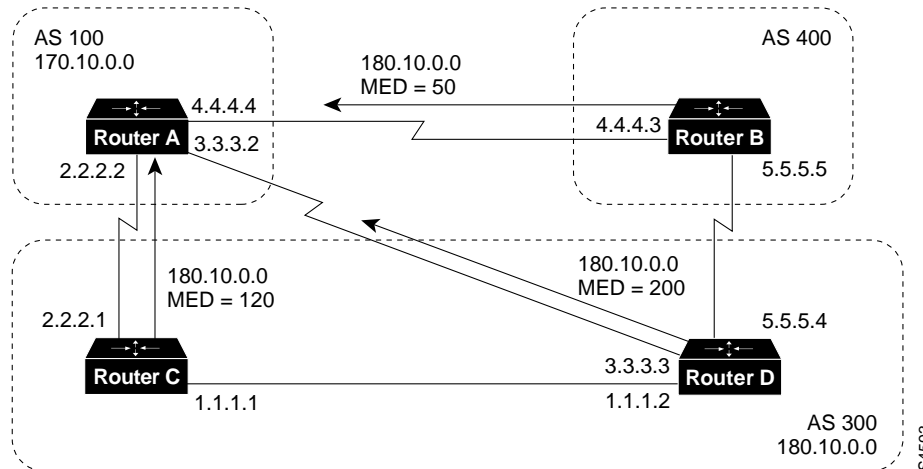
In Figure 12-18, AS 100 receives updates regarding network 180.10.0.0 from Routers B, C, and D. Routers C and D are in AS 300, and Router B is in AS 400.

The following commands configure Routers A, B, C, and D:

```
!Router A
router bgp 100
neighbor 2.2.2.1 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400

!Router B
router bgp 400
neighbor 4.4.4.4 remote-as 100
neighbor 4.4.4.4 route-map SETMEDOUT out
neighbor 5.5.5.4 remote-as 300
!
route-map SETMEDOUT permit 10
set metric 50

!Router C
router bgp 300
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map SETMEDOUT out
neighbor 5.5.5.5 remote-as 400
neighbor 1.1.1.2 remote-as 300
!
route-map SETMEDOUT permit 10
set metric 120

!Router D
router bgp 300
neighbor 3.3.3.2 remote-as 100
neighbor 3.3.3.2 route map SETMEDOUT out
neighbor 1.1.1.1 remote-as 300
route-map SETMEDOUT permit 10
set metric 200
```

By default, BGP compares the MED attributes of routes coming from neighbors in the same external AS (such as AS 300 in Figure 12-18). Router A can only compare the MED attribute coming from Router C (120) to the MED attribute coming from Router D (200) even though the update coming from Router B has the lowest MED value.

Router A will choose Router C as the best path for reaching network 180.10.0.0. To force Router A to include updates for network 180.10.0.0 from Router B in the comparison, use the **bgp always-compare-med** router configuration command, as in the following modified configuration for Router A:

```
!Router A
router bgp 100
neighbor 2.2.2.1 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400
bgp always-compare-med
```

Router A will choose Router B as the best next hop for reaching network 180.10.0.0 (assuming that all other attributes are the same).

You can also set the MED attribute when you configure the redistribution of routes into BGP. For example, on Router B you can inject the static route into BGP with a MED of 50 as in the following configuration:

```
!Router B
router bgp 400
redistribute static
default-metric 50
!
ip route 160.10.0.0 255.255.0.0 null 0
```

The preceding configuration causes Router B to send out updates for 160.10.0.0 with a MED attribute of 50.

Community Attribute

The community attribute provides a way of grouping destinations (called *communities*) to which routing decisions (such as acceptance, preference, and redistribution) can be applied.

Route maps are used to set the community attribute. A few predefined communities are listed in Table 12-1.

Table 12-1   Predefined Communities

Community	Meaning
no-export	Do not advertise this route to EBGP peers.
no-advertise	Do not advertise this route to any peer.
internet	Advertise this route to the internet community; all routers in the network belong to it.

The following route maps set the value of the community attribute:

```
route-map COMMUNITYMAP
match ip address 1
set community no-advertise
!
route-map SETCOMMUNITY
match as-path 1
set community 200 additive
```

If you specify the **additive** keyword, the specified community value is added to the existing value of the community attribute. Otherwise, the specified community value replaces any community value that was set previously.

To send the community attribute to a neighbor, you must use the **neighbor send-community** router configuration command, as in the following example:

```
router bgp 100
neighbor 3.3.3.3 remote-as 300
neighbor 3.3.3.3 send-community
neighbor 3.3.3.3 route-map setcommunity out
```

For examples of how the community attribute is used to filter updates, see the section "Community Filtering," later in this chapter.

Summary of the BGP Path Selection Process

BGP selects only one path as the best path. When the path is selected, BGP puts the selected path in its routing table and propagates the path to its neighbors. BGP uses the following criteria, in the order presented, to select a path for a destination:

1 If the path specifies a next hop that is inaccessible, drop the update.

2 Prefer the path with the largest weight.

3 If the weights are the same, prefer the path with the largest local preference.

4 If the local preferences are the same, prefer the path that was originated by BGP running on this router.

5 If no route was originated, prefer the route that has the shortest AS_path.

6 If all paths have the same AS_path length, prefer the path with the lowest origin type (where IGP is lower than EGP, and EGP is lower than Incomplete).

7 If the origin codes are the same, prefer the path with the lowest MED attribute.

8 If the paths have the same MED, prefer the external path over the internal path.

9 If the paths are still the same, prefer the path through the closest IGP neighbor.

10 Prefer the path with the lowest IP address, as specified by the BGP router ID.

Controlling the Flow of BGP Updates

This section describes techniques for controlling the flow of BGP updates. The techniques include the following:

- Administrative Distance
- BGP Filtering
- BGP Peer Groups
- CIDR and Aggregate Addresses
- Confederations
- Route Reflectors
- Route Flap Dampening

Administrative Distance

Normally, a route could be learned via more than one protocol. Administrative distance is used to discriminate between routes learned from more than one protocol. The route with the lowest administrative distance is installed in the IP routing table. By default, BGP uses the administrative distances shown in Table 12-2.

Table 12-2   BGP Default Distances

Distance	Default Value	Function
External	20	Applied to routes learned from EBGP
Internal	200	Applied to routes learned from IBGP
Local	200	Applied to routes originated by the router

Note   Distance does not influence the BGP path selection algorithm, but it does influence whether BGP-learned routes are installed in the IP routing table.

Usually when a route is learned via EBGP, it is installed in the IP routing table because of its distance (20). Sometimes, however, two ASs have an IGP-learned backdoor route and an EBGP-learned route. Their policy might be to use the IGP-learned path as the preferred path and to use the EBGP-learned path when the IGP path is down. The network in Figure 12-19 shows this situation.

Figure 12-19   Back Door Example
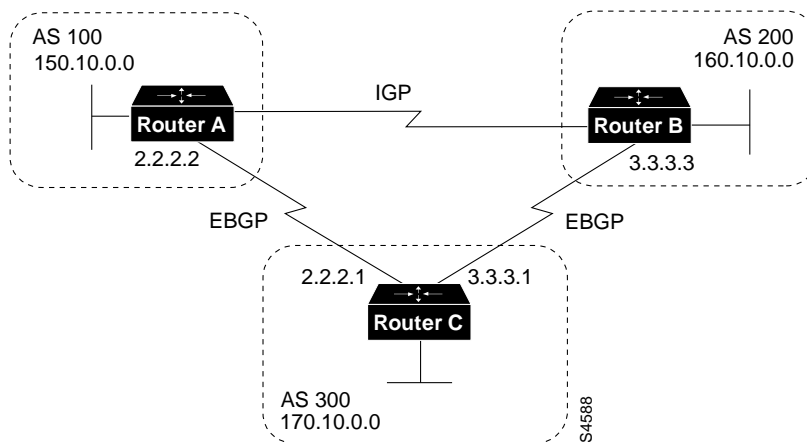
In Figure 12-19, Routers A and C are running EBGP, as are Routers B and C. Routers A and B are running an IGP (such as RIP, IGRP, Enhanced IGRP, or OSPF). The default distances for RIP, IGRP, Enhanced IGRP, and OSPF are 120, 100, 90, and 110, respectively. All of these default distances are higher than the default distance of EBGP (which is 20). Usually, the route with the lowest distance is preferred.

Router A receives updates about 160.10.0.0 from two routing protocols: EBGP and an IGP. Because the default distance for EBGP is lower than the default distance of the IGP, Router A will choose the EBGP-learned route from Router C. If you want Router A to learn about 160.10.0.0 from Router B (IGP), you could use one of the following techniques:

- Change the external distance of EBGP. (*Not recommended because the distance will affect all updates, which might lead to undesirable behavior when multiple routing protocols interact with one another.*)

- Change the distance of the IGP. (*Not recommended because the distance will affect all updates, which might lead to undesirable behavior when multiple routing protocols interact with one another.*)

- Establish a BGP back door. (*Recommended*)

To establish a BGP back door, use the **network backdoor** router configuration command.

The following commands configure Router A in Figure 12-19:

```
!Router A
router eigrp 10
network 150.10.0.0
router bgp 100
neighbor 2.2.2.1 remote-as 300
network 160.10.0.0 backdoor
```

With the **network backdoor** command, Router A treats the EBGP-learned route as local and installs it in the IP routing table with a distance of 200. The network is also learned via Enhanced IGRP (with a distance of 90), so the Enhanced IGRP route is successfully installed in the IP routing table and is used to forward traffic. If the Enhanced IGRP-learned route goes down, the EBGP-learned route will be installed in the IP routing table and used to forward traffic.

Note   Although BGP treats network 160.10.0.0 as a local entry, it does not advertise network 160.10.0.0 as it normally would advertise a local entry.

BGP Filtering

You can control the sending and receiving of updates by using the following filtering methods:

- Prefix Filtering
- AS_path Filtering
- Route Map Filtering
- Community Filtering

Each method can be used to achieve the same result—the choice of method depends on the specific network configuration.

Prefix Filtering

To restrict the routing information that the router learns or advertises, you can filter based on routing updates to or from a particular neighbor. The filter consists of an access list that is applied to updates to or from a neighbor.

The network shown in Figure 12-20 demonstrates the usefulness of prefix filtering.

Figure 12-20   Route Filtering
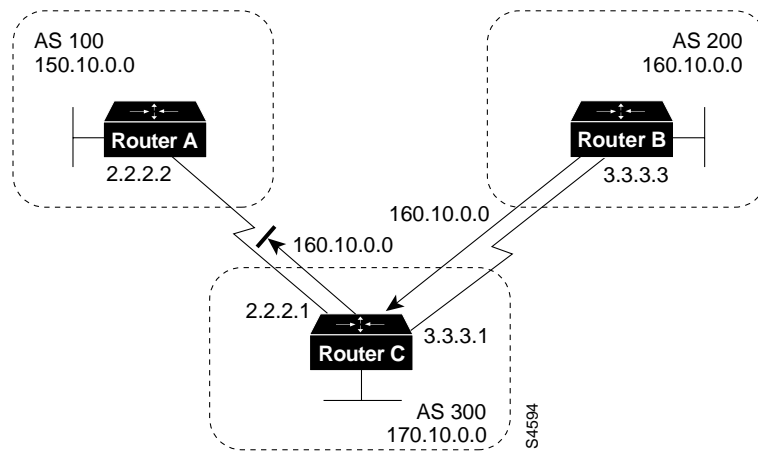
In Figure 12-20, Router B is originating network 160.10.0.0 and sending it to Router C. If you want to prevent Router C from propagating updates for network 160.10.0.0 to AS 100, you can apply an access list to filter those updates when Router C exchanges updates with Router A, as demonstrated by the following configuration for Router C:

```
!Router C
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 distribute-list 1 out
!
access-list 1 deny 160.10.0.0 0.0.255.255
access-list 1 permit 0.0.0.0 255.255.255.255
```

In the preceding configuration, the combination of the **neighbor distribute-list** router configuration command and access list 1 prevents Router C from propagating routes for network 160.10.0.0 when it sends routing updates to neighbor 2.2.2.2 (Router A).

