Designing Large-Scale IP Internetworks

This chapter focuses on the following design implications of the Enhanced Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF) protocols, and the Border Gateway Protocol (BGP):

- Network Topology
- Addressing and Route Summarization
- Route Selection
- Convergence
- Network Scalability
- Security

Enhanced IGRP, OSPF, and BGP are routing protocols for the Internet Protocol (IP). An introductory discussion outlines general routing protocol issues; subsequent discussions focus on design guidelines for the specific IP protocols.

Implementing Routing Protocols

The following discussion provides an overview of the key decisions you must make when selecting and deploying routing protocols. This discussion lays the foundation for subsequent discussions regarding specific routing protocols.

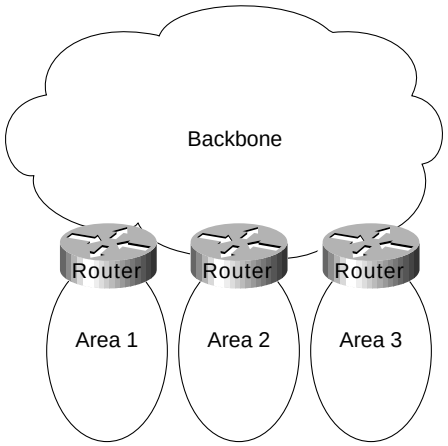
Network Topology

The physical topology of an internetwork is described by the complete set of routers and the networks that connect them. Networks also have a logical topology. Different routing protocols establish the logical topology in different ways.

Some routing protocols do not use a logical hierarchy. Such protocols use addressing to segregate specific areas or domains within a given internetworking environment and to establish a logical topology. For such nonhierarchical, or *flat*, protocols, no manual topology creation is required.

Other protocols require the creation of an explicit hierarchical topology through establishment of a backbone and logical areas. The OSPF and Intermediate System-to-Intermediate System (IS-IS) protocols are examples of routing protocols that use a hierarchical structure. A general hierarchical network scheme is illustrated in Figure 3-1. The explicit topology in a hierarchical scheme takes precedence over the topology created through addressing.

Figure 3-1 Hierarchical network.



If a hierarchical routing protocol is used, the addressing topology should be assigned to reflect the hierarchy. If a flat routing protocol is used, the addressing implicitly creates the topology. There are two recommended ways to assign addresses in a hierarchical network. The simplest way is to give each area (including the backbone) a unique network address. An alternative is to assign address ranges to each area.

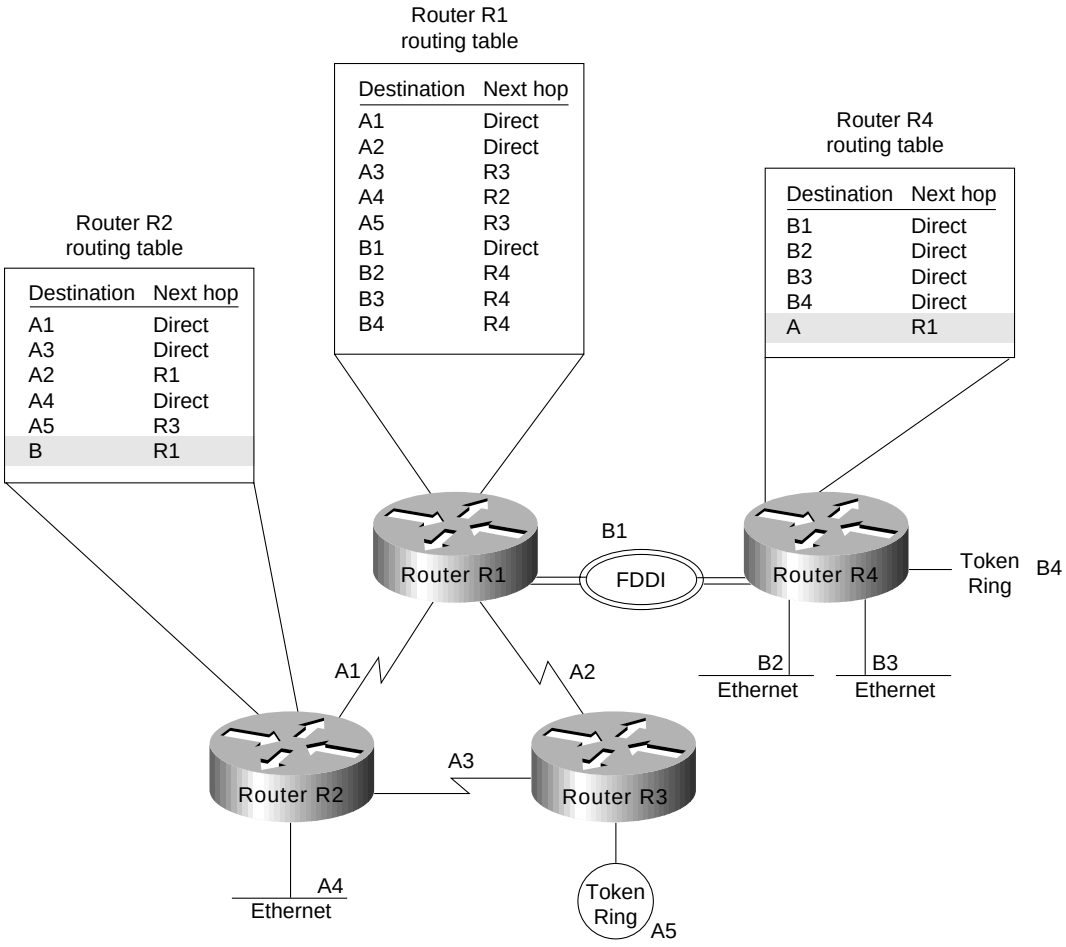
Areas are logical collections of contiguous networks and hosts. Areas also include all the routers having interfaces on any one of the included networks. Each area runs a separate copy of the basic routing algorithm. Therefore, each area has its own topological database.

Addressing and Route Summarization

Route summarization procedures condense routing information. Without summarization, each router in a network must retain a route to every subnet in the network. With summarization, routers can reduce some sets of routes to a single advertisement, reducing both the load on the router and the perceived complexity of the network. The importance of route summarization increases with network size.

Figure 3-2 illustrates an example of route summarization. In this environment, Router R2 maintains one route for all destination networks beginning with B, and Router R4 maintains one route for all destination networks beginning with A. This is the essence of route summarization. Router R1 tracks all routes because it exists on the boundary between A and B.

Figure 3-2 Route summarization example.

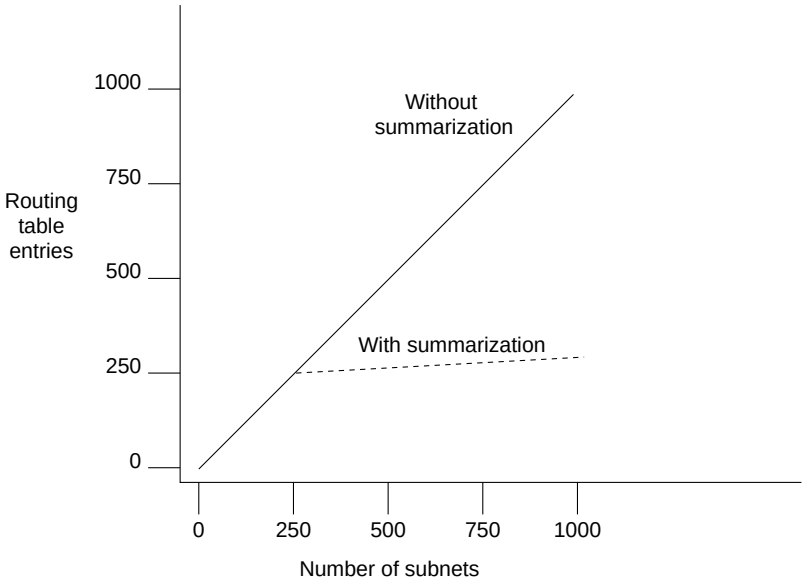


The reduction in route propagation and routing information overhead can be significant. Figure 3-3 illustrates the potential savings. The vertical axis of Figure 3-3 shows the number of routing table entries. The horizontal axis measures the number of subnets. Without summarization, each router in a network with 1,000 subnets must contain 1,000 routes. With summarization, the picture changes considerably. If you assume a Class B network with eight bits of subnet address space, each router needs to know all of the routes for each subnet in its network number (250 routes, assuming that 1,000 subnets fall into four major networks of 250 routers each) plus one route for each of the other networks (three) for a total of 253 routes. This represents a nearly 75-percent reduction in the size of the routing table.

The preceding example shows the simplest type of route summarization: collapsing all the subnet routes into a single network route. Some routing protocols also support route summarization at any bit boundary (rather than just at major network number boundaries) in a network address. A routing protocol can summarize on a bit boundary only if it supports *variable-length subnet masks* (VLSMs).

Some routing protocols summarize automatically. Other routing protocols require manual configuration to support route summarization, as shown in Figure 3-3.

Figure 3-3 Route summarization benefits.

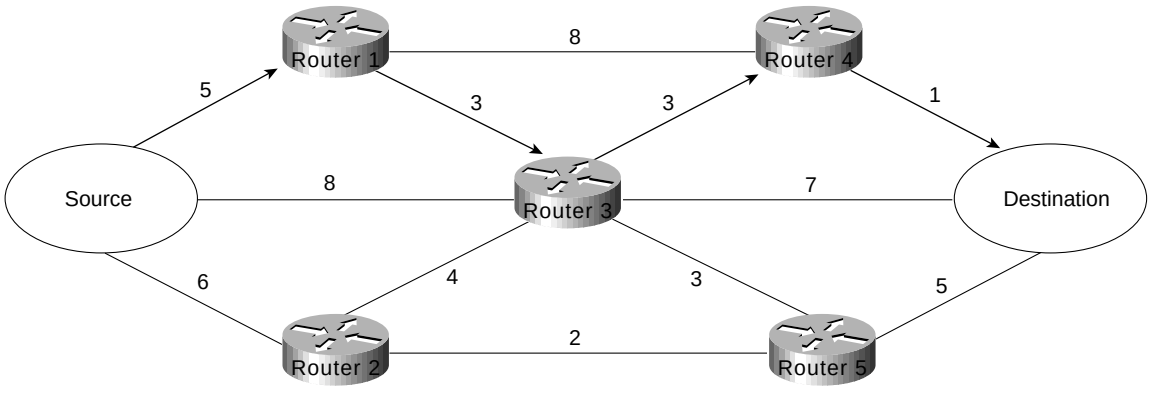


Route Selection

Route selection is trivial when only a single path to the destination exists. However, if any part of that path should fail, there is no way to recover. Therefore, most networks are designed with multiple paths so there are alternatives in case a failure occurs.

Routing protocols compare route metrics to select the best route from a group of possible routes. Route metrics are computed by assigning a characteristic or set of characteristics to each physical network. The metric for the route is an aggregation of the characteristics of each physical network in the route. Figure 3-4 shows a typical meshed network with metrics assigned to each link and the best route from source to destination identified.

Figure 3-4 Routing metrics and route selection.



Routing protocols use different techniques for assigning metrics to individual networks. Further, each routing protocol forms a metric aggregation in a different way. Most routing protocols can use multiple paths if the paths have an equal cost. Some routing protocols can even use multiple paths when paths have an unequal cost. In either case, load balancing can improve overall allocation of network bandwidth.

When multiple paths are used, there are several ways to distribute the packets. The two most common mechanisms are *per-packet load balancing* and *per-destination load balancing*. Per-packet load balancing distributes the packets across the possible routes in a manner proportional to the route metrics. With equal-cost routes, this is equivalent to a round-robin scheme. One packet or destination (depending on switching mode) is distributed to each possible path. Per-destination load balancing distributes packets across the possible routes based on destination. Each new destination is assigned the next available route. This technique tends to preserve packet order.

Note Most TCP implementations can accommodate out-of-order packets. However, out-of-order packets may cause performance degradation.

When fast switching is enabled on a router (default condition), route selection is done on a per-destination basis. When fast switching is disabled, route selection is done on a per-packet basis. For line speeds of 56 Kbps and faster, fast switching is recommended.

Convergence

When *network* topology changes, network traffic must reroute quickly. The phrase "convergence time" describes the time it takes a router to start using a new route after a topology changes. Routers must do three things after a topology changes:

- Detect the change
- Select a new route
- Propagate the changed route information

Some changes are immediately detectable. For example, serial line failures that involve carrier loss are immediately detectable by a router. Other failures are harder to detect. For example, if a serial line becomes unreliable but the carrier is not lost, the unreliable link is not immediately detectable. In addition, some media (Ethernet, for example) do not provide physical indications such as carrier loss. When a router is reset, other routers do not detect this immediately. In general, failure detection is dependent on the media involved and the routing protocol used.

Once a failure has been detected, the routing protocol must select a new route. The mechanisms used to do this are protocol-dependent. All routing protocols must propagate the changed route. The mechanisms used to do this are also protocol-dependent.

Network Scalability

The capability to extend your internetwork is determined, in part, by the scaling characteristics of the routing protocols used and the quality of the network design.

Network scalability is limited by two factors: operational issues and technical issues. Typically, operational issues are more significant than technical issues. Operational scaling concerns encourage the use of large areas or protocols that do not require hierarchical structures. When hierarchical protocols are required, technical scaling concerns promote the use of small areas. Finding the right balance is the art of network design.

From a technical standpoint, routing protocols scale well if their resource use grows less than linearly with the growth of the network. Three critical resources are used by routing protocols: memory, central processing unit (CPU), and bandwidth.