Using access lists to filter supernets is a bit trickier. Assume, for example, that Router B in Figure 12-20 has different subnets of 160.10.x.x, and you want to advertise 160.0.0.0/8 only. The following access list would permit 160.0.0.0/8, 160.0.0.0/9, and so on:

```
access-list 1 permit 160.0.0.0 0.255.255.255
```

To restrict the update to 160.0.0.0/8 only, you have to use an extended access list, such as the following:
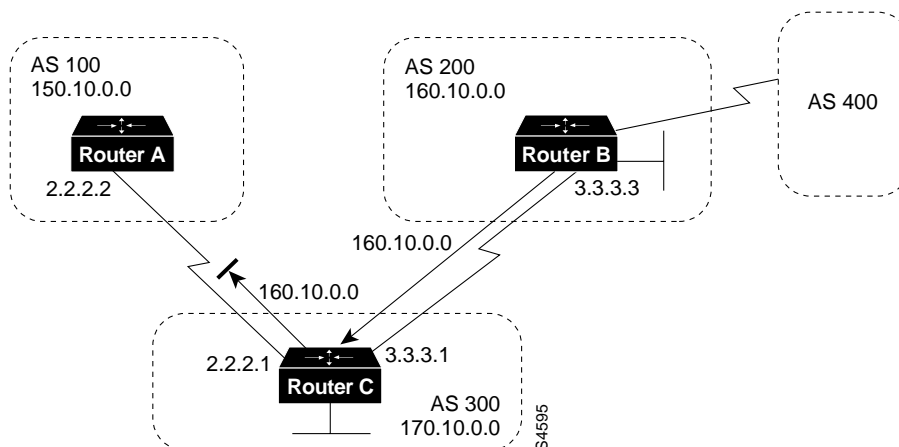
```
access-list 101 permit ip 160.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
```

AS_path Filtering

You can specify an access list on both incoming and outgoing updates based on the value of the AS_path attribute.

The network shown in Figure 12-21 demonstrates the usefulness of AS_path filters.

Figure 12-21   AS_path Filtering


```
!Router C
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 filter-list 1 out
!
ip as-path access-list 1 deny ^200$
ip as-path access-list 1 permit .*
```

In this example, access list 1 denies any update whose AS_path attribute starts with 200 (as specified by ^) and ends with 200 (as specified by $). Because Router B sends updates about 160.10.0.0 whose AS_path attributes start with 200 and end with 200, such updates will match the access list and will be denied. By specifying that the update must also end with 200, the access list permits updates from AS 400 (whose AS_path attribute is 200, 400). If the access list specified ^200 as the regular expression, updates from AS 400 would be denied.

In the second access-list statement, the period (.) symbol means any character, and the asterisk (*) symbol means a repetition of that character. Together, .* matches any value of the AS_path attribute, which in effect permits any update that has not been denied by the previous access-list statement.

If you want to verify that your regular expressions work as intended, use the following EXEC command:

show ip bgp regexp *regular-expression*

The router displays all of the paths that match the specified regular expression.

Route Map Filtering

The **neighbor route-map** router configuration command can be used to apply a route map to incoming and outgoing routes.

Note   The **neighbor route-map** command has no effect on incoming updates when matching is based on IP address.

The network shown in Figure 12-22 demonstrates using route maps to filter BGP updates.

Figure 12-22   BGP Route Map Filtering
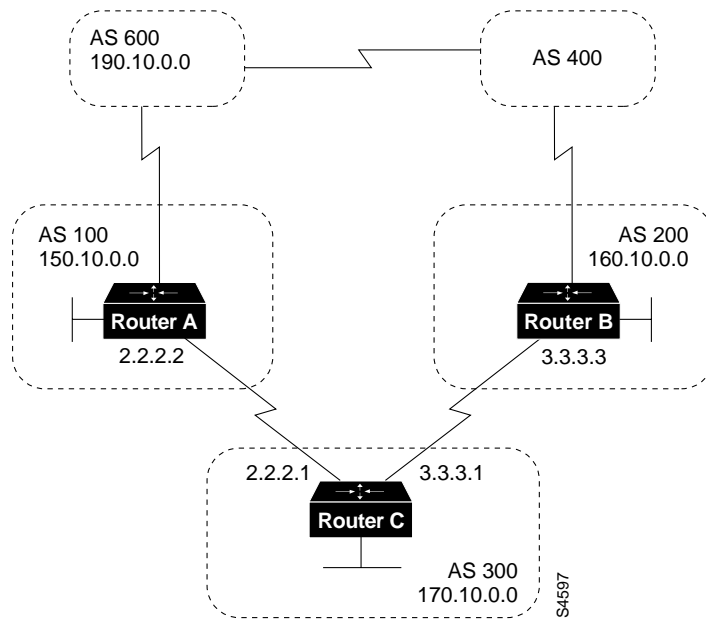
Assume that in Figure 12-22, you want Router C to learn about networks that are local to AS 200 only. (That is, you do not want Router C to learn about AS 100, AS 400, or AS 600 from AS 200.) Also, on those routes that Router C accepts from AS 200, you want the weight attribute to be set to 20. The following configuration for Router C accomplishes this goal:

```
!Router C
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map STAMP in
!
route-map STAMP permit 10
match as-path 1
set weight 20
!
ip as-path access-list 1 permit ^200$
```

In the preceding configuration, access list 1 permits any update whose AS_path attribute begins with 200 and ends with 200 (that is, access list 1 permits updates that originate in AS 200). The weight attribute of the permitted updates is set to 20. All other updates are denied and dropped.

Assume that in Figure 12-22, you want Router C to do the following:

- Accept updates that originate from AS 200 and change their weight attribute to 20.

- Deny updates that contain AS 400.

- Accept any other updates and change their weight attribute to 10.

The following configuration for Router C accomplishes this goal:

```
!Router C
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map STAMP in
```

```
route-map STAMP permit 10
match as-path 1
set weight 20
!
route-map STAMP permit 20
match as-path 2
!
route-map STAMP permit 30
set weight 10
!
ip as-path access-list 1 permit ^200$
ip as-path access-list 2 deny _400_
```

In the preceding configuration, access list 1 permits any update whose AS_path attribute begins with 200 and ends with 200 (that is, access list 1 permits updates that originate in AS 200). The weight attribute of the permitted updates is set to 20. Access list 2 denies updates whose AS_path attribute contains 400. All other updates will have a weight of 10 (by means of instance 30 of the STAMP route map) and will be permitted.

Suppose that in Figure 12-22 Router C advertises its own network (170.10.0.0) to AS 100 and AS 200. When updates about network 170.10.0.0 arrive in AS 600, the routers in AS 600 will have network reachability information via two routes: via AS 100 with an AS_path attribute of (100, 300) and via AS 400 with an AS_path attribute of (400, 200, 300). Assuming that the values of all other attributes are the same, the routers in AS 600 will pick the shortest AS_path attribute: the route through AS 100.

If you want to use the configuration of Router C to influence the choice of paths in AS 600, you can do so by prepending extra AS numbers to the AS_path attribute for routes that Router C advertises to AS 100. A common practice is to repeat the AS number, as in the following configuration:

```
!Router C
router bgp 300
network 170.10.0.0
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map SETPATH out
!
route-map SETPATH permit 10
set as-path prepend 300 300
```

The **set as-path** route map configuration command with the **prepend** keyword causes Router C to prepend 300 twice to the value of the AS_path attribute before it sends updates to the neighbor at IP address 2.2.2.2 (Router A). As a result, the AS_path attribute of updates for network 170.10.0.0 that AS 600 receives via AS 100 will be 100, 300, 300, 300, which is longer than the value of the AS_path attribute of updates for network 170.10.0.0 that AS 600 receives via AS 400 (400, 200, 300). AS 600 will choose (400, 200, 300) as the better path.

Community Filtering

The network shown in Figure 12-23 demonstrates the usefulness of community filters.

Figure 12-23   Community Filtering
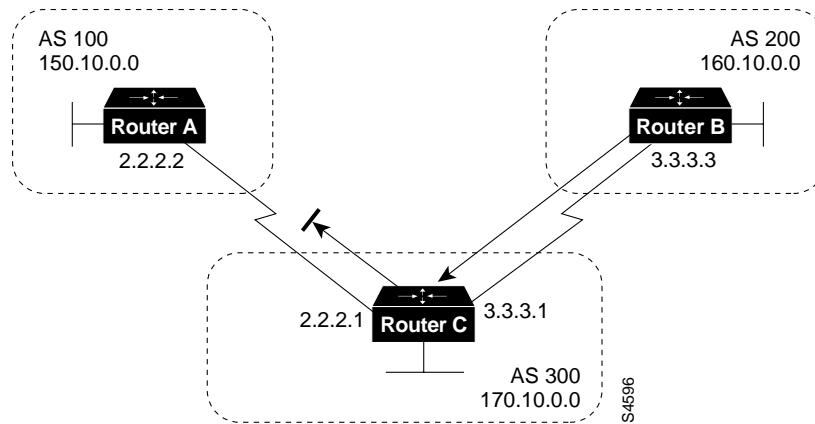
Assume that you do not want Router C to propagate routes learned from Router B to Router A. You can do this by setting the community attribute on updates that Router B sends to Router C, as in the following configuration for Router B:

```
!Router B
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300
neighbor 3.3.3.1 send-community
neighbor 3.3.3.1 route-map SETCOMMUNITY out
!
route-map SETCOMMUNITY permit 10
match ip address 1
set community no-export
!
route-map SETCOMMUNITY permit 20
!
access list 1 permit 0.0.0.0 255.255.255.255
```

For routes that are sent to the neighbor at IP address 3.3.3.1 (Router C), Router B applies the route map named SETCOMMUNITY. The SETCOMMUNITY route map sets the community attribute of any update (by means of access list 1) destined for 3.3.3.1 to no-export. The **neighbor send-community** router configuration command is required to include the community attribute in updates sent to the neighbor at IP address 3.3.3.1.

When Router C receives the updates from Router B, it does not propagate them to Router A because the value of the community attribute is no-export.

Another way to filter updates based on the value of the community attribute is to use the **ip community-list** global configuration command. Assume that Router B has been configured as follows:

```
!Router B
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300
neighbor 3.3.3.1 send-community
neighbor 3.3.3.1 route-map SETCOMMUNITY out
!
route-map SETCOMMUNITY permit 10
match ip address 2
set community 100 200 additive
```

```
route-map SETCOMMUNITY permit 20
!
access list 2 permit 0.0.0.0 255.255.255.255
```

In the preceding configuration, Router B adds 100 and 200 to the community value of any update destined for the neighbor at IP address 3.3.3.1. To configure Router C to use the **ip community-list** global configuration command to set the value of the weight attribute based on whether the community attribute contains 100 or 200, use the following configuration:

```
!Router C
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map check-community in
!
route-map check-community permit 10
match community 1
set weight 20
!
route-map check-community permit 20
match community 2 exact
set weight 10
!
route-map check-community permit 30
match community 3
!
ip community-list 1 permit 100
ip community-list 2 permit 200
ip community-list 3 permit internet
```

In the preceding configuration, any route that has 100 in its community attribute matches community list 1 and has its weight set to 20. Any route whose community attribute is only 200 (by virtue of the **exact** keyword) matches community list 2 and has its weight set to 10. In the last community list (list 3) the use of the **internet** keyword permits all other updates without changing the value of an attribute. (The **internet** keyword specifies all routes because all routes are members of the internet community.)

BGP Peer Groups

A BGP peer group is a group of BGP neighbors that share the same update policies. Update policies are usually set by route maps, distribution lists, and filter lists. Instead of defining the same policies for each individual neighbor, you define a peer group name and assign policies to the peer group.

Members of a peer group inherit all of the configuration options of the peer group. Peer group members can also be configured to override configuration options if the options do not affect outgoing updates. That is, you can only override options that are set for incoming updates.

The use of BGP peer groups is demonstrated by the network shown in Figure 12-24.

Figure 12-24   BGP Peer Groups
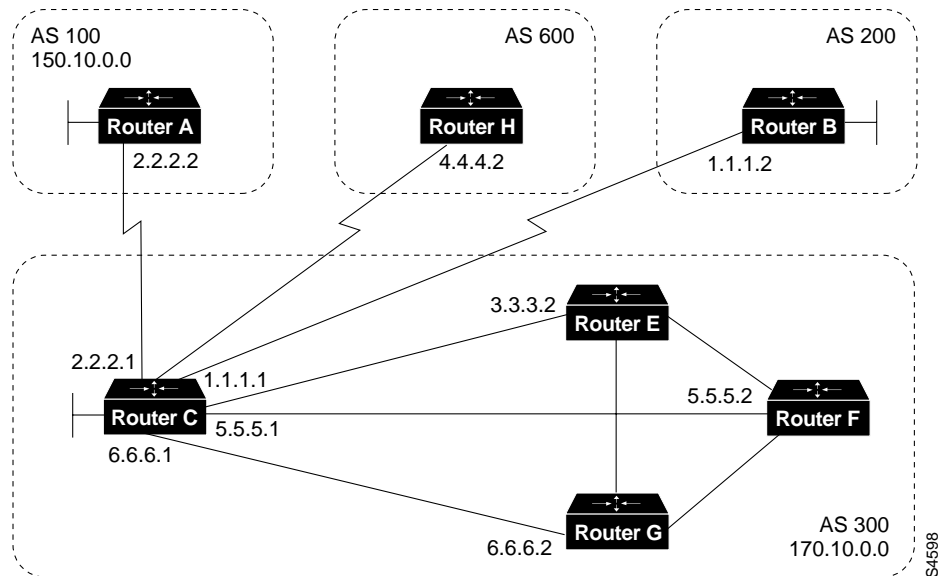
The following commands configure a BGP peer group named INTERNALMAP on Router C and apply it to the other routers in AS 300:

```
!Router C
router bgp 300
neighbor INTERNALMAP peer-group
neighbor INTERNALMAP remote-as 300
neighbor INTERNALMAP route-map INTERNAL out
neighbor INTERNALMAP filter-list 1 out
neighbor INTERNALMAP filter-list 2 in
neighbor 5.5.5.2 peer-group INTERNALMAP
neighbor 6.6.6.2 peer-group INTERNALMAP
neighbor 3.3.3.2 peer-group INTERNALMAP
neighbor 3.3.3.2 filter-list 3 in
```

The preceding configuration defines the following policies for the internalmap peer group:

* A route map named INTERNAL

* A filter list for outgoing updates (filter list 1)

* A filter list for incoming updates (filter list 2)

The configuration applies the peer group to all internal neighbors—Routers E, F, and G. The configuration also defines a filter list for incoming updates from the neighbor at IP address 3.3.3.2 (Router E). This filter list can only be used to override options that affect incoming updates.

The following commands configure a BGP peer group named EXTERNALMAP on Router C and apply it to routers in AS 100, 200, and 600:

```
!Router C
router bgp 300
neighbor EXTERNALMAP peer-group
neighbor EXTERNALMAP route-map SETMED
neighbor EXTERNALMAP filter-list 1 out
neighbor EXTERNALMAP filter-list 2 in
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 peer-group EXTERNALMAP
neighbor 4.4.4.2 remote-as 600
```

```
neighbor 4.4.4.2 peer-group EXTERNALMAP
neighbor 1.1.1.2 remote-as 200
neighbor 1.1.1.2 peer-group EXTERNALMAP
neighbor 1.1.1.2 filter-list 3 in
```

In the preceding configuration, the **neighbor remote-as** router configuration commands are placed outside of the **neighbor peer-group** router configuration commands because different external ASs have to be defined. Also note that this configuration defines filter list 3, which can be used to override configuration options for incoming updates from the neighbor at IP address 1.1.1.2 (Router B).
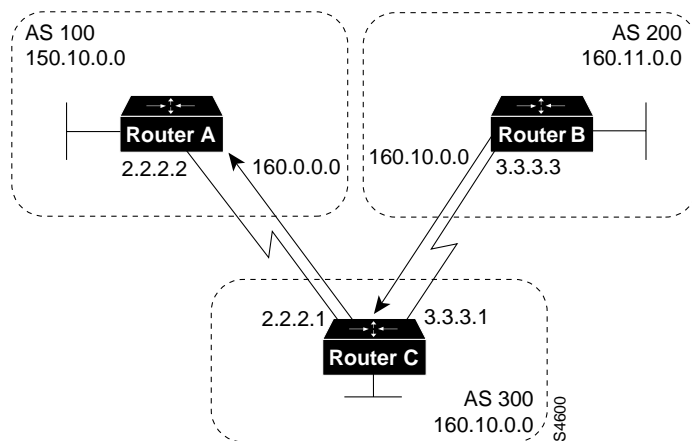
CIDR and Aggregate Addresses

BGP4 supports classless interdomain routing (CIDR), which is a major improvement over BGP3. (CIDR is also known as *supernetting*.) CIDR is a new way of looking at IP addresses that eliminates the concept of classes (Class A, Class B, and so on). For example, network 192.213.0.0, which is an illegal Class C network number, is a legal supernet when it is represented in CIDR notation as 192.213.0.0/16. The /16 indicates that the subnet mask consists of 16 bits (counting from the left). Therefore, 192.213.0.0/16 is similar to 192.213.0.0 255.255.0.0.

CIDR makes it easy to aggregate routes. Aggregation is the process of combining several different routes in such a way that a single route can be advertised, which minimizes the size of routing tables.

Consider the network shown in Figure 12-25.

Figure 12-25   Aggregation

In Figure 12-25, Router B in AS 200 is originating network 160.11.0.0 and advertising it to Router C in AS 300. To configure Router C to propagate the aggregate address 160.0.0.0 to Router A, use the following commands:

```
!Router C
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
network 160.10.0.0
aggregate-address 160.0.0.0 255.0.0.0
```

The **aggregate-address** router configuration command advertises the prefix route (in this case, 160.0.0.0/8) and all of the more specific routes.

> **Note** A router cannot aggregate an address if it does not have a more specific route of that address in the BGP routing table. The more specific route can be injected in the BGP routing table by incoming updates from other ASs, can be redistributed from an IGP, or can be established by the **network** router configuration command.

If you want Router C to propagate the prefix route only, and you do not want it to propagate a more specific route, use the following command:

```
aggregate-address 160.0.0.0 255.0.0.0 summary-only
```

This command propagates the prefix (160.0.0.0/8) and suppresses any more specific routes that the router may have in its BGP routing table.

> **Note** If you use the **network** command to advertise a network, the entry for that network is always injected into BGP updates, even if you specify the **summary-only** keyword with the **aggregate-address** router configuration command.

If you want to suppress specific routes when aggregating routes, you can define a route map and apply it to the aggregate. If, for example, you want Router C in Figure 12-25 to aggregate 160.0.0.0 and suppress the specific route 160.20.0.0, but propagate route 160.10.0.0, use the following commands:

```
!Router C
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
network 160.10.0.0
aggregate-address 160.0.0.0 255.0.0.0 suppress-map CHECK
!
route-map CHECK permit 10
match ip address 1
!
access-list 1 deny 160.20.0.0 0.0.255.255
access-list 1 permit 0.0.0.0 255.255.255.255
```

If you want the router to set the value of an attribute when it propagates the aggregate route, use an attribute map, as demonstrated by the following commands:

```
route-map SETORIGIN permit 10
set origin igp
!
aggregate-address 160.0.0.0 255.0.0.0 attribute-map SETORIGIN
```

Aggregation and Static Routes

The network shown in Figure 12-26 demonstrates how static routes can be used to generate aggregates.

Figure 12-26   CIDR Aggregation Example
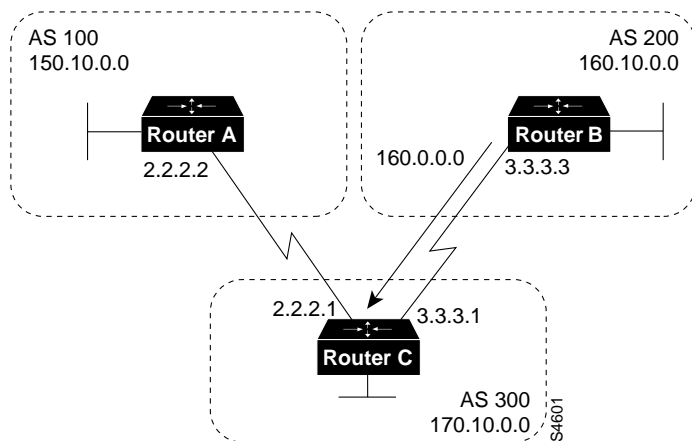
In Figure 12-26, you want Router B to advertise the prefix 160.0.0.0 and suppress all of the more specific routes.

The following configuration for Router B redistributes a static aggregate route into BGP:

```
!Router B
router bgp 200
neighbor 3.3.3.1 remote-as 300
redistribute static
!
ip route 160.0.0.0 255.0.0.0 null 0
```

As a result of this configuration, Router B advertises the aggregate with an origin attribute whose value is Incomplete.

Using the **network** router command instead of the **redistribute** command, as in the following configuration, has the same effect as the preceding configuration except that the origin attribute of updates for network 160.0.0.0 will be set to IGP instead of Incomplete.

```
!Router B
router bgp 200
network 160.0.0.0 mask 255.0.0.0
neighbor 3.3.3.1 remote-as 300
!
ip route 160.0.0.0 255.0.0.0 null 0
```

Note   The use of static routes (as shown in these two examples) is the preferred method of injecting an aggregate route because using static routes avoids unnecessary route flaps.

Aggregation and AS-SET

When aggregates are generated from more specific routes, the AS_path attributes of the more specific routes are combined to form a set called the AS-SET. This set is useful for preventing routing information loops.

The network shown in Figure 12-27 demonstrates the use of AS-SET when aggregating addresses.

Figure 12-27   CIDR Aggregation Example with AS-SET
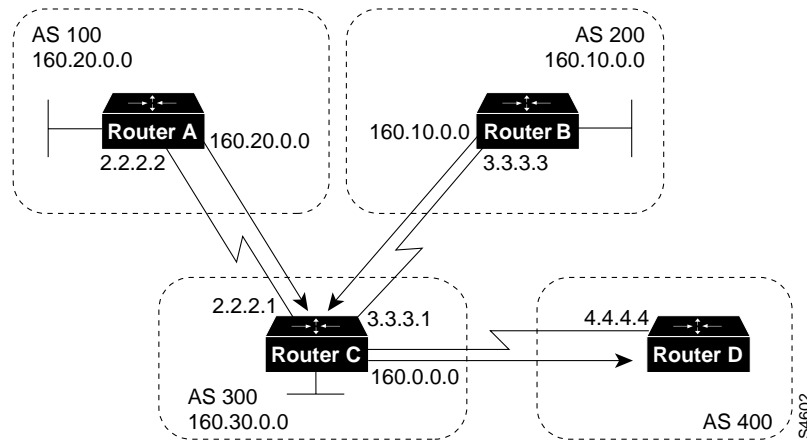
In Figure 12-27, Router C is receiving updates about network 160.20.0.0 from Router A and is receiving updates about network 160.10.0.0 from Router B. If Router C aggregates network 160.0.0.0/8 and sends updates for it to Router D, the AS_path attribute of those updates will indicate that AS 300 is the origin of network 160.0.0.0. If Router D has another route to AS 100, the updates from AS 300 may cause a routing loop. To prevent this problem, use the **aggregate-address** router configuration command with the **as-set** keyword, as in the following configuration for Router C:

```
!Router C
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 4.4.4.4 remote-as 400
aggregate-address 160.0.0.0 255.0.0.0 as-set
```

The **as-set** keyword causes Router C to generate updates for network 160.0.0.0/8 that include information indicating that network 160.0.0.0 belongs to a set (in this case, the set of 100 and 200).

Confederations

A confederation is a technique for reducing the IBGP mesh inside the AS. Consider the network shown in Figure 12-28.