Memory

Routing protocols use memory to store routing tables and topology information. Route summarization cuts memory consumption for all routing protocols. Keeping areas small reduces the memory consumption for hierarchical routing protocols.

CPU

CPU usage is protocol-dependent. Some protocols use CPU cycles to compare new routes to existing routes. Other protocols use CPU cycles to regenerate routing tables after a topology change. In most cases, the latter technique will use more CPU cycles than the former. For link-state protocols, keeping areas small and using summarization reduces CPU requirements by reducing the effect of a topology change and by decreasing the number of routes that must be recomputed after a topology change.

Bandwidth

Bandwidth usage is also protocol-dependent. Three key issues determine the amount of bandwidth a routing protocol consumes:

- *When routing information is sent*—Periodic updates are sent at regular intervals. Flash updates are sent only when a change occurs.
- *What routing information is sent*—Complete updates contain all routing information. Partial updates contain only changed information.
- *Where routing information is sent*—Flooded updates are sent to all routers. Bounded updates are sent only to routers that are affected by a change.

Note These three issues also affect CPU usage.

Distance vector protocols such as Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Internetwork Packet Exchange (IPX) RIP, IPX Service Advertisement Protocol (SAP), and Routing Table Maintenance Protocol (RTMP), broadcast their complete routing table periodically, regardless of whether the routing table has changed. This periodic advertisement varies from every 10 seconds for RTMP to every 90 seconds for IGRP. When the network is stable, distance vector protocols behave well but waste bandwidth because of the periodic sending of routing table updates, even when no change has occurred. When a failure occurs in the network, distance vector protocols do not add excessive load to the network, but they take a long time to reconverge to an alternative path or to flush a bad path from the network.

Link-state routing protocols, such as Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS), and NetWare Link Services Protocol (NLSP), were designed to address the limitations of distance vector routing protocols (slow convergence and unnecessary bandwidth usage). Link-state protocols are more complex than distance vector protocols, and running them adds to the router's overhead. The additional overhead (in the form of memory utilization and bandwidth consumption when link-state protocols first start up) constrains the number of neighbors that a router can support and the number of neighbors that can be in an area.

When the network is stable, link-state protocols minimize bandwidth usage by sending updates only when a change occurs. A hello mechanism ascertains reachability of neighbors. When a failure occurs in the network, link-state protocols flood link-state advertisements (LSAs) throughout an area. LSAs cause every router within the failed area to recalculate routes. The fact that LSAs need to be flooded throughout the area in failure mode and the fact that all routers recalculate routing tables constrain the number of neighbors that can be in an area.

Enhanced IGRP is an advanced distance vector protocol that has some of the properties of link-state protocols. Enhanced IGRP addresses the limitations of conventional distance vector routing protocols (slow convergence and high bandwidth consumption in a steady state network). When the network is stable, Enhanced IGRP sends updates only when a change in the network occurs. Like link-state protocols, Enhanced IGRP uses a hello mechanism to determine the reachability of neighbors. When a failure occurs in the network, Enhanced IGRP looks for feasible successors by sending messages to its neighbors. The search for feasible successors can be aggressive in terms of the traffic it generates (updates, queries, and replies) to achieve convergence. This behavior constrains the number of neighbors that is possible.

In WANs, consideration of bandwidth is especially critical. For example, Frame Relay, which statistically multiplexes many logical data connections (virtual circuits) over a single physical link, allows the creation of networks that share bandwidth. Public Frame Relay networks use bandwidth sharing at all levels within the network. That is, bandwidth sharing may occur within the Frame Relay network of Corporation X, as well as between the networks of Corporation X and Corporation Y.

Two factors have a substantial effect on the design of public Frame Relay networks:

- Users are charged for each permanent virtual circuit (PVC), which encourages network designers to minimize the number of PVCs.
- Public carrier networks sometimes provide incentives to avoid the use of committed information rate (CIR) circuits. Although service providers try to ensure sufficient bandwidth, packets can be dropped.

Overall, WANs can lose packets because of lack of bandwidth. For Frame Relay networks, this possibility is compounded because Frame Relay does not have a broadcast replication facility, so for every broadcast packet that is sent out a Frame Relay interface, the router must replicate it for each PVC on the interface. This requirement limits the number of PVCs that a router can handle effectively.

In addition to bandwidth, network designers must consider the size of routing tables that need to be propagated. Clearly, the design considerations for an interface with 50 neighbors and 100 routes to propagate are very different from the considerations for an interface with 50 neighbors and 10,000 routes to propagate. Table 3-1 gives a rough estimate of the number of WAN neighbors that a routing protocol can handle effectively.

Table 3-1 Routing Protocols and Number of WAN Neighbors

Routing Protocol	Number of Neighbors per Router
Distance vector	50
Link state	30
Advanced distance vector	30

Security

Controlling access to network resources is a primary concern. Some routing protocols provide techniques that can be used as part of a security strategy. With some routing protocols, you can insert a filter on the routes being advertised so that certain routes are not advertised in some parts of the network.

Some routing protocols can authenticate routers that run the same protocol. Authentication mechanisms are protocol specific and generally weak. In spite of this, it is worthwhile to take advantage of the techniques that exist. Authentication can increase network stability by preventing unauthorized routers or hosts from participating in the routing protocol, whether those devices are attempting to participate accidentally or deliberately.

Enhanced IGRP Internetwork Design Guidelines

The Enhanced Interior Gateway Routing Protocol (Enhanced IGRP) is a routing protocol developed by Cisco Systems and introduced with Software Release 9.21 and Cisco Internetworking Operating System (Cisco IOS) Software Release 10.0. Enhanced IGRP combines the advantages of distance vector protocols, such as IGRP, with the advantages of link-state protocols, such as Open Shortest Path First (OSPF). Enhanced IGRP uses the Diffusing Update ALgorithm (DUAL) to achieve convergence quickly.

Enhanced IGRP includes support for IP, Novell NetWare, and AppleTalk. The discussion on Enhanced IGRP covers the following topics:

- Enhanced IGRP Network Topology
- Enhanced IGRP Addressing
- Enhanced IGRP Route Summarization
- Enhanced IGRP Route Selection
- Enhanced IGRP Convergence
- Enhanced IGRP Network Scalability
- Enhanced IGRP Security

Caution If you are using *candidate default route* in IP Enhanced IGRP and have installed multiple releases of Cisco router software within your internetwork that include any versions prior to September 1994, contact your Cisco technical support representative for version compatibility and software upgrade information. Refer to your software release notes for details.

Enhanced IGRP Network Topology

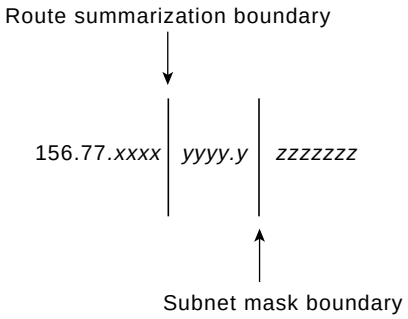
Enhanced IGRP uses a nonhierarchical (or flat) topology by default. Enhanced IGRP automatically summarizes subnet routes of directly connected networks at a network number boundary. This automatic summarization is sufficient for most IP networks. See the section “Enhanced IGRP Route Summarization” later in this chapter for more details.

Enhanced IGRP Addressing

The first step in designing an Enhanced IGRP network is to decide on how to address the network. In many cases, a company is assigned a single NIC address (such as a Class B network address) to be allocated in a corporate internetwork. Bit-wise subnetting and variable-length subnetwork masks (VLSMs) can be used in combination to save address space. Enhanced IGRP for IP supports the use of VLSMs.

Consider a hypothetical network where a Class B address is divided into subnetworks, and contiguous groups of these subnetworks are summarized by Enhanced IGRP. The Class B network 156.77.0.0 might be subdivided as illustrated in Figure 3-5.

Figure 3-5 Variable-length subnet masks (VLSMs) and route summarization boundaries.



In Figure 3-5, the letters x, y, and z represent bits of the last two octets of the Class B network as follows:

- The four *x* bits represent the route summarization boundary.
- The five *y* bits represent up to 32 subnets per summary route.
- The seven *z* bits allow for 126 (128-2) hosts per subnet.

Enhanced IGRP Route Summarization

With Enhanced IGRP, subnet routes of directly connected networks are automatically summarized at network number boundaries. In addition, a network administrator can configure route summarization at any interface with any bit boundary, allowing ranges of networks to be summarized arbitrarily.

Enhanced IGRP Route Selection

Routing protocols compare route metrics to select the best route from a group of possible routes. The following factors are important to understand when designing an Enhanced IGRP internetwork. Enhanced IGRP uses the same vector of metrics as IGRP. Separate metric values are assigned for bandwidth, delay, reliability, and load. By default, Enhanced IGRP computes the metric for a route by using the minimum bandwidth of each hop in the path and adding a media-specific delay for each hop. The metrics used by Enhanced IGRP are as follows:

- *Bandwidth*—Bandwidth is deduced from the interface type. Bandwidth can be modified with the **bandwidth** command.
- *Delay*—Each media type has a propagation delay associated with it. Modifying delay is very useful to optimize routing in network with satellite links. Delay can be modified with the **delay** command.
- *Reliability*—Reliability is dynamically computed as a rolling weighted average over five seconds.
- *Load*—Load is dynamically computed as a rolling weighted average over five seconds.

When Enhanced IGRP summarizes a group of routes, it uses the metric of the best route in the summary as the metric for the summary.

Note For information on Enhanced IGRP load sharing, see the section "SRB Technology Overview and Implementation Issues" in Chapter 4, "Designing SRB Internetworks."

Enhanced IGRP Convergence

Enhanced IGRP implements a new convergence algorithm known as DUAL (Diffusing Update ALgorithm). DUAL uses two techniques that allow Enhanced IGRP to converge very quickly. First, each Enhanced IGRP router stores its neighbors' routing tables. This allows the router to use a new route to a destination instantly if another *feasible* route is known. If no feasible route is known based upon the routing information previously learned from its neighbors, a router running Enhanced IGRP becomes *active* for that destination and sends a query to each of its neighbors, asking for an alternative route to the destination. These queries propagate until an alternative route is found. Routers that are not affected by a topology change remain *passive* and do not need to be involved in the query and response.

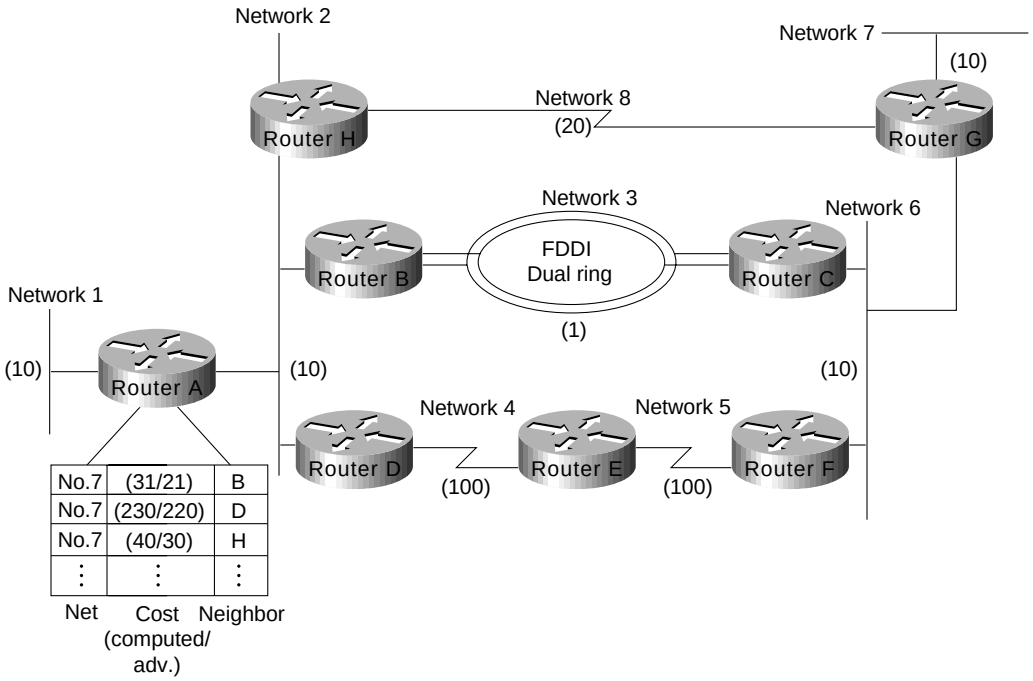
A router using Enhanced IGRP receives full routing tables from its neighbors when it first communicates with the neighbors. Thereafter, only *changes* to the routing tables are sent and only to *routers* that are *affected* by the change. A *successor* is a neighboring router that is currently being used for packet forwarding, provides the *least cost* route to the destination, and is not part of a routing loop. Information in the routing table is based on *feasible successors.* Feasible successor routes can be used in case the existing route fails. Feasible successors provide the *next least-cost* path without introducing routing loops.

The routing table keeps a list of the computed costs of reaching networks. The topology table keeps a list of all routes advertised by neighbors. For each network, the router keeps the real cost of getting to that network and also keeps the advertised cost from its neighbor. In the event of a failure, convergence is instant if a feasible successor can be found. A neighbor is a feasible successor if it meets the feasibility condition set by DUAL. DUAL finds feasible successors by the performing the following computations:

- Determines membership of V_1. V_1 is the set of all neighbors whose advertised distance to network *x* is less than FD. (FD is the feasible distance and is defined as the best metric during an active-to-passive transition.)
- Calculates $D_{min.}$ D_{min} is the minimum computed cost to network *x*.
- Determines membership of V_2. V_2 is the set of neighbors that are in V1 whose computed cost to network *x* equals D_{min}.