Figure 12-28   Confederations
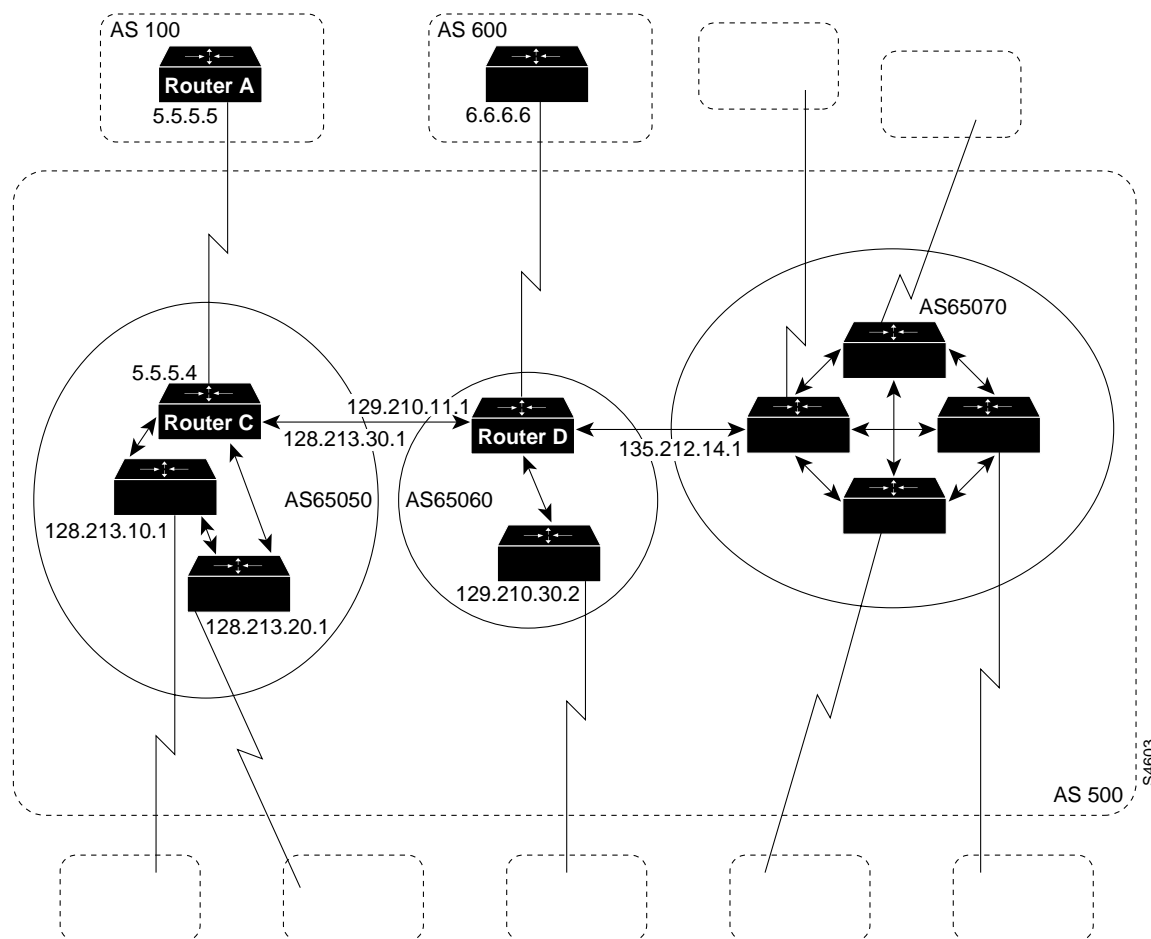
In Figure 12-28, AS 500 consists of nine BGP speakers (although there might be other routers that are not configured for BGP). Without confederations, BGP would require that the routers in AS 500 be fully meshed. That is, each router would need to run IBGP with each of the other eight routers, and each router would need to connect to an external AS and run EBGP, for a total of nine peers for each router.

Confederations reduce the number of peers within the AS, as shown in Figure 12-28. You use confederations to divide the AS into multiple mini-ASs and assign the mini-ASs to a confederation. Each mini-AS is fully meshed, and IBGP is run among its members. Each mini-AS has a connection to the other mini-ASs within the confederation. Even though the mini-ASs have EBGP peers to ASs within the confederation, they exchange routing updates as if they were using IBGP—that is, the next hop, MED, and local preference information is preserved. To the outside world, the confederation looks like a single AS.

The following commands configure Router C:

```
!Router C
router bgp 65050
bgp confederation identifier 500
bgp confederation peers 65060 65070
neighbor 128.213.10.1 remote-as 65050
neighbor 128.213.20.1 remote-as 65050
neighbor 128.210.11.1 remote-as 65060
```

```
neighbor 135.212.14.1 remote-as 65070
neighbor 5.5.5.5 remote-as 100
```

The **router bgp** global configuration command specifies that Router C belongs to AS 50.

The **bgp confederation identifier** router configuration command specifies that Router C belongs to confederation 500.

The first two **neighbor remote-as** router configuration commands establish IBGP connections to the other two routers within AS 65050. The second two **neighbor remote-as** commands establish BGP connections with confederation peers 65060 and 65070. The last **neighbor remote-as** command establishes an EBGP connection with external AS 100.

The following commands configure Router D:

```
!Router D
router bgp 65060
bgp confederation identifier 500
bgp confederation peers 65050 65070
neighbor 129.210.30.2 remote-as 65060
neighbor 128.213.30.1 remote-as 65050
neighbor 135.212.14.1 remote-as 65070
neighbor 6.6.6.6 remote-as 600
```

The **router bgp** global configuration command specifies that Router D belongs to AS 65060.

The **bgp confederation identifier** router configuration command specifies that Router D belongs to confederation 500.

The first **neighbor remote-as** router configuration command establishes an IBGP connection to the other router within AS 65060. The second two **neighbor remote-as** commands establish BGP connections with confederation peers 65050 and 65070. The last **neighbor remote-as** command establishes an EBGP connection with AS 600.

The following commands configure Router A:

```
!Router A
router bgp 100
neighbor 5.5.5.4 remote-as 500
```

The **neighbor remote-as** command establishes an EBGP connection with Router C. Router A is unaware of AS 65050, AS 65060, or AS 65070. Router A only has knowledge of AS 500.

Route Reflectors

Route reflectors are another solution for the explosion of IBGP peering within an AS. As described earlier in the section "Synchronization," a BGP speaker does not advertise a route learned from another IBGP speaker to a third IBGP speaker. Route reflectors ease this limitation and allow a router to advertise (reflect) IBGP-learned routes to other IBGP speakers, thereby reducing the number of IBGP peers within an AS.

The network shown in Figure 12-29 demonstrates how route reflectors work.

Figure 12-29   Simple Route Reflector Example
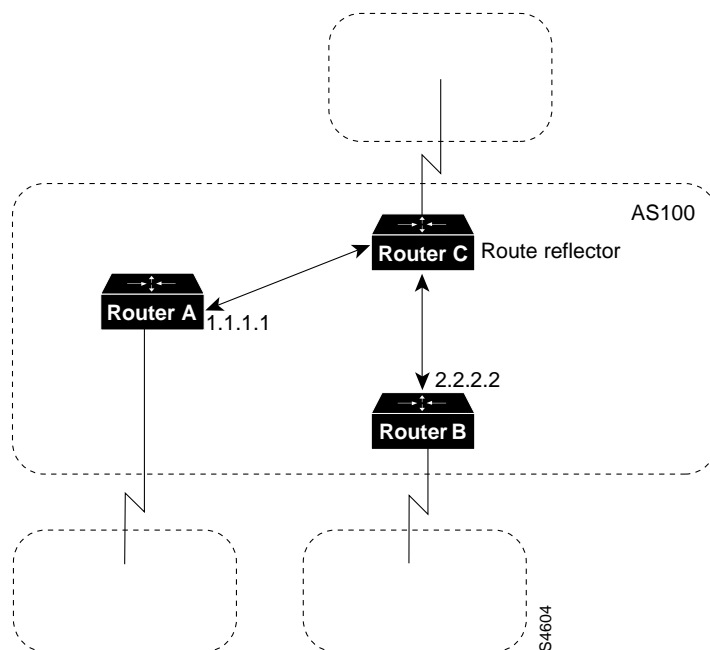
Without a route reflector, the network shown in Figure 12-29 would require a full IBGP mesh (that is, Router A would have to be a peer of Router B). If Router C is configured as a route reflector, IBGP peering between Routers A and B is not required because Router C will reflect updates from Router A to Router B and from Router B to Router A. To configure Router C as a route reflector, use the following commands:

```
!Router C
router bgp 100
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
```

The router whose configuration includes **neighbor route-reflector-client** router configuration commands is the route reflector. The routers identified by the **neighbor route-reflector-client** commands are clients of the route reflector. When considered as a whole, the route reflector and its clients are called a *cluster*. Other IBGP peers of the route reflector that are not clients are called *nonclients*.

An AS can have more than one route reflector. When an AS has more than one route reflector, each route reflector treats other route reflectors as normal IBGP speakers. There can be more than one route reflector in a cluster, and there can be more than one cluster in an AS.

In the advanced configuration shown in Figure 12-30, the AS is divided into multiple clusters, with each cluster having one route reflector. Each route reflector is configured as a nonclient peer of each other route reflector in a fully meshed topology.

Note   Route reflector clients should not establish peer relationships with IBGP speakers outside of their cluster.

Figure 12-30   Advanced Route Reflectors Example
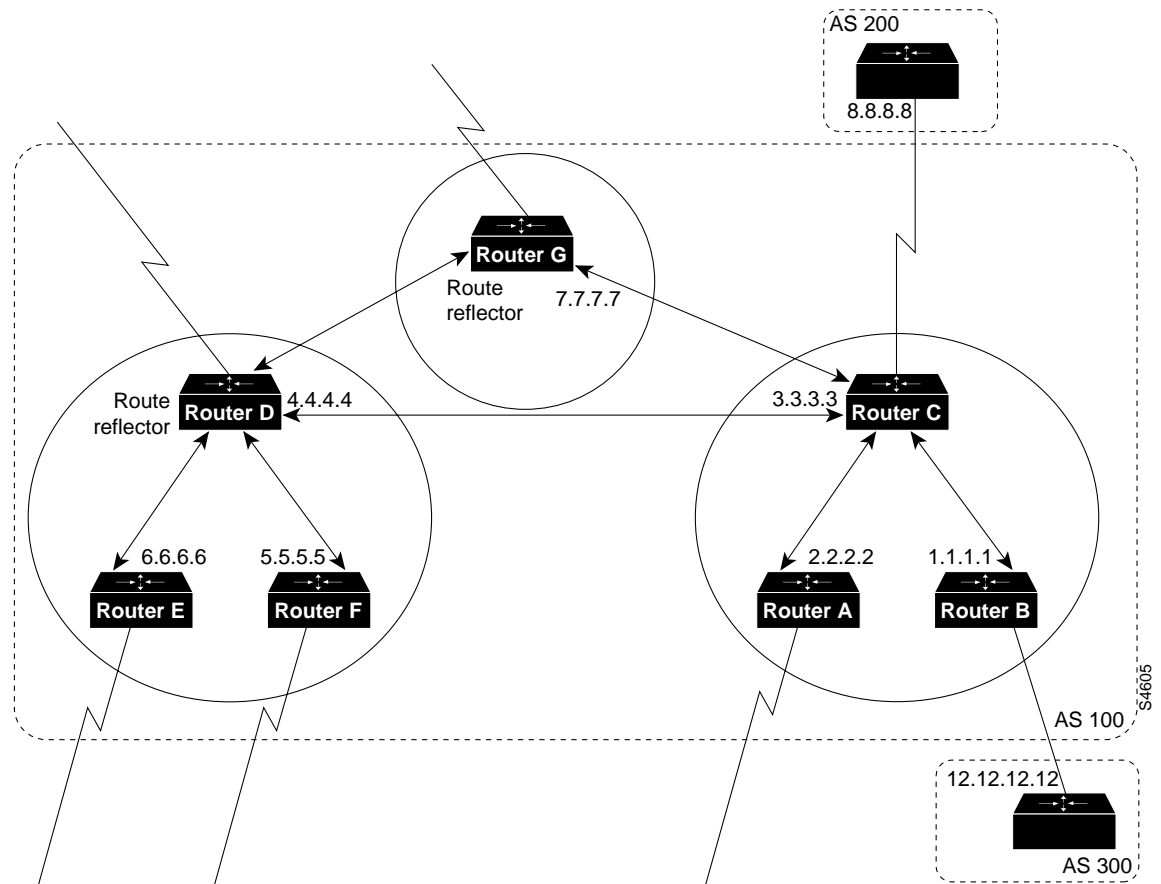
In Figure 12-30, Routers A, B, and C form a cluster, and Router C is the route reflector. Routers D, E, and F form a second cluster, of which Router D is the route reflector. Router G forms a third cluster. Note that Routers C, D, and G are fully meshed and that the routers within a cluster are not fully meshed.

When a route reflector in Figure 12-30 receives an update, it takes the following actions, depending on the type of peer that sent the update:

- Update from a nonclient peer—Send the update to all clients in the cluster.

- Update from a client peer—Send the update to all nonclient peers and to all client peers.

- Update from EBGP peer—Send the update to all nonclient peers and to all client peers.

The following configurations establish the route reflectors in AS 100:

```
!Router C
router bgp 100
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 4.4.4.4 remote-as 100
neighbor 8.8.8.8 remote-as 200
```

```
!Router B
router bgp 100
neighbor 3.3.3.3 remote-as 100
neighbor 12.12.12.12 remote-as 300

!Router D
router bgp 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 3.3.3.3 remote-as 100
neighbor 7.7.7.7 remote-as 100
```

If a set clause is used to modify an attribute, a routing loop may occur when the IBGP-learned routes are reflected. BGP automatically prevents the set clause of outgoing route maps from affecting routes reflected to IBGP peers. Another automatic restriction concerns the **neighbor next-hop-self** router configuration command. Because the next hop of reflected routes should not be changed, the **neighbor next-hop-self** command only affects the next hop of EBGP-learned routes when used with route reflectors.

Two techniques prevent routing loops in route reflector configurations:

- Using an Originator ID
- Using a Cluster List

Using an Originator ID

The originator ID is a 4-byte BGP attribute that is created by the route reflector. This attribute carries the router ID of the originator of the route in the local AS. If, because of poor configuration, the update comes back to the originator, the originator ignores it.

Using a Cluster List

Usually a cluster has a single route reflector, in which case, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid single points of failure, a cluster might have more than one route reflector. When a cluster has more than one route reflector, all of the route reflectors in the cluster need to be configured with a 4-byte cluster ID. The cluster ID allows route reflectors to recognize updates from other route reflectors in the same cluster.

A cluster list is a sequence of cluster IDs that an update has traversed. When a route reflector sends a route from its clients to nonclients outside of the cluster, it appends the local cluster ID to the cluster list. If the route reflector receives an update whose cluster list contains the local cluster ID, the update is ignored.

In Figure 12-31, Routers D, E, F, and H belong to the same cluster; Routers D and H are route reflectors for the same cluster. Note that Routers D and H maintain a fully meshed peering relationship with the other route reflectors in AS 100 (that is, with Routers C and G). If Router D goes down, Router H is prepared to take its place.

Figure 12-31   Route Reflectors and Cluster Lists
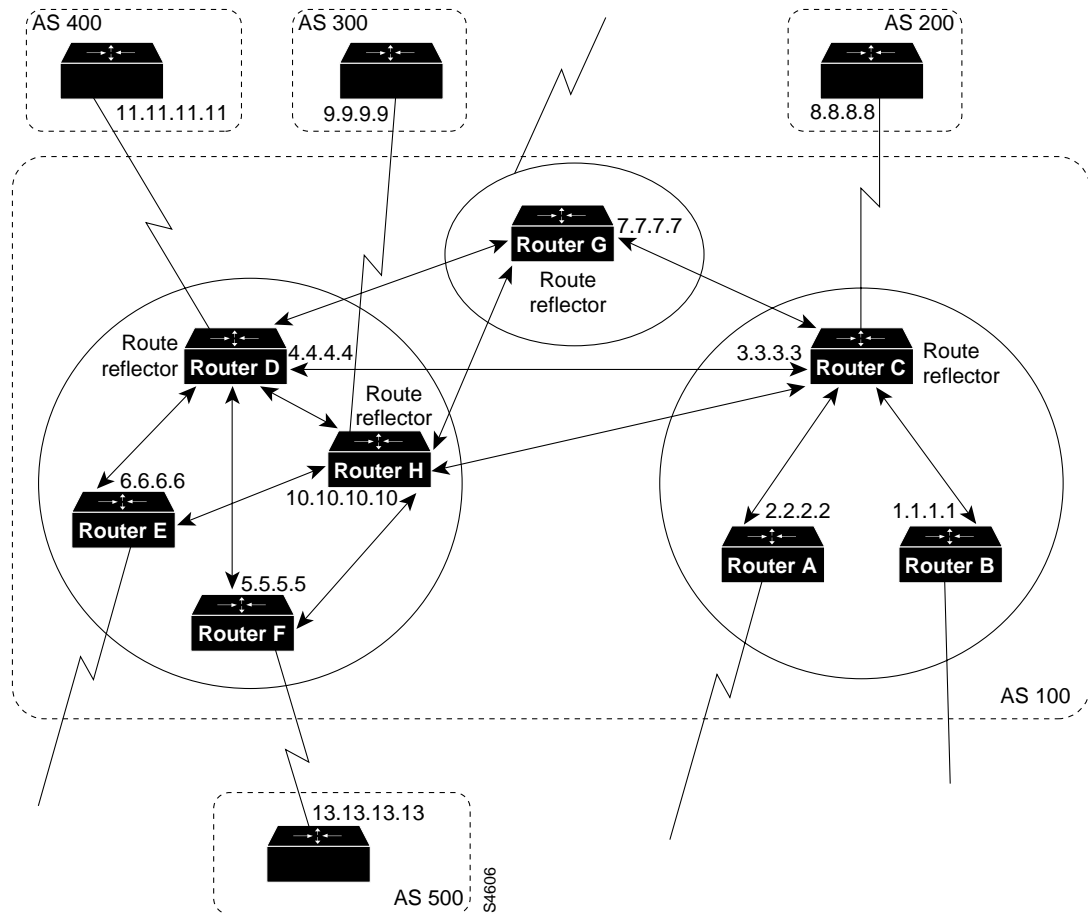
The following commands configure Routers C, D, F, and H:

```
!Router C
router bgp 100
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
neighbor 4.4.4.4 remote-as 100
neighbor 7.7.7.7 remote-as 100
neighbor 10.10.10.10 remote-as 100
neighbor 8.8.8.8 remote-as 200

!Router D
neighbor 10.10.10.10 remote-as 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 3.3.3.3 remote-as 100
neighbor 7.7.7.7 remote-as 100
neighbor 11.11.11.11 remote-as 400
bgp cluster-id 10
```

```
!Router F
router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 4.4.4.4 remote-as 100
neighbor 13.13.13.13 remote-as 500

!Router H
router bgp 100
neighbor 4.4.4.4 remote-as 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 3.3.3.3 remote-as 100
neighbor 9.9.9.9 remote-as 300
bgp cluster-id 10
```

The configurations for Routers D and H include the **bgp cluster-id** router configuration command, which sets the cluster ID to 10. The configuration for Router C does not include the **bgp cluster-id** command because Router C is the only route reflector in its cluster.

Note   You should not configure a peer group within a route reflector cluster. Clients inside a cluster do not have direct IBGP peers; instead, they exchange updates through the route reflector. Configuring peer groups within such a cluster might cause a withdrawal to the source of a route on the route reflector to be sent to all clients in the cluster. If you use the **no bgp client-to-client reflection** command to turn off client-to-client reflection on the route reflector and if you enable redundant BGP peering between the clients, you can use peer groups within a cluster.

Route Reflectors and Conventional BGP Speakers

It is normal for an AS in which route reflectors are configured to have BGP speakers that do not support route reflection. Such routers are known as conventional BGP speakers.

In Figure 12-32, Routers D, E, and F form a route reflector cluster, and Routers A, B, and C are conventional BGP speakers.

Figure 12-32   Route Reflectors and Conventional BGP Speakers
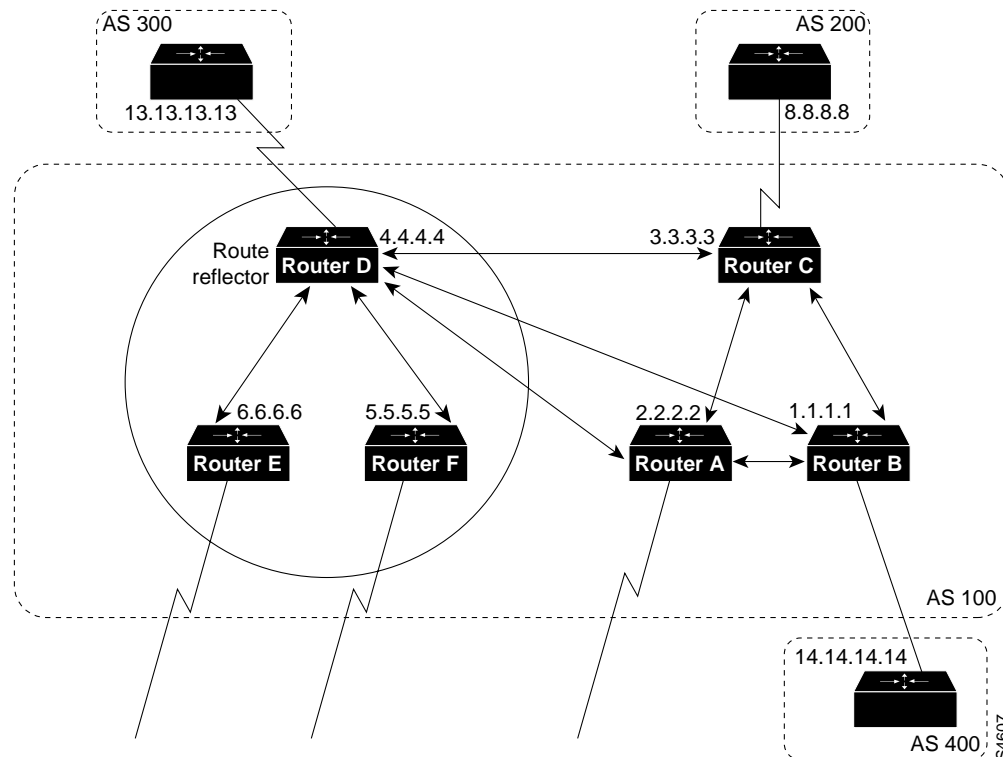
In Figure 12-32, each conventional BGP speaker is peered with the route reflector (Router D), and Routers A, B, and C are peered among each other.

The following commands configure Routers C and D:

```
!Router C
router bgp 100
neighbor 4.4.4.4 remote-as 100
neighbor 2.2.2.2 remote-as 100
neighbor 1.1.1.1 remote-as 100
neighbor 8.8.8.8 remote-as 200

!Router D
router bgp 100
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 3.3.3.3 remote-as 100
neighbor 2.2.2.2 remote-as 100
neighbor 1.1.1.1 remote-as 100
neighbor 13.13.13.13 remote-as 300
```

When it is time to make the conventional BGP speakers members of a cluster, Router C can be configured to be the route reflector, and Routers A and B can be its clients.

Route Flap Dampening

Route flap dampening (introduced in Cisco Internetwork Operating System [Cisco IOS] Release 11.0) is a mechanism for minimizing the instability caused by route flapping. The following terms are used to describe route flap dampening:

- Penalty—A numeric value that is assigned to a route when it flaps.

- Half-life time—A configurable numeric value that describes the time required to reduce the penalty by one half.

- Suppress limit—A numeric value that is compared with the penalty. If the penalty is greater than the suppress limit, the route is suppressed.

- Suppressed—A route that is not advertised even though it is up. A route is suppressed if the penalty is more than the suppressed limit.