The feasibility condition is met when V_2 has one or more members. The concept of feasible successors is illustrated in Figure 3-6. Consider Router A's topology table entries for Network 7. Router B is the *successor* with a computed cost of 31 to reach Network 7, compared to the computed costs of Router D (230) and Router H (40).

Figure 3-6 DUAL feasible successor.



If Router B becomes unavailable, Router A will go through the following three-step process to find a feasible successor for Network 7:

Step 1 Determining which neighbors have an advertised distance to Network 7 that is less than Router A's feasible distance (FD) to Network 7. The FD is 31 and Router H meets this condition. Therefore, Router H is a member of V_1.

Step 2 Calculating the minimum computed cost to Network 7. Router H provides a cost of 40, and Router D provides a cost of 230. D_{min} is, therefore, 40.

Step 3 Determining the set of neighbors that are in V_1 whose computed cost to Network 7 equals D_{min} (40). Router H meets this condition.

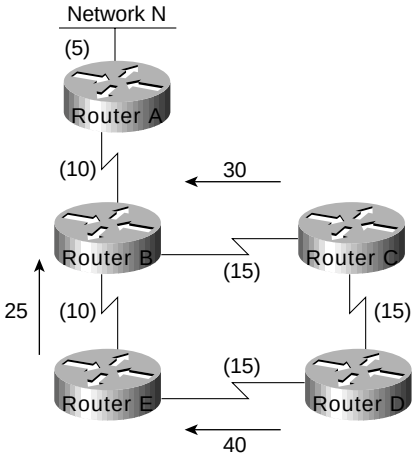
The feasible successor is Router H which provides a least cost route of 40 from Router A to Network 7. If Router H now also becomes unavailable, Router A performs the following computations:

Step 1 Determines which neighbors have an advertised distance to Network 7 that is less than the FD for Network 7. Because both Router B and H have become unavail- able, only Router D remains. However, the advertised cost of Router D to Network 7 is 220, which is greater than Router A's FD (31) to Network 7. Router D, therefore, cannot be a member of V_1. The FD remains at 31—the FD can only change during an active-to-passive transition, and this did not occur. There was no transition to active state for Network 7; this is known as a *local computation*.

Step 2 Because there are no members of V1, there can be no feasible successors. Router A, therefore, transitions from passive to active state for Network 7 and queries its neighbors about Network 7. There was a transition to active; this is known as a *diffusing computation*.

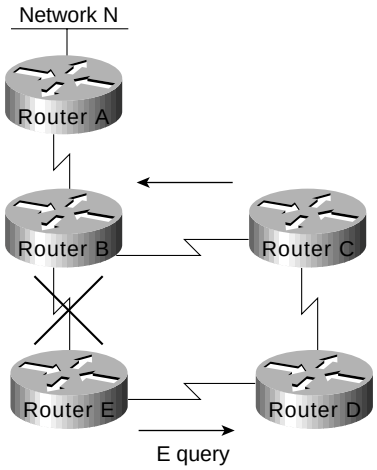
The following example and graphics further illustrate how Enhanced IGRP supports virtually instantaneous convergence in a changing internetwork environment. In Figure 3-7, all routers can access one another and Network N. The computed cost to reach other routers and Network N is shown. For example, the cost from Router E to Router B is 10. The cost from Router E to Network N is 25 (cumulative of 10 + 10 + 5 = 25).

Figure 3-7 DUAL example (part 1): initial network connectivity.



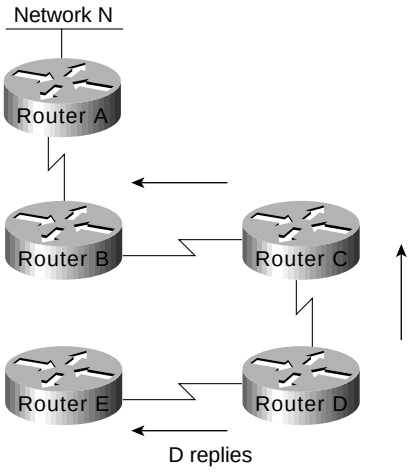
In Figure 3-8, the connection between Router B and Router E fails. Router E sends a multicast query to all of its neighbors and puts Network N into an active state.

Figure 3-8 DUAL example (part 2): sending queries.



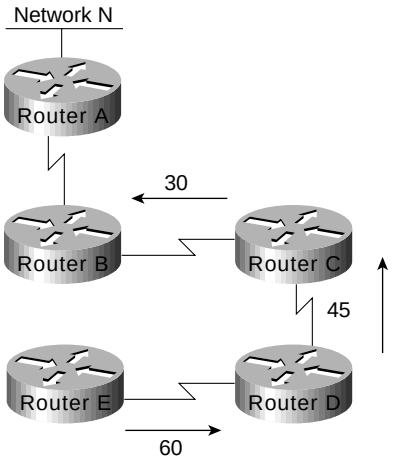
Next, as illustrated in Figure 3-9, Router D determines that it has a feasible successor. It changes its successor from Router E to Router C and sends a reply to Router E.

Figure 3-9 UAL example (part 3): switching to a feasible successor.



In Figure 3-10, Router E has received replies from all neighbors and therefore brings Network N out of active state. Router E puts Network N into its routing table at a distance of 60.

Figure 3-10 Flow of intersubnet traffic with layer 3 switches.



Note Router A, Router B, and Router C were not involved in route recomputation. Router D recomputed its path to Network N without first needing to learn new routing information from its downstream neighbors.

Enhanced IGRP Network Scalability

Network scalability is limited by two factors: operational issues and technical issues. Operationally, Enhanced IGRP provides easy configuration and growth. Technically, Enhanced IGRP uses resources at less than a linear rate with the growth of a network.

Memory

A router running Enhanced IGRP stores all routes advertised by neighbors so that it can adapt quickly to alternative routes. The more neighbors a router has, the more memory a router uses. Enhanced IGRP automatic route aggregation bounds the routing table growth naturally. Additional bounding is possible with manual route aggregation.

CPU

Enhanced IGRP uses the DUAL algorithm to provide fast convergence. DUAL recomputes only routes which are affected by a topology change. DUAL is not computationally complex, so it does not require a lot of CPU.

Bandwidth

Enhanced IGRP uses partial updates. Partial updates are generated only when a change occurs; only the changed information is sent, and this changed information is sent only to the routers affected. Because of this, Enhanced IGRP is very efficient in its usage of bandwidth. Some additional bandwidth is used by Enhanced IGRP's HELLO protocol to maintain adjacencies between neighboring routers.

Enhanced IGRP Security

Enhanced IGRP is available only on Cisco routers. This prevents accidental or malicious routing disruption caused by hosts in a network. In addition, route filters can be set up on any interface to prevent learning or propagating routing information inappropriately.

OSPF Internetwork Design Guidelines

OSPF is an Interior Gateway Protocol (IGP) developed for use in Internet Protocol (IP)-based internetworks. As an IGP, OSPF distributes routing information between routers belonging to a single autonomous system (AS). An AS is a group of routers exchanging routing information via a common routing protocol. The OSPF protocol is based on shortest-path-first, or link-state, technology.

The OSPF protocol was developed by the OSPF working group of the Internet Engineering Task Force (IETF). It was designed expressly for the Internet Protocol (IP) environment, including explicit support for IP subnetting and the tagging of externally derived routing information. OSPF Version 2 is documented in Request for Comments (RFC) 1247.

Whether you are building an OSPF internetwork from the ground up or converting your internetwork to OSPF, the following design guidelines provide a foundation from which you can construct a reliable, scalable OSPF-based environment.

Two design activities are critically important to a successful OSPF implementation:

- Definition of area boundaries
- Address assignment

Ensuring that these activities are properly planned and executed will make all the difference in your OSPF implementation. Each is addressed in more detail with the discussions that follow. These discussions are divided into nine sections:

- OSPF Network Topology
- OSPF Addressing and Route Summarization

- OSPF Route Selection
- OSPF Convergence
- OSPF Network Scalability
- OSPF Security
- OSPF NSSA (Not-So-Stubby Area) Capabilities
- OSPF On Demand Circuit Protocol Issues
- OSPF over Non-Broadcast Networks

OSPF Network Topology

OSPF works best in a hierarchical routing environment. The first and most important decision when designing an OSPF network is to determine which routers and links are to be included in the backbone and which are to be included in each area. There are several important guidelines to consider when designing an OSPF topology:

- *The number of routers in an area*—OSPF uses a CPU-intensive algorithm. The number of calculations that must be performed given *n* link-state packets is proportional to *n log n*. As a result, the larger and more unstable the area, the greater the likelihood for performance problems associated with routing protocol recalculation. Generally, an area should have no more than 50 routers. Areas with unstable links should be smaller.
- *The number of neighbors for any one router*—OSPF floods all link-state changes to all routers in an area. Routers with many neighbors have the most work to do when link-state changes occur. In general, any one router should have no more than 60 neighbors.
- *The number of areas supported by any one router*—A router must run the link-state algorithm for each link-state change that occurs for every area in which the router resides. Every area border router is in at least two areas (the backbone and one area). In general, to maximize stability, one router should not be in more than three areas.
- *Designated router selection*—In general, the designated router and backup designated router on a local-area network (LAN) have the most OSPF work to do. It is a good idea to select routers that are not already heavily loaded with CPU-intensive activities to be the designated router and backup designated router. In addition, it is generally not a good idea to select the same router to be designated router on many LANs simultaneously.

The discussions that follow address topology issues that are specifically related to the backbone and the areas.

Backbone Considerations

Stability and *redundancy* are the most important criteria for the backbone. Stability is increased by keeping the size of the backbone reasonable. This is caused by the fact that every router in the backbone needs to recompute its routes after every link-state change. Keeping the backbone small reduces the likelihood of a change and reduces the amount of CPU cycles required to recompute routes. As a general rule, each area (including the backbone) should contain no more than 50 routers. If link quality is high and the number of routes is small, the number of routers can be increased. Redundancy is important in the backbone to prevent partition when a link fails. Good backbones are designed so that no single link failure can cause a partition.

OSPF backbones must be contiguous. All routers in the backbone should be directly connected to other backbone routers. OSPF includes the concept of virtual links. A virtual link creates a path between two area border routers (an area border router is a router connects an area to the backbone)

that are not directly connected. A virtual link can be used to heal a partitioned backbone. However, it is not a good idea to design an OSPF network to require the use of virtual links. The stability of a virtual link is determined by the stability of the underlying area. This dependency can make troubleshooting more difficult. In addition, virtual links cannot run across stub areas. See the section "Backbone-to-Area Route Advertisement" later in this chapter for a detailed discussion of stub areas.

Avoid placing hosts (such as workstations, file servers, or other shared resources) in the backbone area. Keeping hosts out of the backbone area simplifies internetwork expansion and creates a more stable environment.

Area Considerations

Individual areas must be contiguous. In this context, a contiguous area is one in which a continuous path can be traced from any router in an area to any other router in the same area. This does not mean that all routers must share common network media. It is not possible to use virtual links to connect a partitioned area. Ideally, areas should be richly connected internally to prevent partitioning. The two most critical aspects of area design follow:

- Determining how the area is addressed
- Determining how the area is connected to the backbone

Areas should have a contiguous set of network and/or subnet addresses. Without a contiguous address space, it is not possible to implement route summarization. The routers that connect an area to the backbone are called *area border routers*. Areas can have a single area border router or they can have multiple area border routers. In general, it is desirable to have more than one area border router per area to minimize the chance of the area becoming disconnected from the backbone.

When creating large-scale OSPF internetworks, the definition of areas and assignment of resources within areas must be done with a pragmatic view of your internetwork. The following are general rules that help ensure that your internetwork remains flexible and provides the kind of performance needed to deliver reliable resource access:

- *Consider physical proximity when defining areas*—If a particular location is densely connected, create an area specifically for nodes at that location.
- *Reduce the maximum size of areas if links are unstable*—If your internetwork includes unstable links, consider implementing smaller areas to reduce the effects of route flapping. Whenever a route is lost or comes online, each affected area must converge on a new topology. The Dykstra algorithm will run on all the affected routers. By segmenting your internetwork into smaller areas, you can isolate unstable links and deliver more reliable overall service.

OSPF Addressing and Route Summarization

Address assignment and route summarization are inextricably linked when designing OSPF internetworks. To create a scalable OSPF internetwork, you should implement route summarization. To create an environment capable of supporting route summarization, you must implement an effective hierarchical addressing scheme. The addressing structure that you implement can have a profound impact on the performance and scalability of your OSPF internetwork. The following sections discuss OSPF route summarization and three addressing options:

- Separate network numbers for each area
- Network Information Center (NIC)-authorized address areas created using bit-wise subnetting and VLSM
- Private addressing, with a *demilitarized zone* (DMZ) buffer to the official Internet world

Note You should keep your addressing scheme as simple as possible, but be wary of oversimplifying your address assignment scheme. Although simplicity in addressing saves time later when operating and troubleshooting your network, taking shortcuts can have certain severe consequences. In building a scalable addressing environment, use a structured approach. If necessary, use bit-wise subnetting— but make sure that route summarization can be accomplished at the area border routers.