- Reuse limit—A configurable numeric value that is compared with the penalty. If the penalty is less than the reuse limit, a suppressed route that is up will no longer be suppressed.

- History entry—An entry that is used to store flap information about a route that is down.

A route that is flapping receives a penalty of 1000 for each flap. When the accumulated penalty reaches a configurable limit, BGP suppresses advertisement of the route even if the route is up. The accumulated penalty is decremented by the half-life time. When the accumulated penalty is less than the reuse limit, the route is advertised again (if it is still up).

Note Dampening is not applied to routes that are learned via IBGP. This restriction avoids forwarding loops and prevents IBGP peers from having a higher penalty for routes that are external to the AS.

The network shown in Figure 12-33 demonstrates route flap dampening.

Figure 12-33   Route Flap Dampening
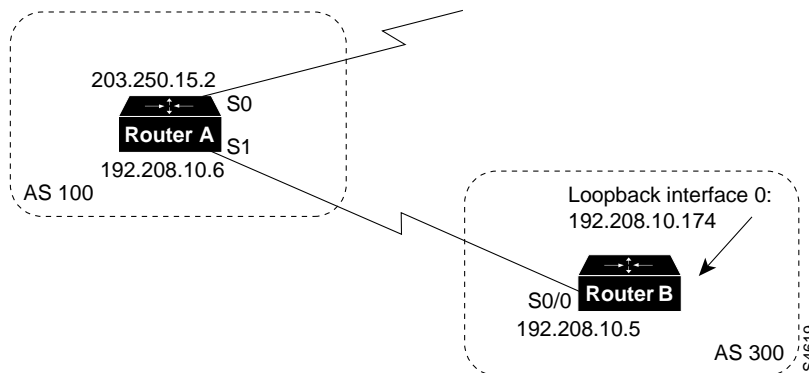
The following commands configure Routers A and B:

```
!RouterA
hostname RouterA
!
interface serial 0
ip address 203.250.15.2 255.255.255.252
```

```
interface serial 1
ip address 192.208.10.6 255.255.255.252
!
router bgp 100
bgp dampening
network 203.250.15.0
neighbor 192.208.10.5 remote-as 300

!RouterB
hostname RouterB
!
interface loopback 0
ip address 192.208.10.174 255.255.255.192
!
interface serial 0/0
ip address 192.208.10.5 255.255.255.252
!
router bgp 300
network 192.208.10.0
neighbor 192.208.10.6 remote-as 100
```

Router A is configured for route dampening. Assuming that the EBGP link to Router B is stable, the BGP table on Router A looks like this:

```
RouterB# show ip bgp
table version is 24, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network          Next Hop         Metric LocPrf Weight Path
*>   192.208.10.0     192.208.10.5          0             0 300 i
*>   203.250.15.0     0.0.0.0               0         32768 i
```

To simulate a route flap, enter this command on Router B:

```
clear ip bgp 192.208.10.6
```

Now, the BGP table on Router A looks like this:

```
RouterA# show ip bgp
table version is 24, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network          Next Hop         Metric LocPrf Weight Path
 h   192.208.10.0     192.208.10.5          0             0 300 i
*>   203.250.15.0     0.0.0.0               0         32768 i
```

Because the route for 192.208.10.0. has flapped, the BGP entry for 192.208.10.0 has been withdrawn and put into the history state.

The output of the **show ip bgp** EXEC command for network 192.208.10.0 is as follows:

```
RouterA# show ip bgp 192.208.10.0
BGP routing table entry for 192.208.10.5 255.255.255.0, version 25
Paths: (1 available, no best path)
300 (history entry)
    192.208.10.5 from 192.208.10.5 (192.208.10.174)
Origin IGP, metric 0, external
Dampinfo: penalty 1000, flapped 1 times in 0:02:03
```

The route has been given a penalty (1000) for flapping but the penalty is still below the suppress limit (default 2000). Because the route is down, it is marked as a history entry. If the route flaps a few more times, the **show ip bgp** command displays the following:

```
RouterA# show ip bgp
table version is 32, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop          Metric LocPrf Weight Path
*d    192.208.10.0     192.208.10.5           0             0 300 i
*>    203.250.15.0     0.0.0.0                0         32768 i
```

The output of the **show ip bgp** command for network 192.208.10.0 is as follows:

```
RouterA# show ip bgp 192.208.10.0
BGP routing table entry for 192.208.10.5 255.255.255.0, version 32
Paths: (1 available, no best path)
300, (suppressed due to dampening)
    192.208.10.5 from 192.208.10.5 (192.208.10.174)
Origin IGP, metric 0, external
Dampinfo: penalty 2615, flapped 3 times in 0:05:18, reuse in 0:27:00
```

The route is up, but because the penalty is greater than the suppress limit, it is suppressed. The route will be reused when the penalty reaches the reuse limit (default 750). The dampening information will be purged when the penalty becomes less than half of the reuse limit (750/2 = 350).

Practical Design Example

Figure 12-34 shows a BGP network that demonstrates the types of topologies that are typical among ISPs.

Figure 12-34   Practical Design Example for ISPs
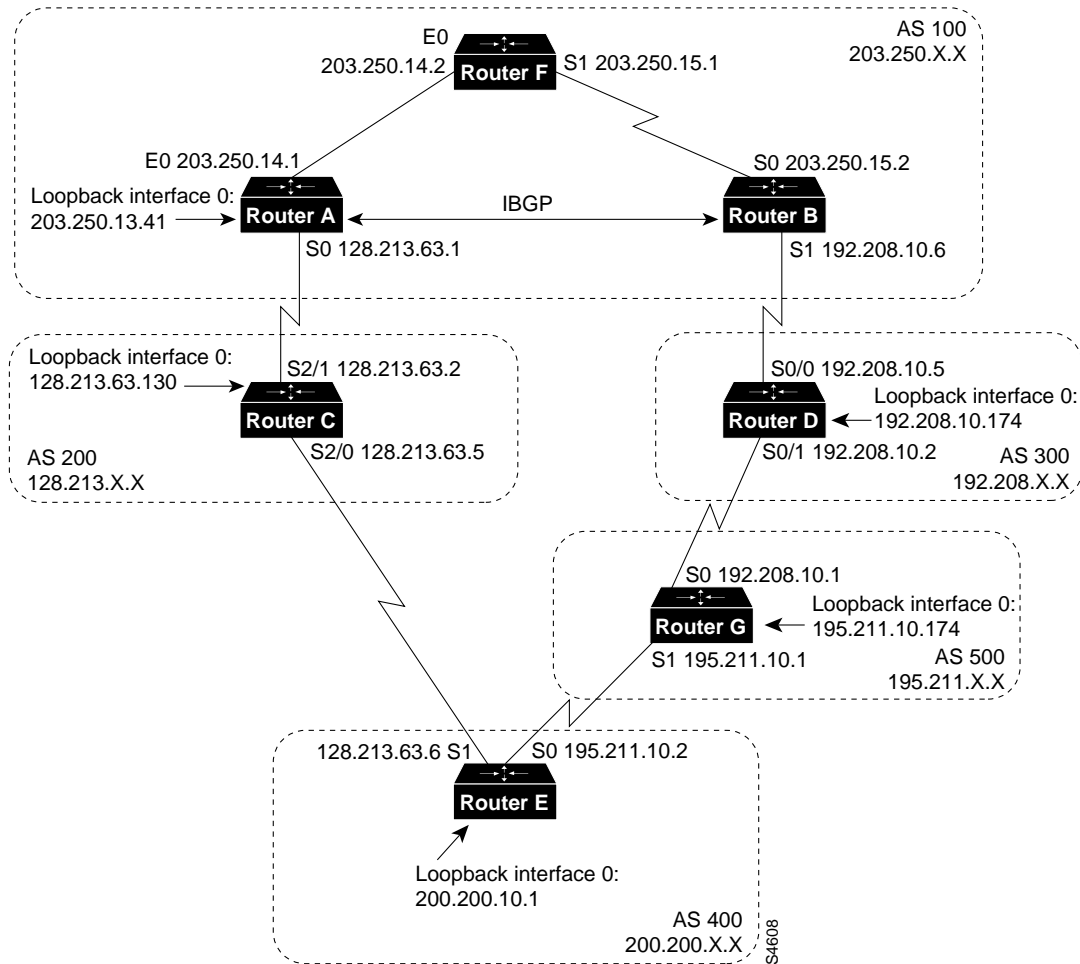
Whenever an AS is connected to two ISPs via EBGP, IBGP should be run within the AS for better control over routes. The following configurations for the routers shown in Figure 12-34 run OSPF as the IGP and run IBGP between Routers A and B inside AS 100.

The following configurations are preliminary configurations for the routers shown in Figure 12-34. These preliminary configurations are incomplete so that BGP troubleshooting techniques can be demonstrated. For the complete configurations, see the section, "Final Configurations," later in this chapter.

```
!Router A
hostname RouterA
!
interface loopback 0
ip address 203.250.13.41 255.255.255.0
!
interface ethernet 0
ip address 203.250.14.1 255.255.255.0
!
interface serial 0
ip address 128.213.63.1 255.255.255.252
!
router ospf 10
network 203.250.0.0 0.0.255.255 area 0
```

```
router bgp 100
network 203.250.13.0 mask 255.255.255.0
network 203.250.14.0 mask 255.255.255.0
neighbor 128.213.63.2 remote-as 200
neighbor 203.250.15.2 remote-as 100
neighbor 203.250.15.2 update-source loopback 0

!Router B
hostname RouterB
!
interface serial 0
ip address 203.250.15.2 255.255.255.252
!
interface serial 1
ip address 192.208.10.6 255.255.255.252
!
router ospf 10
network 203.250.0.0 0.0.255.255 area 0
!
router bgp 100
network 203.250.15.0
neighbor 192.208.10.5 remote-as 300
neighbor 203.250.13.41 remote-as 100

!Router C
hostname RouterC
!
interface loopback 0
ip address 128.213.63.130 255.255.255.192
!
interface serial 2/0
ip address 128.213.63.5 255.255.255.252
!
interface serial 2/1
ip address 128.213.63.2 255.255.255.252
!
router bgp 200
network 128.213.0.0
neighbor 128.213.63.1 remote-as 100
neighbor 128.213.63.6 remote-as 400

!Router D
hostname RouterD
!
interface loopback 0
ip address 192.208.10.174 255.255.255.192
!
interface serial 0/0
ip address 192.208.10.5 255.255.255.252
!
interface serial 0/0
ip address 192.208.10.5 255.255.255.252
!
router bgp 300
network 192.208.10.0
neighbor 192.208.10.1 remote-as 500
neighbor 192.208.10.6 remote-as 100

!Router E
hostname RouterE
!
interface loopback 0
ip address 200.200.10.1 255.255.255.0
```

```
interface serial 0
ip address 195.211.10.2 255.255.255.252
!
interface serial 1
ip address 128.213.63.6 255.255.255.252
!
router bgp 400
network 200.200.10.0
neighbor 128.213.63.5 remote-as 200
neighbor 195.211.10.1 remote-as 500

!Router F
hostname RouterF
!
interface ethernet 0
ip address 203.250.14.2 255.255.255.0
!
interface serial 1
ip address 203.250.15.1 255.255.255.252
!
router ospf 10
network 203.250.0.0 0.0.255.255 area 0

!Router G
hostname RouterG
!
interface loopback 0
ip address 195.211.10.174 255.255.255.192
!
interface serial 0
ip address 192.208.10.1 255.255.255.252
!
interface serial 1
ip address 195.211.10.1 255.255.255.252
!
router bgp 500
network 195.211.10.0
neighbor 192.208.10.2 remote-as 300
neighbor 195.211.10.2 remote-as 400
```

When you redistribute IGP routes into BGP, you need to control the routes that are injected into BGP. For that reason, it is always better to advertise routes by using the **network** router configuration command or by redistributing static routes, as shown in the examples in this section. This method also avoids route flaps.