OSPF Route Summarization

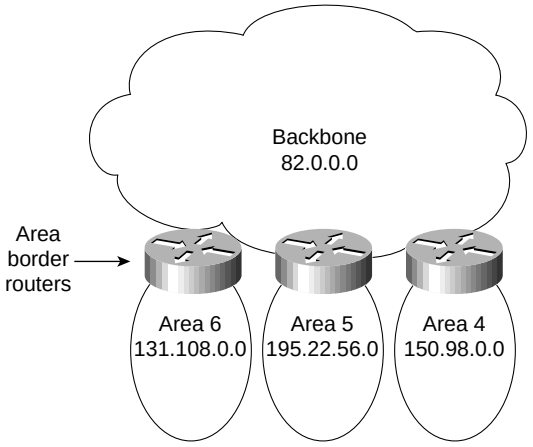
Route summarization is extremely desirable for a reliable and scalable OSPF internetwork. The effectiveness of route summarization, and your OSPF implementation in general, hinges on the addressing scheme that you adopt. Summarization in an OSPF internetwork occurs between each area and the backbone area. Summarization must be configured manually in OSPF. When planning your OSPF internetwork, consider the following issues:

- Be sure that your network addressing scheme is configured so that the range of subnets assigned within an area is contiguous.
- Create an address space that will permit you to split areas easily as your network grows. If possible, assign subnets according to simple octet boundaries. If you cannot assign addresses in an easy-to-remember and easy-to-divide manner, be sure to have a thoroughly defined addressing structure. If you know how your entire address space is assigned (or will be assigned), you can plan for changes more effectively.
- Plan ahead for the addition of new routers to your OSPF environment. Be sure that new routers are inserted appropriately as area, backbone, or border routers. Because the addition of new routers creates a new topology, inserting new routers can cause unexpected routing changes (and possibly performance changes) when your OSPF topology is recomputed.

Separate Address Structures for Each Area

One of the simplest ways to allocate addresses in OSPF is to assign a separate network number for each area. With this scheme, you create a backbone and multiple areas, and assign a separate IP network number to each area. Figure 3-11 illustrates this kind of area allocation.

Figure 3-11 Assignment of NIC addresses example.



The following are the basic steps for creating such a network:

Step 1 Define your structure (identify areas and allocate nodes to areas).

Step 2 Assign addresses to networks, subnets, and end stations.

In the network illustrated in Figure 3-11, each area has its own unique NIC-assigned address. These can be Class A (the backbone in Figure 3-11), Class B (areas 4 and 6), or Class C (Area 5). The following are some clear benefits of assigning separate address structures to each area:

- Address assignment is relatively easy to remember.
- Configuration of routers is relatively easy and mistakes are less likely.
- Network operations are streamlined because each area has a simple, unique network number.

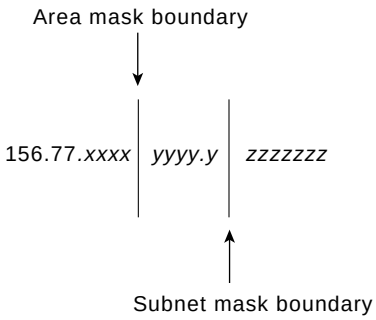
In the example illustrated in Figure 3-11, the route summarization configuration at the area border routers is greatly simplified. Routes from Area 4 injecting into the backbone can be summarized as follows: *All routes starting with 150.98 are found in Area 4*.

The main drawback of this approach to address assignment is that it wastes address space. If you decide to adopt this approach, be sure that area border routers are configured to do route summarization. Summarization must be explicitly set; it is disabled by default in OSPF.

Bit-Wise Subnetting and VLSM

Bit-wise subnetting and variable-length subnetwork masks (VLSMs) can be used in combination to save address space. Consider a hypothetical network where a Class B address is subdivided using an area mask and distributed among 16 areas. The Class B network, 156.77.0.0, might be sub- divided as illustrated in Figure 3-12.

Figure 3-12 Areas and subnet masking.



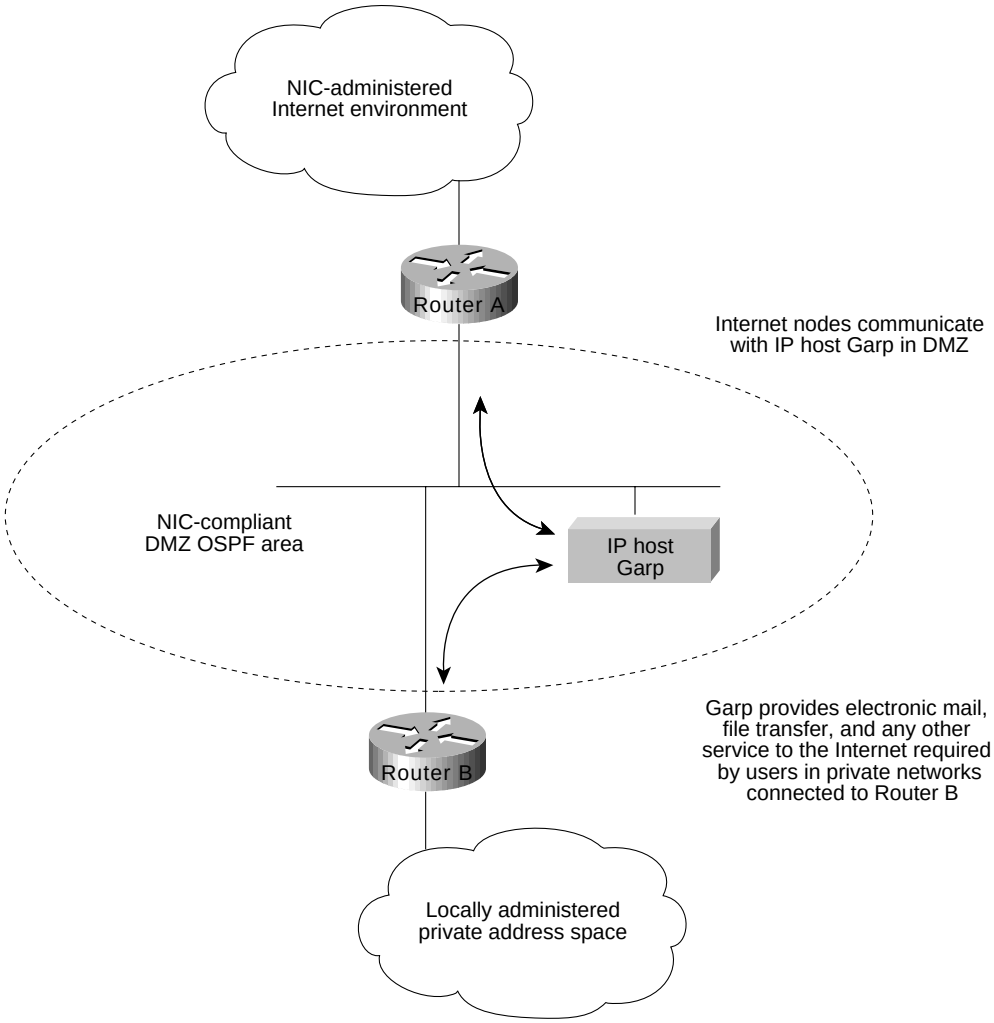
In Figure 3-12, the letters *x*, *y*, and *z* represent bits of the last two octets of the Class B network as follows:

- The four *x* bits are used to identify 16 areas.
- The five *y* bits represent up to 32 subnets per area.
- The seven *z* bits allow for 126 (128-2) hosts per subnet.Private Addressing

Private addressing is another option often cited as simpler than developing an area scheme using bit-wise subnetting. Although private address schemes provide an excellent level of flexibility and do not limit the growth of your OSPF internetwork, they have certain disadvantages. For instance, developing a large-scale internetwork of privately addressed IP nodes limits total access to the Internet, and mandates the implementation of what is referred to as a *demilitarized zone* (DMZ). If you need to connect to the Internet, Figure 3-13 illustrates the way in which a DMZ provides a buffer of valid NIC nodes between a privately addressed network and the Internet.

All nodes (end systems and routers) on the network in the DMZ must have NIC-assigned IP addresses. The NIC might, for example, assign a single Class C network number to you. The DMZ shown in Figure 3-13 has two routers and a single application gateway host (Garp). Router A provides the interface between the DMZ and the Internet, and Router B provides the firewall between the DMZ and the private address environment. All applications that need to run over the Internet must access the Internet through the application gateway.

Figure 3-13 Connecting to the Internet from a privately addressed network.



Route Summarization Techniques

Route summarization is particularly important in an OSPF environment because it increases the stability of the network. If route summarization is being used, routes within an area that change do not need to be changed in the backbone or in other areas. Route summarization addresses two important questions of route information distribution:

- What information does the backbone need to know about each area? The answer to this question focuses attention on area-to-backbone routing information.
- What information does each area need to know about the backbone and other areas? The answer to this question focuses attention on backbone-to-area routing information.

Area-to-Backbone Route Advertisement

There are several key considerations when setting up your OSPF areas for proper summarization:

- OSPF route summarization occurs in the area border routers.
- OSPF supports VLSM, so it is possible to summarize on any bit boundary in a network or subnet address.
- OSPF requires manual summarization. As you design the areas, you need to determine summarization at each area border router.

Backbone-to-Area Route Advertisement

There are four potential types of routing information in an area:

- *Default*—If an explicit route cannot be found for a given IP network or subnetwork, the router will forward the packet to the destination specified in the default route.
- *Intra-area routes*—Explicit network or subnet routes must be carried for all networks or subnets inside an area.
- *Interarea routes*—Areas may carry explicit network or subnet routes for networks or subnets that are in this AS but not in this area.
- *External routes*—When different ASs exchange routing information, the routes they exchange are referred to as external routes.

In general, it is desirable to restrict routing information in any area to the minimal set that the area needs. There are three types of areas, and they are defined in accordance with the routing information that is used in them:

- *Nonstub areas*—Nonstub areas carry a default route, static routes, intra-area routes, interarea routes, and external routes. An area must be a nonstub area when it contains a router that uses both OSPF and any other protocol, such as the Routing Information Protocol (RIP). Such a router is known as an autonomous system border router (ASBR). An area must also be a nonstub area when a virtual link is configured across the area. Nonstub areas are the most resource-intensive type of area.
- *Stub areas*—Stub areas carry a default route, intra-area routes and interarea routes, but they do not carry external routes. Stub areas are recommended for areas that have only one area border router and they are often useful in areas with multiple area border routers. See “Controlling Interarea Traffic” later in this chapter for a detailed discussion of the design trade-offs in areas with multiple area border routers.There are two restrictions on the use of stub areas: Virtual links cannot be configured across them and they cannot contain an ASBR.
- *Stub areas without summaries*—Software releases 9.1(11), 9.21(2), and 10.0(1) and later support stub areas without summaries, allowing you to create areas that carry only a default route and intra-area routes. Stub areas without summaries do not carry interarea routes or external routes. This type of area is recommended for simple configurations in which a single router connects an area to the backbone.

Table 3-2 shows the different types of areas according to the routing information that they use.

Routing Information Used in OSPF Areas

Area Type	Default Route	Intra-area Routes	Interarea Routes	External Routes
Nonstub	Yes	Yes	Yes	Yes
Stub	Yes	Yes	Yes	No
Stub without summaries	Yes	Yes	No	No

Stub areas are configured using the **area** *area-id* **stub** router configuration command. Routes are summarized using the **area** *area-id* **range** *address mask* router configuration command. Refer to your *Router Products Configuration Guide* and *Router Products Command Reference* publications for more information regarding the use of these commands.

OSPF Route Selection

When designing an OSPF internetwork for efficient route selection, consider three important topics:

- Tuning OSPF Metrics
- Controlling Interarea Traffic
- Load Balancing in OSPF Internetworks

Tuning OSPF Metrics

The default value for OSPF metrics is based on bandwidth. The following characteristics show how OSPF metrics are generated:

- Each link is given a metric value based on its bandwidth. The metric for a specific link is the inverse of the bandwidth for that link. Link metrics are normalized to give FDDI a metric of 1. The metric for a route is the sum of the metrics for all the links in the route.

Note In some cases, your network might implement a media type that is faster than the fastest default media configurable for OSPF (FDDI). An example of a faster media is ATM. By default, a faster media will be assigned a cost equal to the cost of an FDDI link—a link-state metric cost of 1. Given an environment with both FDDI and a faster media type, you must manually configure link costs to configure the faster link with a lower metric. Configure any FDDI link with a cost greater than 1, and the faster link with a cost less than the assigned FDDI link cost. Use the **ip ospf cost** interface configuration command to modify link-state cost.

- When route summarization is enabled, OSPF uses the metric of the best route in the summary.
- There are two forms of external metrics: type 1 and type 2. Using an external type 1 metric results in routes adding the internal OSPF metric to the external route metric. External type 2 metrics do not add the internal metric to external routes. The external type 1 metric is generally preferred. If you have more than one external connection, either metric can affect how multiple paths are used.

Controlling Interarea Traffic

When an area has only a single area border router, all traffic that does not belong in the area will be sent to the area border router. In areas that have multiple area border routers, two choices are available for traffic that needs to leave the area:

- Use the area border router closest to the originator of the traffic. (Traffic leaves the area as soon as possible.)
- Use the area border router closest to the destination of the traffic. (Traffic leaves the area as late as possible.)

If the area border routers inject only the default route, the traffic goes to the area border router that is closest to the source of the traffic. Generally, this behavior is desirable because the backbone typically has higher bandwidth lines available. However, if you want the traffic to use the area border router that is nearest the destination (so that traffic leaves the area as late as possible), the area border routers should inject summaries into the area instead of just injecting the default route.

Most network designers prefer to avoid asymmetric routing (that is, using a different path for packets that are going from A to B than for those packets that are going from B to A). It is important to understand how routing occurs between areas to avoid asymmetric routing.

Load Balancing in OSPF Internetworks

Internetwork topologies are typically designed to provide redundant routes in order to prevent a partitioned network. Redundancy is also useful to provide additional bandwidth for high traffic areas. If equal-cost paths between nodes exist, Cisco routers automatically load balance in an OSPF environment.

Cisco routers can use up to four equal-cost paths for a given destination. Packets might be distributed either on a per-destination (when fast switching) or a per-packet basis. Per-destination load balancing is the default behavior. Per-packet load balancing can be enabled by turning off fast switching using the **no ip route-cache** interface configuration command. For line speeds of 56 Kbps and faster, it is recommended that you enable fast switching.

OSPF Convergence

One of the most attractive features about OSPF is the capability to quickly adapt to topology changes. There are two components to routing convergence:

- *Detection of topology changes*—OSPF uses two mechanisms to detect topology changes. Interface status changes (such as carrier failure on a serial link) is the first mechanism. The second mechanism is failure of OSPF to receive a hello packet from its neighbor within a timing window called a *dead timer.* After this timer expires, the router assumes the neighbor is down. The dead timer is configured using the **ip ospf dead-interval** interface configuration command. The default value of the dead timer is four times the value of the Hello interval. That results in a dead timer default of 40 seconds for broadcast networks and two minutes for nonbroadcast networks.
- *Recalculation of routes*—After a failure has been detected, the router that detected the failure sends a link-state packet with the change information to all routers in the area. All the routers recalculate all of their routes using the Dykstra (or SPF) algorithm. The time required to run the algorithm depends on a combination of the size of the area and the number of routes in the database.

OSPF Network Scalability

Your ability to scale an OSPF internetwork depends on your overall network structure and addressing scheme. As outlined in the preceding discussions concerning network topology and route summarization, adopting a hierarchical addressing environment and a structured address assignment will be the most important factors in determining the scalability of your internetwork. Network scalability is affected by operational and technical considerations:

- Operationally, OSPF networks should be designed so that areas do not need to be split to accommodate growth. Address space should be reserved to permit the addition of new areas.
- Technically, scaling is determined by the utilization of three resources: memory, CPU, and bandwidth, all discussed in the following sections.

Memory

An OSPF router stores all of the link states for all of the areas that it is in. In addition, it can store summaries and externals. Careful use of summarization and stub areas can reduce memory use substantially.

CPU

An OSPF router uses CPU cycles whenever a link-state change occurs. Keeping areas small and using summarization dramatically reduces CPU use and creates a more stable environment for OSPF.

Bandwidth

OSPF sends partial updates when a link-state change occurs. The updates are flooded to all routers in the area. In a quiet network, OSPF is a quiet protocol. In a network with substantial topology changes, OSPF minimizes the amount of bandwidth used.

OSPF Security

Two kinds of security are applicable to routing protocols:

- *Controlling the routers that participate in an OSPF network*

  OSPF contains an optional authentication field. All routers within an area must agree on the value of the authentication field. Because OSPF is a standard protocol available on many platforms, including some hosts, using the authentication field prevents the inadvertent startup of OSPF in an uncontrolled platform on your network and reduces the potential for instability.

- *Controlling the routing information that routers exchange*

  All routers must have the same data within an OSPF area. As a result, it is not possible to use route filters in an OSPF network to provide security.

OSPF NSSA (Not-So-Stubby Area) Overview

Prior to NSSA, to disable an area from receiving external (Type 5) link-state advertisements (LSAs), the area needed to be defined as a stub area. Area Border Routers (ABRs) that connect stub areas do not flood any external routes they receive into the stub areas. To return packets to destinations outside of the stub area, a default route through the ABR is used.

RFC 1587 defines a hybrid area called the Not-So-Stubby Area (NSSA). An OSPF NSSA is similar to an OSPF stub area but allows for the following capabilities:

- Importing (redistribution) of external routes as Type 7 LSAs into NSSAs by NSSA Autonomous System Boundary Routers (ASBRs).
- Translation of specific Type 7 LSAs routes into Type 5 LSAs by NSSA ABRs.

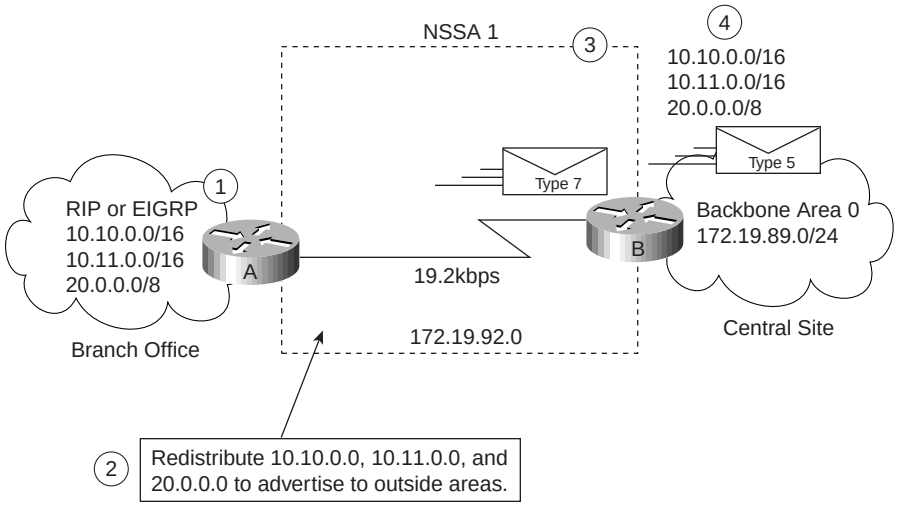
Using OSPF NSSA

Use OSPF NSSA in the following scenarios:

- When you want to summarize or filter Type 5 LSAs before they are forwarded into an OSPF area. The OSPF Specification (RFC 1583) prohibits the summarizing or filtering of Type 5 LSAs. It is an OSPF requirement that Type 5 LSAs always be flooding throughout a routing domain. When you define an NSSA, you can import specific external routes as Type 7 LSAs into the NSSA. In addition, when translating Type 7 LSAs to be imported into nonstub areas, you can summarize or filter the LSAs before importing them as Type 5 LSAs.
- If you are an Internet service provider (ISP) or a network administrator that has to connect a central site using OSPF to a remote site that is using a different protocol, such as RIP or EIGRP, you can use NSSA to simplify the administration of this kind of topology. Prior to NSSA, the connection between the corporate site ABR and the remote router used RIP or EIGRP. This meant maintaining two routing protocols. Now, with NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA, as shown in Figure 3-14. You cannot expand the normal OSPF area to the remote site because the Type 5 external will overwhelm both the slow link and the remote router.

In Figure 3-14, the central site and branch office are interconnected through a slow WAN link. The branch office is not using OSPF, but the central site is. Rather than define an RIP domain to connect the sites, you can define an NSSA.

Figure 3-14 OSPF NSSA operation.



In this scenario, Router A is defined as an ASBR (autonomous system border router). It is configured to redistribute any routes within the RIP/EIGRP domain to the NSSA. The following lists what happens when the area between the connecting routers is defined as an NSSA:

1 Router A receives RIP or EGRP routes for networks 10.10.0.0/16, 10.11.0.0/16, and 20.0.0.0/8.

2. Because Router A is also connected to an NSSA, it redistributes the RIP or EIGRP routers as Type 7 LSAs into the NSSA.
3. Router B, an ABR between the NSSA and the backbone Area 0, receives the Type 7 LSAs.
4. After the SPF calculation on the forwarding database, Router B translates the Type 7 LSAs into Type 5 LSAs and then floods them throughout Backbone Area 0. It is at this point that router B could have summarized routes 10.10.0.0/16 and 10.11.0.0/16 as 10.0.0.0/8, or could have filtered one or more of the routes.

Type 7 LSA Characteristics

Type 7 LSAs have the following characteristics:

- They are originated only by ASBRs that are connected between the NSSA and autonomous system domain.
- They include a forwarding address field. This field is retained when a Type 7 LSA is translated as a Type 5 LSA.
- They are advertised only within an NSSA.
- They are not flooded beyond an NSSA. The ABR that connects to another nonstub area reconverts the Type 7 LSA into a Type 5 LSA before flooding it.
- NSSA ABRs can be configured to summarize or filter Type 7 LSAs into Type 5 LSAs.
- NSSA ABRs can advertise a Type 7 default route into the NSSA.
- Type 7 LSAs have a lower priority than Type 5 LSAs, so when a route is learned with a Type 5 LSA and Type 7 LSA, the route defined in the Type 5 LSA will be selected first.

Configuring OSPF NSSA

The steps used to configure OSPF NSSA are as follows:

Step 1 Configure standard OSPF operation on one or more interfaces that will be attached to NSSAs.

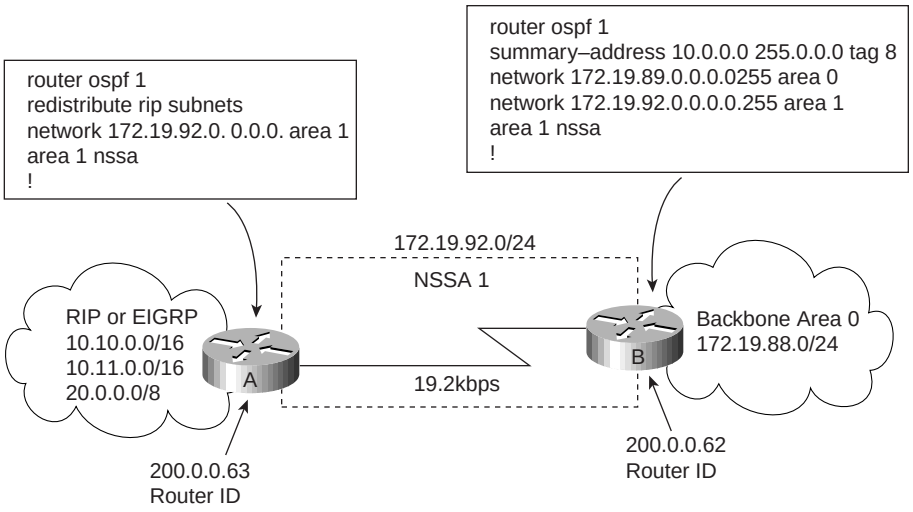
Step 2 Configure an area as NSSA using the following commands:

```
router(config)#area area-id nssa
```

Step 3 (Optional) Control the summarization or filtering during the translation. Figure 3-15 shows how Router will summarize routes using the following command:

```
router(config)#summary-address prefix mask [not-advertise] [tag tag]
```

Figure 3-15 Configuring OSPF NSSA.



NSSA Implementation Considerations

Be sure to evaluate these considerations before implementing NSSA. As shown in Figure 3-15, you can set a Type 7 default route that can be used to reach external destinations. The command to issue a Type 7 default route is as follows:

```
router(config)#area area-id nssa [default-information-originate]
```

When configured, the router generates a Type 7 default into the NSSA by the NSSA ABR. Every router within the same area must agree that the area is NSSA; otherwise, the routers will not be able to communicate with one another.

If possible, avoid doing explicit redistribution on NSSA ABR because you could get confused about which packets are being translated by which router.

OSPF On Demand Circuit

OSPF On Demand Circuit is an enhancement to the OSPF protocol that allows efficient operation over on-demand circuits such as ISDN, X.25 SVCs, and dial-up lines. This feature supports RFC 1793, OSPF Over On Demand Circuits. This RFC is useful in understanding the operation of this feature. It has good examples and explains the operation of OSPF in this type of environment.

Prior to this feature, OSPF periodic Hello and link-state advertisement (LSA) updates would be exchanged between routers that connected the on-demand link even when there were no changes in the Hello or LSA information.

With OSPF On Demand Circuit, periodic Hellos are suppressed and periodic refreshes of LSAs are not flooded over demand circuits. These packets bring up the links only when they are exchanged for the first time, or when there is a change in the information they contain. This operation allows the underlying data link layer to be closed when the network topology is stable, thus keeping the cost of the demand circuit to a minimum.

This feature is a standards-based mechanism that is similar to the Cisco Snapshot feature used for distance vector protocols such as RIP.

Why Use OSPF On Demand Circuit?

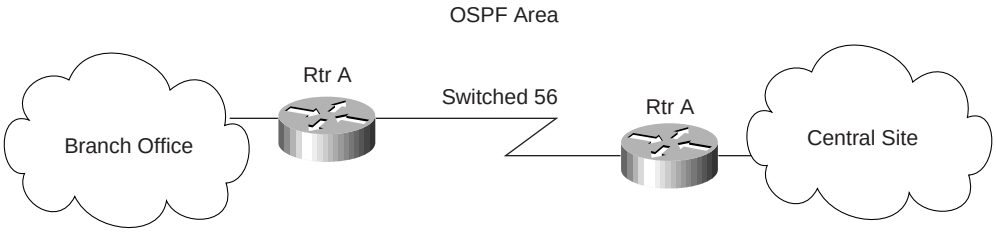
This feature is useful when you want to have an OSPF backbone at the central site and you want to connect telecommuters or branch offices to the central site. In this case, OSPF On Demand Circuit allows the benefits of OSPF over the entire domain without excessive connection costs. Periodic refreshes of Hello updates and LSA updates and other protocol overhead are prevented from enabling the on-demand circuit when there is no "real" data to transmit.

Overhead protocols such as Hellos and LSAs are transferred over the on-demand circuit only upon initial setup and when they reflect a change in the topology. This means that topology-critical changes that require new shortest path first (SPF) calculations are transmitted in order to maintain network topology integrity, but periodic refreshes that do not include changes are not transmitted across the link.