Determining the State of BGP

Assume that in Figure 12-34 the connection between Routers B and D is down. The following information is displayed when you enter the **show ip bgp** EXEC command on Router B:

```
RouterB# show ip bgp
table version is 4, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network          Next Hop          Metric LocPrf Weight Path
*i128.213.0.0      128.213.63.2           0     100       0  200 i
*i192.208.10.0     128.213.63.2                 100       0  200 400 500 300 i
*i195.211.10.0     128.213.63.2                 100       0  200 400 500 i
*i200.200.10.0     128.213.63.2                 100       0  200 400 i
*>i203.250.13.0    203.250.13.41          0     100       0  i
*>i203.250.14.0    203.250.13.41          0     100       0  i
*> 203.250.15.0    0.0.0.0                0            32768  i
```

The letter i at the beginning of a line means that the entry was learned via an internal BGP peer. The letter i at the end of a line indicates that the path information comes from an IGP. The first entry reads as follows: Network 128.213.0.0 is learned via path 200 and has a next hop of 128.213.63.2. Note that any locally generated entry, such as 203.250.15.0 has a next hop of 0.0.0.0.

The > symbol indicates that BGP has chosen the best route based on the decision steps described in the section "Summary of the BGP Path Selection Process," earlier in this chapter. BGP picks only the one route that it determines to be the best route. It installs this route in the IP routing table and advertises it to other BGP peers. Note the next hop attribute of 128.213.63.2, which is the EBGP next hop carried into IBGP.

Following is the contents of the IP routing table on Router B:

```
RouterB# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/75] via 203.250.15.1, 02:50:45, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 02:40:46, Serial0
```

Note than none of the BGP entries appears in the IP routing table. One problem is that the next hop for these entries (128.213.63.2) is unreachable. This is because there is no way to reach that next hop via the IGP (in this case, OSPF). Router B has not learned about 128.213.63.0 via OSPF.

Correcting Next Hop Problems

For the network shown in Figure 12-34, the next hop problem can be corrected in one of two ways:

- On Router A, use the **neighbor next-hop-self** router configuration command to change the next hop between Router A and Router B.

- On Router A, run OSPF on interface serial 0 and make it passive. This way, Router B will know how to reach the next hop 128.213.63.2.

The following configuration for Router A runs OSPF on interface serial 0 and makes it passive:

```
!Router A
hostname RouterA
!
interface loopback 0
ip address 203.250.13.41 255.255.255.0
!
interface ethernet 0
ip address 203.250.14.1 255.255.255.0
!
interface serial 0
ip address 128.213.63.1 255.255.255.252
!
router ospf 10
passive-interface serial 0
network 203.250.0.0 0.0.255.255 area 0
network 128.213.0.0 0.0.255.255 area 0
!
router bgp 100
network 203.250.13.0 mask 255.255.255.0
```

```
network 203.250.14.0 mask 255.255.255.0
neighbor 128.213.63.2 remote-as 200
neighbor 203.250.15.2 remote-as 100
neighbor 203.250.15.2 update-source loopback 0
```

Now the BGP neighbor table on Router B contains the following routes:

```
RouterB# show ip bgp
table version is 10, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network          Next Hop        Metric LocPrf Weight Path
*>i128.213.0.0      128.213.63.2          0    100      0 200 i
*>i192.208.10.0     128.213.63.2               100      0 200 400 500 300 i
*>i195.211.10.0     128.213.63.2               100      0 200 400 500 i
*>i200.200.10.0     128.213.63.2               100      0 200 400 i
*>i203.250.13.0     203.250.13.41         0    100      0 i
*>i203.250.14.0     203.250.13.41         0    100      0 i
*> 203.250.15.0     0.0.0.0               0         32768 i
```

Note that a > symbol appears in all of the entries, which means that BGP is satisfied with the next hop address.

Now the IP routing table on Router B contains the following routes:

```
RouterB# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/75] via 203.250.15.1, 00:04:46, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:04:46, Serial0
     128.213.0.0 255.255.255.252 is subnetted, 1 subnets
O       1.28.213.63.0 [110/138] via 203.250.15.1, 00:04:47, Serial 0
```

Note that the BGP entries still do not appear in the IP routing table. The only difference is that 128.213.63.0 is now reachable via OSPF. The problem is synchronization: BGP is not synchronized with the IGP, so it does not put the entries in the IP routing table, and it does not send the entries in BGP updates. Router F is not aware of networks 192.208.10.0 or 195.211.10.0 because BGP routes are not redistributed into OSPF yet.

Turning Off Synchronization

If you enter the **no synchronization** router configuration command on Router B and then examine the IP routing table on Router B, you see the following contents of the IP routing table on Router B:

```
RouterB# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort not set

B    200.200.10.0 [200/0] via 128.213.63.2, 00:01:07
B    195.211.10.0 [200/0] via 128.213.63.2, 00:01:07
B    192.208.10.0 [200/0] via 128.213.63.2, 00:01:07
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
            [110/75] via 203.250.15.1, 00:12:37, Serial0
B       203.250.13.0 255.255.255.0 [200/0] via 203.250.13.41, 00:01:08
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:12:37, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
B       128.213.0.0 255 255.0.0 [200/0] via 128.213.63.2, 00:01:08
O       128.213.63.0 255.255.255.252
            [110/138] via 203.250.15.1, 00:12:37, Serial0
```

The routing table looks fine, but there is no way to reach those networks because Router F in the middle does not know how to reach them, as shown by the following output of the **show ip route** EXEC command on Router F:

```
RouterF# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/11] via 203.250.14.1, 00:14:15, Ethernet0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 255.255.255.252 is subnetted, 1 subnets
O       128.213.63.0 [110/74] via 203.250 14 1, 00:14:15, Ethernet0
```

If packets to the BGP network are forwarded to Router F, they will be dropped, so turning off synchronization does not resolve this particular problem. OSPF still needs to be redistributed into BGP on Router A so that Router F learns about BGP routes.

Redistributing OSPF

The following configuration for Router A has been modified to redistribute OSPF (the new command is in bold):

```
!Router A
hostname RouterA
!
interface loopback 0
ip address 203.250.13.41 255.255.255.0
```

```
interface ethernet 0
ip address 203.250.14.1 255.255.255.0
!
interface serial 0
ip address 128.213.63.1 255.255.255.252
!
router ospf 10
redistribute bgp 100 metric 2000 subnets
passive-interface serial 0
network 203.250.0.0 0.0.255.255 area 0
network 128.213.0.0 0.0.255.255 area 0
!
router bgp 100
network 203.250.0.0 mask 255.255.0.0
neighbor 128.213.63.2 remote-as 200
neighbor 203.250.15.2 remote-as 100
neighbor 203.250.15.2 update-source loopback 0
```

Now the routing table looks as follows:

```
RouterB# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort not set
O E2 200.200.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 195.211.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 192.208.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
            [110/75] via 203.250.15.1, 00:00:15, Serial0
O E2    203.250.13.0 255.255.255.0
            [110/2000] via 203.250.15.1, 00:00:15, Serial0
        203.250.15.0 255.255.255.252 is subnetted, 2 subnets
C         203.250.15.8 is directly connected, Loopback1
C         203.250.15.0 is directly connected, Serial0
O       203.250.14.0 [110/74] via 203.250.15.1, 00:00:15, Serial0
        128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2      128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:00:15,Serial0
O       128.213.63.0 255.255.255.252
            [110/138] via 203.250.15.1, 00:00:16, Serial0
```

The BGP entries have disappeared because OSPF has a better distance (110) than IBGP (200).

Turning off synchronization on Router A will cause Router A to advertise network 203.250.15.0. This step is required because Router A will not synchronize with OSPF because of mask differences. For the same reason, synchronization should also be turned off on Router B so that it can advertise network 203.250.13.0.

In addition, OSPF should be enabled on interface serial 1 on Router B and made passive so that Router A learns about next hop 192.208.10.5 via an IGP.

The modified configurations for Routers A and B are as follows. (New commands are in bold.)

```
!Router A
hostname RouterA
!
interface loopback 0
ip address 203.250.13.41 255.255.255.0
!
interface ethernet 0
ip address 203.250.14.1 255.255.255.0
!
interface serial 0
ip address 128.213.63.1 255.255.255.252
!
router ospf 10
redistribute bgp 100 metric 2000 subnets
passive-interface serial 0
network 203.250.0.0 0.0.255.255 area 0
network 128.213.0.0 0.0.255.255 area 0
!
router bgp 100
no synchronization
network 203.250.13.0 mask 255.255.255.0
network 203.250.14.0 mask 255.255.255.0
neighbor 128.213.63.2 remote-as 200
neighbor 203.250.15.2 remote-as 100
neighbor 203.250.15.2 update-source loopback 0

!Router B
hostname RouterB
!
interface serial 0
ip address 203.250.15.2 255.255.255.252
!
interface serial 1
ip address 192.208.10.6 255.255.255.252
!
router ospf 10
redistribute bgp 100 metric 1000 subnets
passive-interface serial 1
network 203.250.0.0 0.0.255.255 area 0
network 192.208.0.0 0.0.255.255 area 0
!
router bgp 100
network 203.250.15.0
neighbor 192.208.10.5 remote-as 300
neighbor 203.250.13.41 remote-as 100
```

Now bring up interface serial 1 on Router B and see what the BGP neighbor table looks like on Router A:

```
RouterA# show ip bgp
table version is 117, local router ID is 203.250.13.41
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network          Next Hop          Metric LocPrf Weight Path
*> 128.213.0.0       128.213.63.2           0    100       0 200 i
*>i192.208.10.0      192.208.10.5           0    100       0 300 i
*>i195.211.10.0      192.208.10.5                100       0 300 500 i
*                    128.213.63.2                          0 200 400 500 i
*> 203.250.13.0      0.0.0.0                0            32768 i
*> 203.250.14.0      0.0.0.0                0            32768 i
*>i203.250.15.0      203.250.15.2           0    100       0 i
```

Following is the output of the **show ip route** EXEC command on Router A:

```
RouterA# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort not set

     192.208.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.208.10.0 255.255.255.0
            [110/1000] via 203.250.14.2, 00:41:25, Ethernet0
O    192.208.10 4 255.255.255.252
            [110/138] via 203.250.14.2, 00:41:25, Ethernet0
C    203.250.13.0 is directly connected, Loopback0
     203.250.15.0 is variably subnetted, 3 subnets, 3 masks
O       203.250.15.10 255.255.255.255
            [110/75] via 203.250.14.2, 00:41:25, Ethernet0
O       203.250.15.0 255.255.255.252
            [110/74] via 203.250.14.2, 00:41:25, Ethernet0
B       203.250.15.0 255.255.255.0 [200/0] via 203.250.15.2, 00:41:25
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
B       128.213.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:41:26
C       128.213.63.0 255.255.255.252 is directly connected, Serial0
B*   200.200.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:02:38
```

Following is the output of the **show ip bgp** EXEC command on Router B:

```
RouterB# show ip bgp
table version is 12, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network          Next Hop        Metric LocPrf Weight Path
*>i128.213.0.0        128.213.63.2         0    100      0 200 i
*                     192.208.10.5                       0 300 500 400 200 i
*> 195.208.10.0       192.208.10.5         0             0 300 i
*> 195.211.10.0       192.208.10.5                       0 300 500 i
*>i200.200.10.0       128.213.63.2              100      0 200 400 i
*>                    192.208.10.5                       0 300 500 400 i
*>i203.250.13.0       203.250.13.41        0    100      0 i
*>i203.250.14.0       203.250.13.41        0    100      0 i
*> 203.250.15.0       0.0.0.0              0          32768 i
```

Managing Asymmetry

There are several ways to design the network for AS 100 to communicate with the ISP networks in AS 200 and AS 300. One way is to have a primary ISP and a backup ISP. AS 100 could learn partial routes from one of the ISPs and default routes to both ISPs. In this example, AS 100 receives partial routes from AS 200 and only local routes from AS 300. Both Routers A and B generate default routes into OSPF, with Router B being the more preferred route because of its lower MED attribute. This allows you to balance outgoing traffic between the two ISPs.

Potential asymmetry might occur if traffic going out from Router A comes back via Router B. This might occur if networks are advertised to both of the ISPs. From outside the AS, the networks are reachable via both of the ISPs and either Router A or B could be used to reach them. You might find out that all incoming traffic to your AS is coming via one single point even though you have multiple points to the internetwork.

One other potential reason for asymmetry is the different advertised path length to reach your AS. One ISP might be closer to a certain destination than another. In this example, traffic from AS 400 destined for AS 100 always comes in via Router A because of the shorter path. You might try to affect that decision by using the **set as-path route** map configuration command with the **prepend** keyword to prepend AS numbers to your updates to make the AS_path attribute longer. But, if AS 400 has somehow set its exit point to be via AS 200 based on attributes such as local preference, MED attribute, weight, there is nothing you can do.

Final Configurations

Following is the final configuration for Router A. (New or modified commands are in bold.)

```
!Router A
hostname RouterA
!
interface loopback 0
ip address 203.250.13.41 255.255.255.0
!
interface ethernet 0
ip address 203.250.14.1 255.255.255.0
!
interface serial 0
ip address 128.213.63.1 255.255.255.252
!
router ospf 10
redistribute bgp 100 metric 2000 subnets
passive-interface serial 0
network 203.250.0.0 0.0.255.255 area 0
network 128.213.0.0 0.0.255.255 area 0
default-information originate metric 2000
!
router bgp 100
no synchronization
neighbor 128.213.63.2 remote-as 200
neighbor 128.213.63.2 route-map setlocalpref in
neighbor 203.250.15.2 remote-as 100
neighbor 203.250.15.2 update-source loopback 0
!
ip default-network 200.200.0.0
!
route-map setlocalpref permit 10
set local-preference 200
```

The final configuration for Router A sets the local preference for routes coming from AS 200 to 200. The configuration also uses the **ip default-network** global configuration command to specify network 200.200.0.0 as the candidate default route. The **ip default-information originate** router configuration command is used to inject the default route inside the OSPF domain. For RIP, network 0.0.0.0 is automatically redistributed into RIP without additional configuration. For IGRP and Enhanced IGRP, default information is injected into the IGP domain after BGP is redistributed. Also, with IGRP and Enhanced IGRP, you can redistribute a static route for 0.0.0.0 into the IGP domain.

Following is the final configuration for Router B. (New or modified commands are in bold.)

```
!Router B
hostname RouterB
!
interface serial 0
ip address 203.250.15.2 255.255.255.252
!
interface serial 1
ip address 192.208.10.6 255.255.255.252
```

```
router ospf 10
redistribute bgp 100 metric 1000 subnets
passive-interface serial 1
network 203.250.0.0 0.0.255.255 area 0
network 192.208.0.0 0.0.255.255 area 0
default-information originate metric 1000
!
router bgp 100
no synchronization
network 203.250.15.0
neighbor 192.208.10.5 remote-as 300
neighbor 192.208.10.5 route-map LOCALONLY in
neighbor 203.250.13.41 remote-as 100
!
ip default-network 192.208.10.0
ip as-path access-list 1 permit ^300 500$
ip as-path access-list 2 permit ^300$
!
route-map LOCALONLY permit 10
match as-path 1
set local-preference 300
!
route-map LOCALONLY permit 20
match as-path 2
```

The configuration for Router B sets the local preference for updates coming from AS 300 having an AS_path attribute of 300, 500 to 300, which is higher than the IBGP updates coming in from Router A in AS 100. This way, AS 100 will pick Router B for AS 500's local routes. Any other routes on Router B (if there are any) will be sent internally with a local preference of 100, which is lower than the local preference of 200 coming in from Router A. This arrangement causes Router A to be preferred. Further, because of the length of the AS_path attribute, Router B is used to reach routes local to AS 300.

Note that Router B accepts the local routes of AS 300 and AS 500 only. Any updates whose AS_path attribute does not match are dropped. If you want to advertise the local routes and the neighbor routes (customers of the ISP), you can use ^300_[0-9]* as the regular expression. The following is the output of the **show ip bgp** EXEC command for regular expression ^300$:

```
RouterB# show bgp regexp ^300$
BGP table version is 14, local router ID is 203.250.15.2
Status code: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 192.208.10.0     192.28.10.5            0    300      0 300
```

Following is the final configuration for Router C. (New and modified commands are in bold.)

```
!Router C
hostname RouterC
!
interface loopback 0
ip address 128.213.63.130 255.255.255.192
!
interface serial 2/0
ip address 128.213.63.5 255.255.255.252
!
interface serial 2/1
ip address 128.213.63.2 255.255.255.252
!
router bgp 200
network 128.213.0.0
aggregate-address 128.213.0.0 255.255.0.0 summary-only
```

```
neighbor 128.213.63.1 remote-as 100
neighbor 128.213.63.1 distribute-list 1 out
neighbor 128.213.63.6 remote-as 400
!
access-list 1 deny 195.211.0.0 0.0.255.255
access-list 1 permit any
```

The configuration for Router C aggregates network 128.213.0.0/16 and specifies the routes that are to be injected into AS 100. If the ISP refuses to do this task, you have to filter routes coming into AS 100 on Router A.

Following are the final configurations for Routers D and E. (New or modified commands are in bold.)

```
!Router D
hostname RouterD
!
interface loopback 0
ip address 192.208.10.174 255.255.255.192
!
interface serial 0/0
ip address 192.208.10.5 255.255.255.252
!
interface serial 0/1
ip address 192.208.10.2 255.255.255.252
!
router bgp 300
network 192.208.10.0
neighbor 192.208.10.1 remote-as 500
neighbor 192.208.10.6 remote-as 100

!Router E
hostname RouterE
!
interface loopback 0
ip address 200.200.10.1 255.255.255.0
interface serial 0
ip address 195.211.10.2 255.255.255.252
!
interface serial 1
ip address 128.213.63.6 255.255.255.252
!
router bgp 400
network 200.200.10.0
aggregate-address 200.200.0.0 255.255.0.0 summary-only
neighbor 128.213.63.5 remote-as 200
neighbor 195.211.10.1 remote-as 500
```

Router E is aggregating network 200.200.0.0/16.

Following are the final configurations for Routers F and G. (New or modified commands are in bold.)

```
!Router F
hostname RouterF
!
interface ethernet 0
ip address 203.250.14.2 255.255.255.0
!
interface serial 1
ip address 203.250.15.1 255.255.255.252
!
router ospf 10
network 203.250.0.0 0.0.255.255 area 0
!Router G
hostname RouterG
```

```
!
interface loopback 0
ip address 195.211.10.174 255.255.255.192
!
interface serial 0
ip address 192.208.10.1 255.255.255.252
!
interface serial 1
ip address 195.211.10.1 255.255.255.252
!
router bgp 500
network 195.211.10.0
aggregate-address 195.211.0.0 255.255.0.0 summary-only
neighbor 192.208.10.2 remote-as 300
neighbor 192.208.10.2 send-community
neighbor 192.208.10.2 route-map setcommunity out
neighbor 195.211.10.2 remote-as 400
!
access-list 2 permit any
access-list 101 permit ip 195.211.0.0 0.0.255.255 255.255.255.0 0.0.0.255
!
route-map setcommunity permit 10
match ip address 101
set community no-export
!
route-map setcommunity permit 20
match ip address 2
```

The configuration for Router G demonstrates the use of community filtering by adding the no-export community to more specific Class C routes of 195.211.0.0/16 that are sent to Router D. This way, Router D will not export that route to Router B.

Following is the final content of BGP routing table on Router A:

```
RouterA# show ip bgp
table version is 21, local router ID is 203.250.13.41
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network          Next Hop         Metric LocPrf Weight Path
*> 128.213.0.0       128.213.63.2           0    200      0 200 i
*>i192.208.10.0      192.208.10.5           0    300      0 300 i
*> 200.200.0.0/16    128.213.63.2                200      0 200 400 i
*> 203.250.13.0      0.0.0.0                0          32768 i
*> 203.250.14.0      0.0.0.0                0          32768 i
*>i203.250.15.0      203.250.15.2           0    100      0 i
```

Following is the final content of the IP routing table on Router A:

```
RouterA# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 128.213.63.2 to network 200.200.0.0

     192.208.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.208.10.0 255.255.255.0
            [110/1000] via 203.250.14.2, 00:41:25, Ethernet0
O       192.208.10.4 255.255.255.252
            [110/138] via 203.250.14.2, 00:41:25, Ethernet0
C    203.250.13.0 is directly connected, Loopback0
     203.250.15.0 is variably subnetted, 3 subnets, 3 masks
O       203.250.15.10 255.255.255.255
            [110/75] via 203.250.14.2, 00:41:25, Ethernet0
O       203.250.15.0 255.255.255.252
            [110/74] via 203.250.14.2, 00:41:25, Ethernet0
B       203.250.15.0 255.255.255.0 [200/0] via 203.250.15.2, 00:41:25
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
B       128.213.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:41:26
C       128.213.63.0 255.255.255.252 is directly connected, Serial0
B*   200.200.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:02:38
```

Following is the final content of IP routing table on Router F:

```
RouterF# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 203.250.15.2 to network 0.0.0.0

     192.208.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.208.10.0 255.255.255.0
            [110/1000] via 203.250.15.2, 00:48:50, Serial1
O       192.208.10.4 255.255.255.252
            [110/128] via 203.250.15.2, 01:12:09, Serial1
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
            [110/11] via 203.250.14.1, 01:12:09, Ethernet0
O E2    203.250.13.0 255.255.255.0
            [110/2000] via 203.250.14.1, 01:12:09, Ethernet0
     203.250.15.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.15.10 255.255.255.255
            [110/65] via 203.250.15.2, 01:12:09, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 255.255.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0
            [110/2000] via 203.250.14.1, 00:45:01, Ethernet0
O E2 200.200.0.0 255.255.0.0 [110/1000] via 203.250.14.1, 00:03:47, Ethernet0
O*E2 0.0.0.0 0.0.0.0 [110/1000] via 203.250.15.2, 00:03:33, Serial1
```

Note that on Router F, the routing table indicates that networks local to AS 300, such as 192.208.10.0 are to be reached via Router B. Other known networks, such as 200.200.0.0 are to be reached via Router A. The gateway of last resort is set to Router B. If something happens to the connection between Router B and Router D, the default advertised by Router A will kick in with a MED attribute of 2000.

Following is the final content of BGP routing table on Router B:

```
RouterB# show ip bgp
table version is 14, local router ID is 203.250.15.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network          Next Hop       Metric LocPrf Weight Path
*>i128.213.0.0      128.213.63.2          0    200      0 200 i
*> 192.208.10.0     192.208.10.5          0    300      0 300 i
*>i200.200.0.0/16   128.213.63.2               200      0 200 400 i
*>i203.250.13.0     203.250.13.41         0    100      0 i
*>i203.250.14.0     203.250.13.41         0    100      0 i
*> 203.250.15.0     0.0.0.0               0         32768 i
```

Following is the final content of the IP routing table on Router B:

```
RouterF# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 203.250.15.2 to network 192.208.10.0

*    192.208.10.0 is variably subnetted, 2 subnets, 2 masks
B*    192.208.10.0 255.255.255.0 [20/0] via 192.208.10.5, 00:50:46
C     192.208.10.4 255.255.255.252 is directly connected, Serial1
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O     203.250.13.41 255.255.255.255
         [110/75] via 203.250.15.1, 01:20:33, Serial0
O E2  203.250.13.0 255.255.255.0
         [110/2000] via 203.250.15.1, 01:15:40, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 2 subnets, 2 masks
O     203.250.15.10 255.255.255.255
         [110/65] via 203.250.15.2, 01:12:09, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 255.255.0.0 is variably subnetted, 2 subnets
C     203.250.15.8 id directly connected, Loopback1
C     203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 01:20:33, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2  128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:46:55, Serial0
```

Summary

The primary function of a BGP system is to exchange network reachability information with other BGP systems. This information is used to construct a graph of AS connectivity from which routing loops are pruned and with which AS-level policy decisions are enforced. BGP provides a number of techniques for controlling the flow of BGP updates, such as route, path, and community filtering. It also provides techniques for consolidating routing information, such as CIDR aggregation, confederations, and route reflectors. BGP is a powerful tool for providing loop-free interdomain routing within and between ASs.