OSPF On Demand Circuit Operation

Figure 3-16 illustrates general OSPF operation over on-demand circuits.

Figure 3-16 OSPF area.



The following steps describe the procedure shown in Figure 3-16:

1. Upon initialization, Router A brings up the on demand circuit to exchange Hellos and synchronize LSA databases with Router B. Because both routers are configured for OSPF On Demand Circuit, each router's Hello packets and database description packets have the demand circuit (DC) bit set. As a result, both routers know to suppress periodic Hello packet updates. When each router floods LSAs over the network, the LSAs will have the DoNotAge (DNA) bit set. This means that the LSAs will not age. They can be updated if a new LSA is received with changed information, but no periodic LSA refreshes will be issued over the demand circuit.
2. When Router A receives refreshed LSAs for existing entries in its database, it will determine whether the LSAs include changed information. If not, Router A will update the existing LSA entries, but it will not flood the information to Router B. Therefore, both routers will have the same entries, but the entry sequence numbers may not be identical.
3. When Router A does receive an LSA for a new route or an LSA that includes changed information, it will update its LSA database, bring up the on-demand circuit, and flood the information to Router B. At this point, both routers will have identical sequence numbers for this LSA entry.
4. If there is no data to transfer while the link is up for the updates, the link is terminated.
5. When a host on either side needs to transfer data to another host at the remote site, the link will be brought up.

Configuring OSPF On Demand Circuit

The steps used to configure OSPF On Demand Circuit are summarized as follows:

Step 1 Configure your on-demand circuit. For example:

```
interface bri 0
ip address 10.1.1.1 255.255.255.0
encapsulation ppp
dialer idle-timeout 3600
dialer map ip name rtra 10.1.1.2 broadcast 1234
dialer group 1
ppp authentication chap
dialer list 1 protocol ip permit
```

Step 2 Enable OSPF operation, as follows:

```
router(config)#router ospf process-id
```

Step 3 Configure OSPF on an on-demand circuit using the following interface command:

```
interface bri 0
ip ospf demand-circuit
```

If the router is part of a point-to-point topology, only one end of the demand circuit needs to be configured with this command, but both routers need to have this feature loaded. All routers that are part of a point-to-multipoint topology need to be configured with this command.

Implementation Considerations for OSPF On Demand Circuit

Evaluate the following considerations before implementing OSPF On Demand Circuit:

1. Because LSAs indicating topology changes are flooded over an on-demand circuit, you are advised to put demand circuits within OSPF stub areas or within NSSAs to isolate the demand circuits from as many topology changes as possible.
2. To take advantage of the on-demand circuit functionality within a stub area or NSSA, every router in the area must have this feature loaded. If this feature is deployed within a regular area, all other regular areas must also support this feature before the demand circuit functionality can take effect. This is because external LSAs are flooded throughout all areas.
3. Do not enable this feature on a broadcast-based network topology because Hellos cannot be successfully suppressed, which means the link will remain up.

OSPF Over Non-Broadcast Networks

NBMA networks are those networks that support many (more than two) routers, but have no broadcast capability. Neighboring routers are maintained on these nets using OSPF's Hello Protocol. However, due to the lack of broadcast capability, some configuration information may be necessary to aid in the discovery of neighbors. On non-broadcast networks, OSPF protocol packets that are normally multicast need to be sent to each neighboring router, in turn. An X.25 Public Data Network (PDN) is an example of a non-broadcast network. Note the following:

- *OSPF runs in one of two modes over non-broadcast networks*. The first mode, called non-broadcast multiaccess or NBMA, simulates the operation of OSPF on a broadcast network. The second mode, called point-to-multipoint, treats the non-broadcast network as a collection of point-to-point links. Non-broadcast networks are referred to as NBMA networks or point-to-multipoint networks, depending on OSPF's mode of operation over the network.
- *In NBMA mode, OSPF emulates operation over a broadcast network.* A Designated Router is elected for the NBMA network, and the Designated Router originates an LSA for the network. The graph representation for broadcast networks and NBMA networks is identical.

NBMA Mode

NBMA mode is the most efficient way to run OSPF over non-broadcast networks, both in terms of link-state database size and in terms of the amount of routing protocol traffic. However, it has one significant restriction: It requires all routers attached to the NBMA network to be able to communicate directly. This restriction may be met on some non-broadcast networks, such as an ATM subnet utilizing SVCs. But it is often not met on other non-broadcast networks, such as PVC-only Frame Relay networks.

On non-broadcast networks in which not all routers can communicate directly, you can break the non-broadcast network into logical subnets, with the routers on each subnet being able to communicate directly. Then each separate subnet can be run as an NBMA network or a point-to-point network if each virtual circuit is defined as a separate logical subnet. This setup, however, requires quite a bit of administrative overhead, and is prone to misconfiguration. It is probably better to run such a non-broadcast network in Point-to-MultiPoint mode.

Point-to-MultiPoint Mode

Point-to-MultiPoint networks have been designed to work simply and naturally when faced with partial mesh connectivity. In Point-to-MultiPoint mode, OSPF treats all router-to-router connections over the non-broadcast network as if they were point-to-point links. No Designated Router is elected for the network, nor is there an LSA generated for the network. It may be necessary to configure the set of neighbors that are directly reachable over the Point-to-MultiPoint network. Each neighbor is identified by its IP address on the Point-to-MultiPoint network. Because no Designated Routers are elected on Point-to-MultiPoint networks, the Designated Router eligibility of configured neighbors is undefined.

Alternatively, neighbors on Point-to-MultiPoint networks may be dynamically discovered by lower-level protocols such as Inverse ARP. In contrast to NBMA networks, Point-to-MultiPoint networks have the following properties:

1. Adjacencies are established between all neighboring routers. There is no Designated Router or Backup Designated Router for a Point-to-MultiPoint network. No network-LSA is originated for Point-to-MultiPoint networks. Router Priority is not configured for Point-to-MultiPoint interfaces, nor for neighbors on Point-to-MultiPoint networks.
2. When originating a router-LSA, Point-to-MultiPoint interface is reported as a collection of “point-to-point links” to all of the interface’s adjacent neighbors, together with a single stub link advertising the interface’s IP address with a cost of 0.
3. When flooding out a non-broadcast interface (when either in NBMA or Point-to-MultiPoint mode) the Link State Update or Link State Acknowledgment packet must be replicated in order to be sent to each of the interface’s neighbors.

The following is an example of point-to-multipoint configuration on a NBMA (Frame Relay in this case) network. Attached is the resulting routing table and Router Link state along with other pertinent information:

```
interface Ethernet0
 ip address 130.10.6.1 255.255.255.0
!
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi
!
interface Serial0.1 multipoint
 ip address 130.10.10.3 255.255.255.0
 ip ospf network point-to-multipoint
 ip ospf priority 10
 frame-relay map ip 130.10.10.1 140 broadcast
 frame-relay map ip 130.10.10.2 150 broadcast
!
router ospf 2
 network 130.10.10.0 0.0.0.255 area 0
 network 130.10.6.0 0.0.0.255 area 1

R6#sh ip ospf int s 0.1
Serial0.1 is up, line protocol is up
Internet Address 130.10.10.3/24, Area 0
Process ID 2, Router ID 140.10.1.1, Network Type POINT_TO_MULTIPOINT, Cost: 6,
Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
Hello due in 00:00:18
Neighbor Count is 2, Adjacent neighbor count is 2
Adjacent with neighbor 130.10.10.2
Adjacent with neighbor 130.10.5.129

R6#sh ip ospf ne

Neighbor ID PriStateDead Time  Address        Interface
130.10.10.20FULL/  00:01:37130.10.10.2     Serial0.1
130.10.5.129 0FULL/  -00:01:53     130.10.10.1     Serial0.1
R6#

R6#sh ip ro
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
       U - per-user static route

Gateway of last resort is not set

130.10.0.0/16 is variably subnetted, 9 subnets, 3 masks
O130.10.10.2/32 [110/64] via 130.10.10.2, 00:03:28, Serial0.1
C130.10.10.0/24 is directly connected, Serial0.1
O130.10.10.1/32 [110/64] via 130.10.10.1, 00:03:28, Serial0.1
O IA130.10.0.0/22 [110/74] via 130.10.10.1, 00:03:28, Serial0.1
O130.10.4.0/24 [110/74] via 130.10.10.2, 00:03:28, Serial0.1
C130.10.6.0/24 is directly connected, Ethernet0

R6#sh ip ospf data router 140.10.1.1

  OSPF Router with ID (140.10.1.1) (Process ID 2)

Router Link States (Area 0)

  LS age: 806
```

```
Options: (No TOS-capability)
LS Type: Router Links
Link State ID: 140.10.1.1
Advertising Router: 140.10.1.1
LS Seq Number: 80000009
Checksum: 0x42C1
Length: 60
Area Border Router
 Number of Links: 3

  Link connected to: another Router (point-to-point)
   (Link ID) Neighboring Router ID: 130.10.10.2
   (Link Data) Router Interface address: 130.10.10.3
    Number of TOS metrics: 0
     TOS 0 Metrics: 64

  Link connected to: another Router (point-to-point)
   (Link ID) Neighboring Router ID: 130.10.5.129
   (Link Data) Router Interface address: 130.10.10.3
    Number of TOS metrics: 0
     TOS 0 Metrics: 64

  Link connected to: a Stub Network
   (Link ID) Network/subnet number: 130.10.10.3
   (Link Data) Network Mask: 255.255.255.255
    Number of TOS metrics: 0
     TOS 0 Metrics: 0
```

BGP Internetwork Design Guidelines

The Border Gateway Protocol (BGP) is an interautonomous system routing protocol. The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems. This network reachability information includes information on the list of Autonomous Systems (ASs) that reachability information traverses. BGP-4 provides a new set of mechanisms for supporting classless interdomain routing. These mechanisms include support for advertising an IP prefix and eliminate the concept of network *class* within BGP. BGP-4 also introduces mechanisms that allow aggregation of routes, including aggregation of AS paths. These changes provide support for the proposed supernetting scheme. This section describes how BGP works and it can be used to participate in routing with other networks that run BGP. The following topics are covered:

- BGP operation
- BGP attributes
- BGP path selection criteria
- Understanding and defining BGP routing policies

BGP Operation

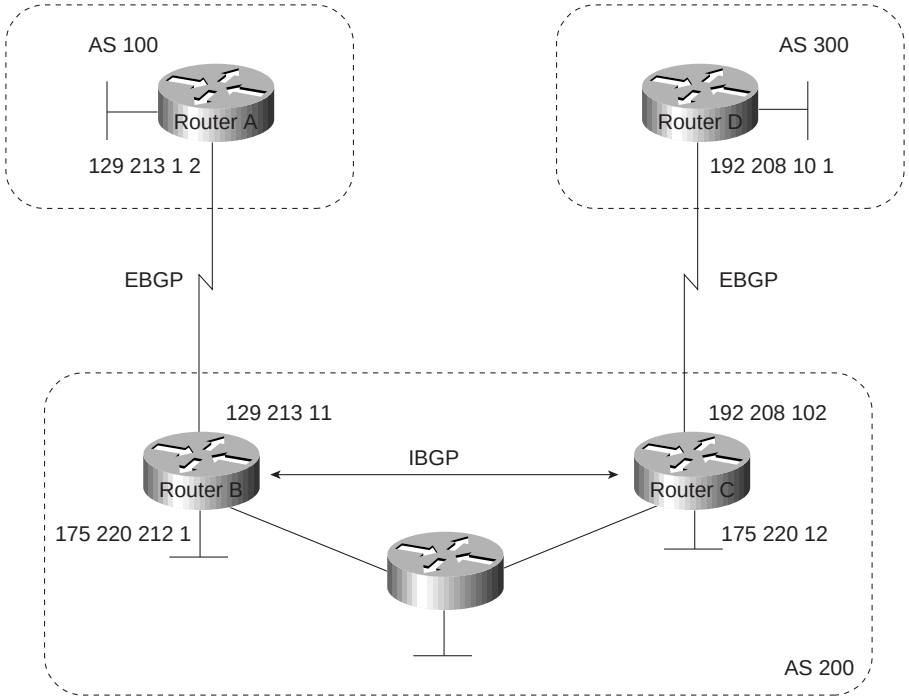
This section presents fundamental information about BGP, including the following topics:

- 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). 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 (EBGP)" later in this chapter), the commands for configuring EBGP and IBGP are the same. This chapter 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 3-17 shows a network that demonstrates the difference between EBGP and IBGP.

Figure 3-17 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 3-17, 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).

Note Routers A and B are running EBGP, and Routers B and C are running IBGP, as shown in Figure 3-17. 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 each other, IBGP peers do not have to be directly connected.

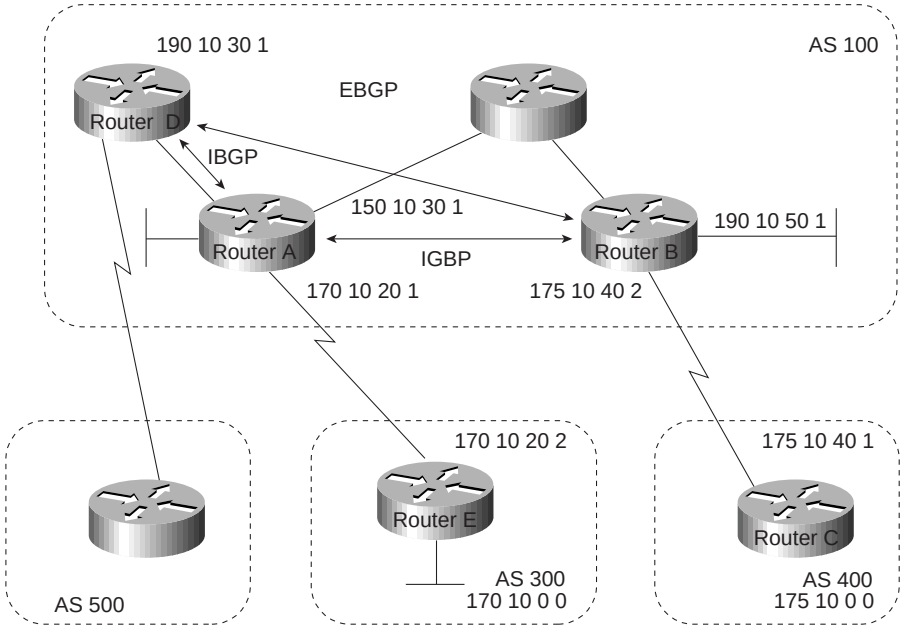
All BGP speakers within an AS must establish a peer relationship with one another. That is, the BGP speakers within an AS must be fully meshed logically. BGP-4 provides two techniques that alleviate the requirement for a logical full mesh: confederations and route reflectors. For information about these 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.

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, more scalable, and provides more efficient ways of controlling the exchange of information within the AS. It also presents a consistent view of the AS to external neighbors. For example, IBGP provides ways to control the exit point from an AS. Figure 3-18 shows a topology that demonstrates IBGP.

Figure 3-18 Internal BGP example.

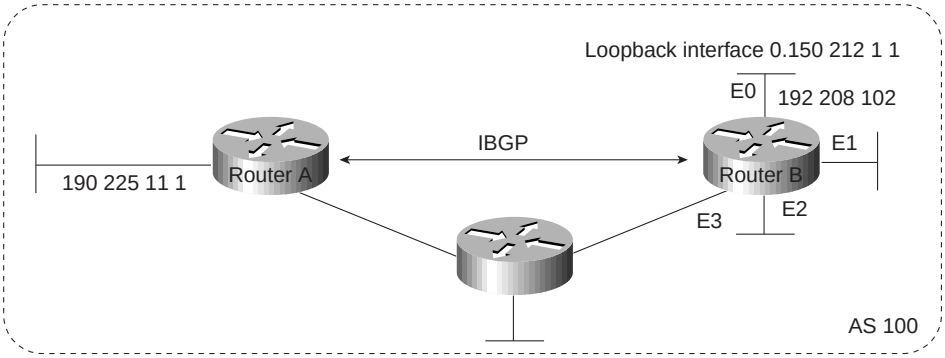


When a BGP speaker receives an update from other BGP speakers in its own AS (that is, via IBGP), the receiving BGP speaker uses EBGP to forward the update to external BGP speakers only. This behavior of IBGP is why it is necessary for BGP speakers within an AS to be fully meshed.

For example, in Figure 3-18, 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 3-19 shows a network in which using the loopback interface is advantageous.

Figure 3-19 Use of loopback interfaces.



In Figure 3-19, 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.

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

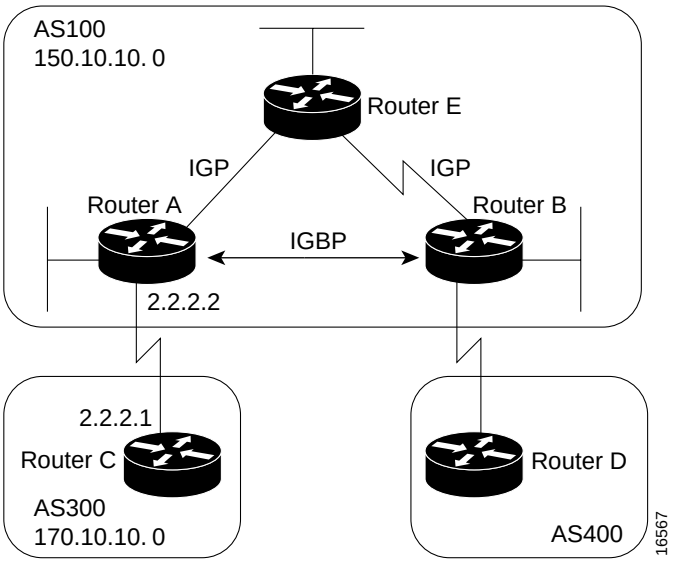
External BGP (EBGP)

When two BGP speakers that are not in the same AS run BGP to exchange routing information, they are said to be running EBGP.

Synchronization

When an AS provides transit service to other ASs when 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 3-20 demonstrates this synchronization rule.

Figure 3-20 EBGP synchronization rule.



In Figure 3-20, 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. It states that if an AS (such as AS 100 in Figure 3-20) 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.

Disabling Synchronization

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.

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.

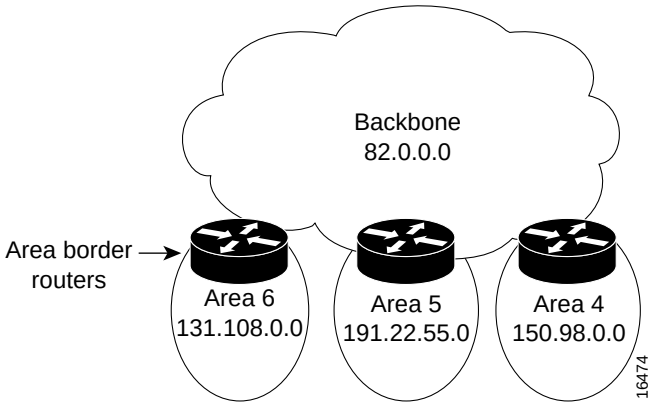
The **match** and **set route map** configuration commands are used to define the condition portion of a route map. The **match** command specifies a criteria that must be matched, and the **set** command specifies an action that is to be taken if the routing update meets the condition defined by the **match** command. The 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 the 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 until 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. Figure 3-21 shows a topology that demonstrates the use of route maps.

Figure 3-21 Route map example.



In Figure 3-21, 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.

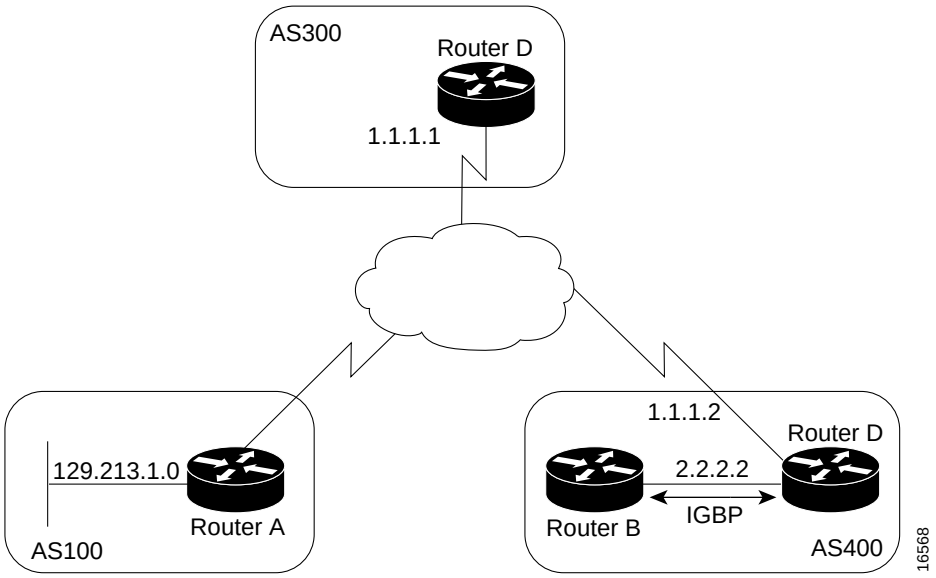
Advertising Networks

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

This section uses the topology shown in Figure 3-22 to demonstrate how networks that originate from an AS can be advertised.

Figure 3-22 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 3-22 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 3-22, 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.

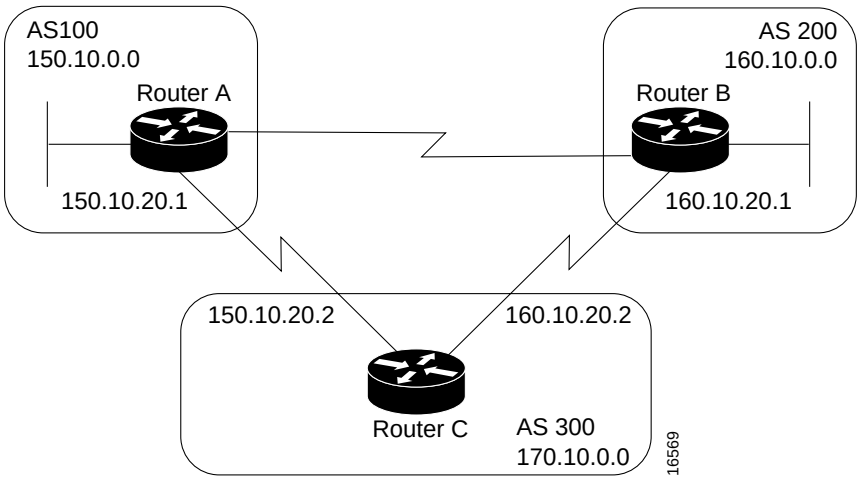
Using the **network** Command

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 3-23 shows another topology that demonstrates the effects of the **network** command.

Figure 3-23 Network advertisement example 2.



The following configurations use the **network** command to configure the routers shown in Figure 3-23:

```
!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 3-23, 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 Attributes

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:

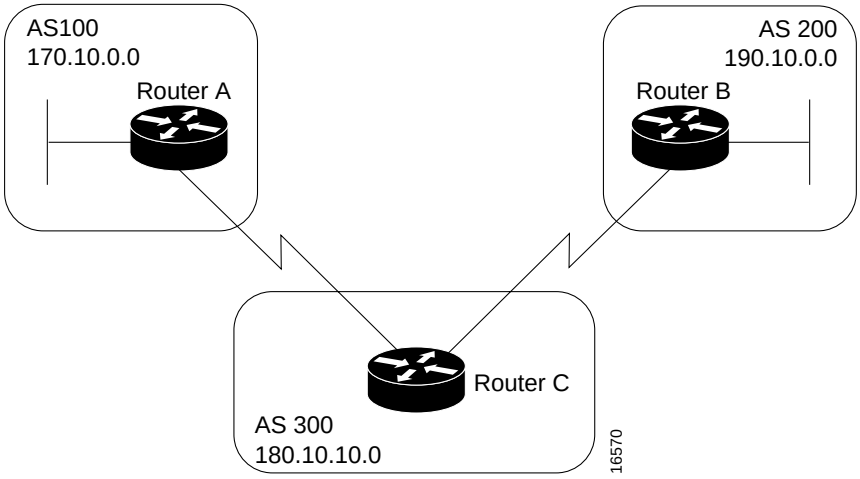
- AS_path Attribute
- Origin Attribute
- Next Hop Attribute
- Weight Attribute
- Local Preference Attribute

- 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 3-24.

Figure 3-24 AS_path attribute.



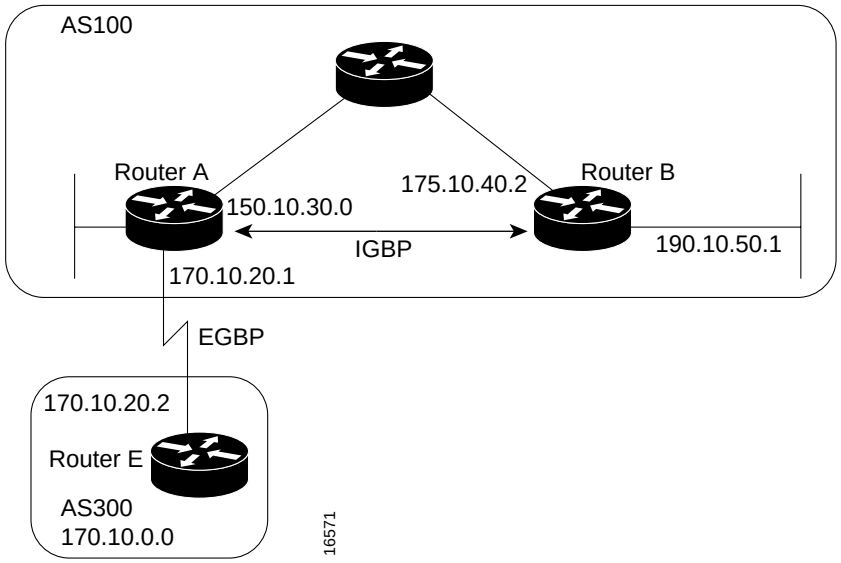
Origin Attribute

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 3-25 shows a network that demonstrates the value of the origin attribute.

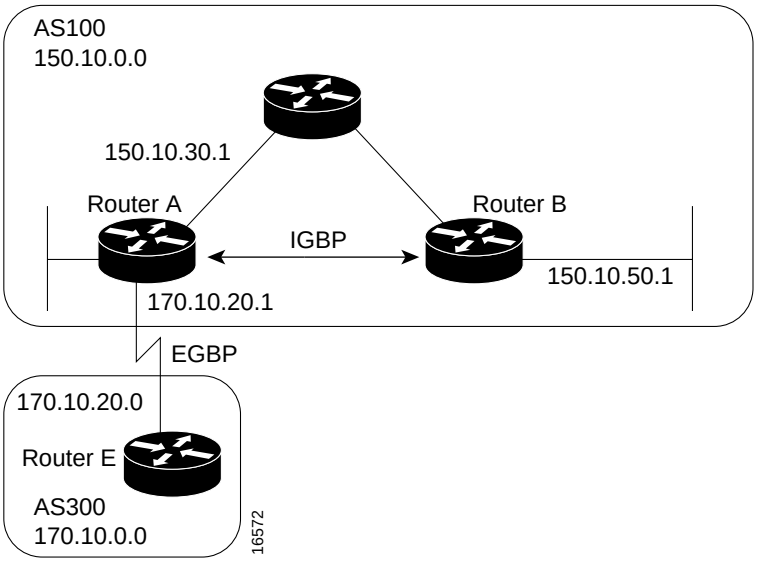
Figure 3-25 Origin attribute.



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 3-26.

Figure 3-26 Next hop attribute.



In Figure 3-26, 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.

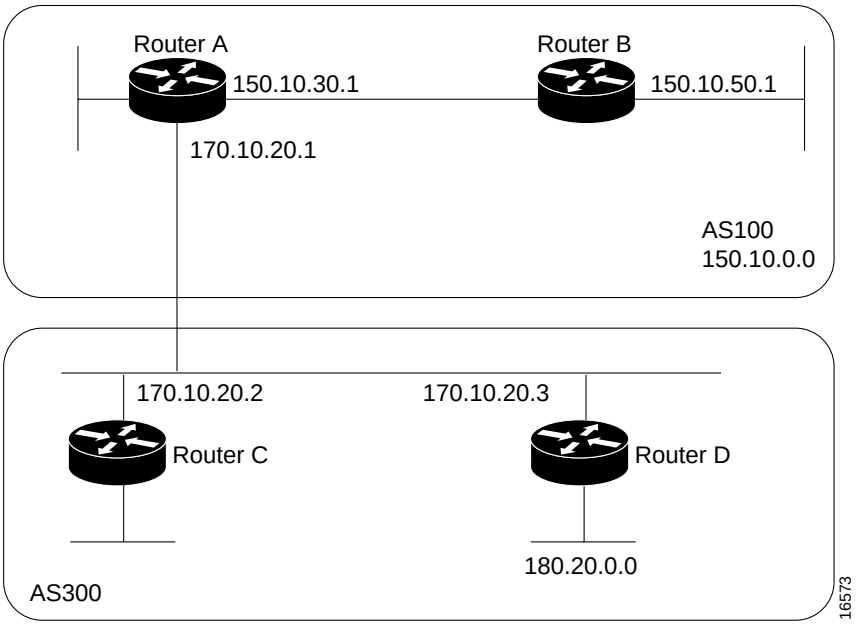
BGP specifies that the next hop of EBGP-learned routes should be carried without modification into IBGP. Because of that rule, Router A advertises 170.10.0.0 to its IBGP peer (Router B) with a next hop attribute of 170.10.20.2. As a result, according to Router B, the next hop to reach 170.10.0.0 is 170.10.20.2, 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.

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 3-27.

Figure 3-27 Multiaccess media network.

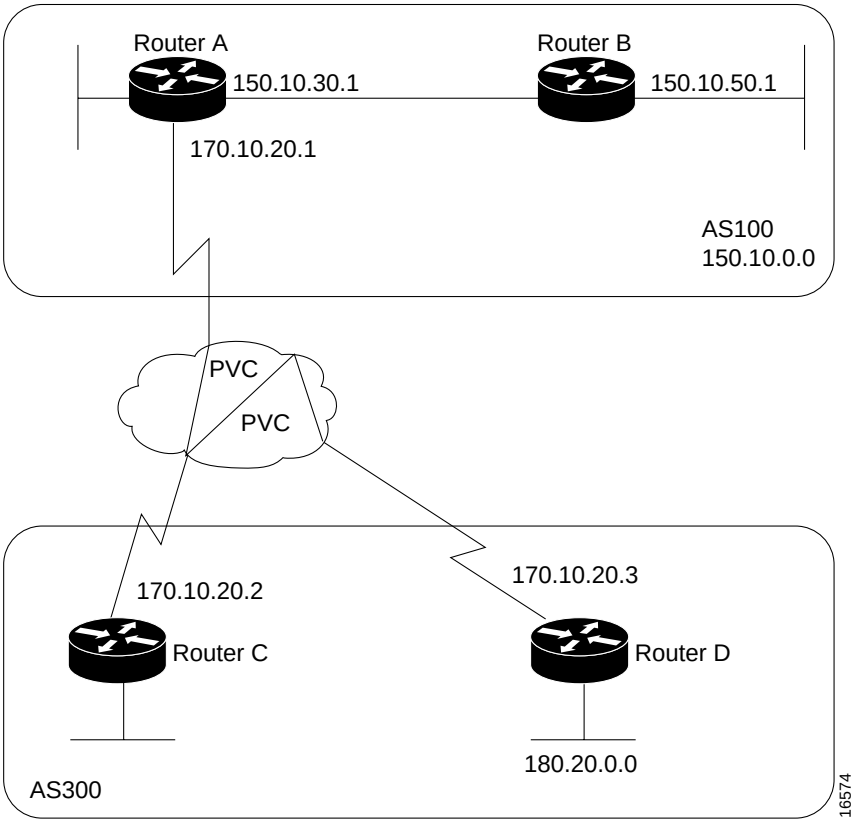


In Figure 3-27, Routers C and D in AS 300 are running OSPF. Router C is running BGP with Router A. Router C can reach network 180.20.0.0 via 170.10.20.3. When Router C sends a BGP update to Router A regarding 180.20.0.0, it sets the next hop attribute to 170.10.20.3, instead of its own IP address (170.10.20.2). 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 3-28, three networks are connected by a nonbroadcast media access (NBMA) cloud, such as Frame Relay.

Figure 3-28 Next Hop attritbute 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:

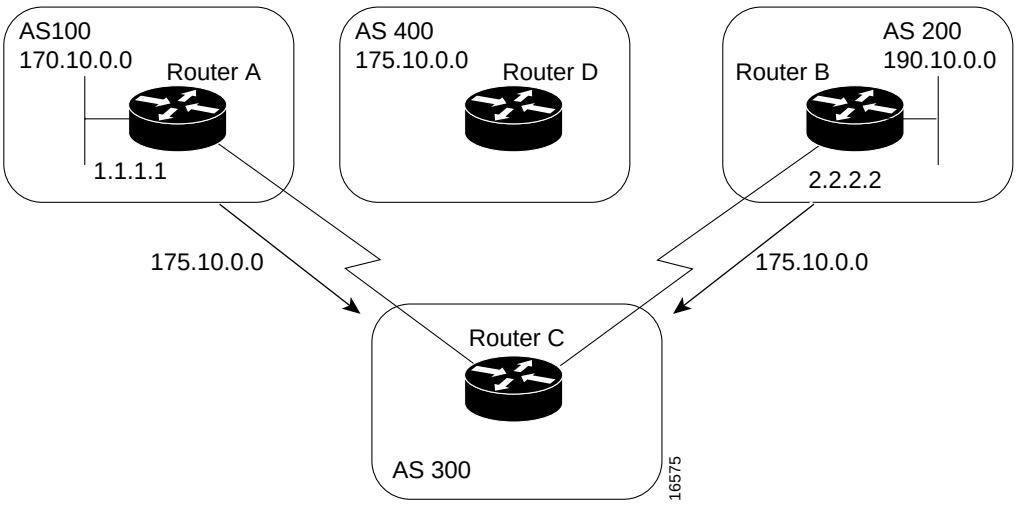
```
!Router C
router bgp 300
neighbor 170.10.20.1 remote-as 100
neighbor 170.10.20.1 next-hop-self
```

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 3-29.

Figure 3-29 Weight attribute example.



In Figure 3-29, 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.) 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:

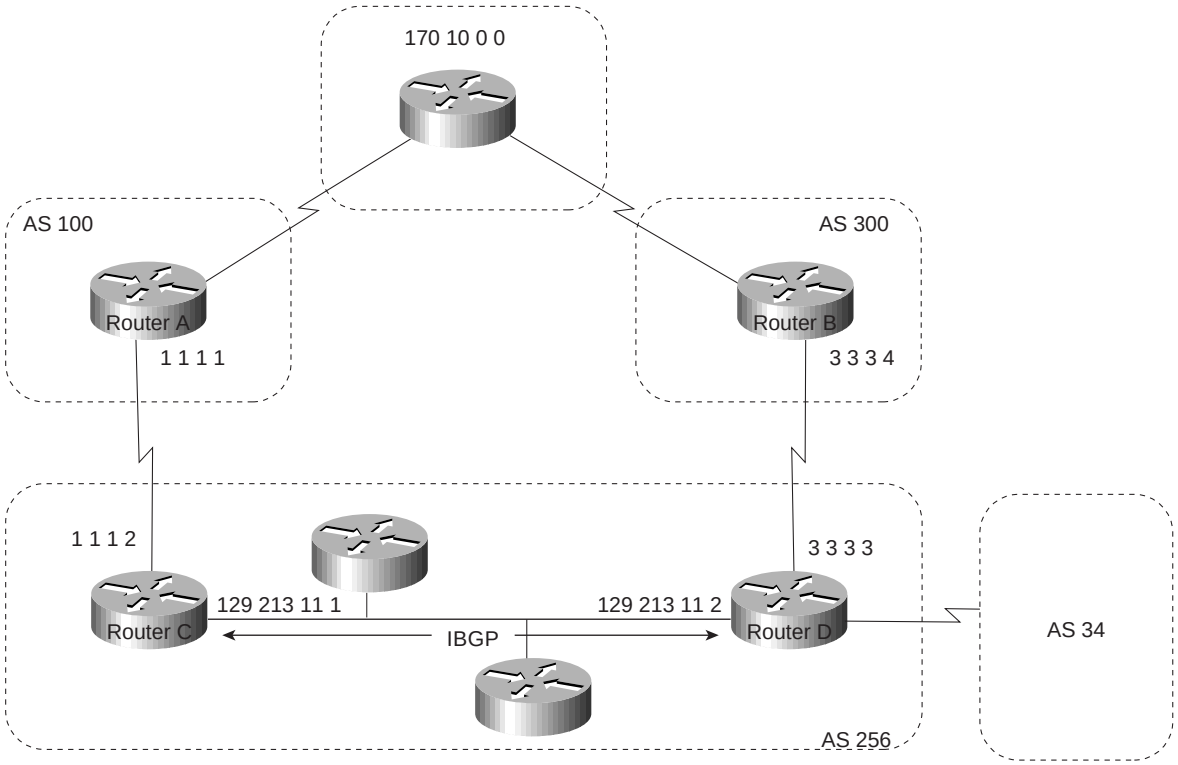
```
!Router C
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 weight 2000
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 weight 1000
```

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 relevant only 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 3-30 demonstrates the local preference attribute.

Figure 3-30 Local preference.



In Figure 3-30, 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 3-30, 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

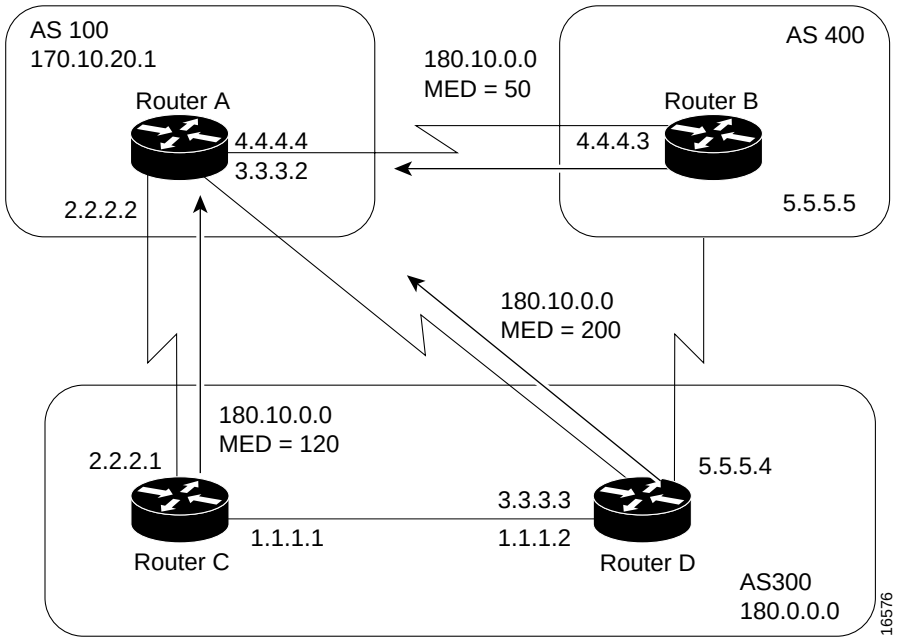
The multi-exit discriminator (MED) attribute is a hint to external neighbors about the preferred path into an AS when there are multiple entry points into the AS. A lower MED value is preferred over a higher MED value. The default value of the MED attribute is 0.

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 3-31 demonstrates the use of the MED attribute.

Figure 3-31 MED example.



In Figure 3-31, 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 3-31). 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 3-3.

Table 3-2 Predefined Communities

Community	Meaning
no-export	Do not advertise this route to EBGP peers.
no-advertised	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.

BGP Path Selection Criteria

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.

Understanding and Defining BGP Routing Policies

This section describes how to understand and define BGP Policies to control 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 3-3.

Table 3-3 BGP Administrative 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.

BGP Filtering

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

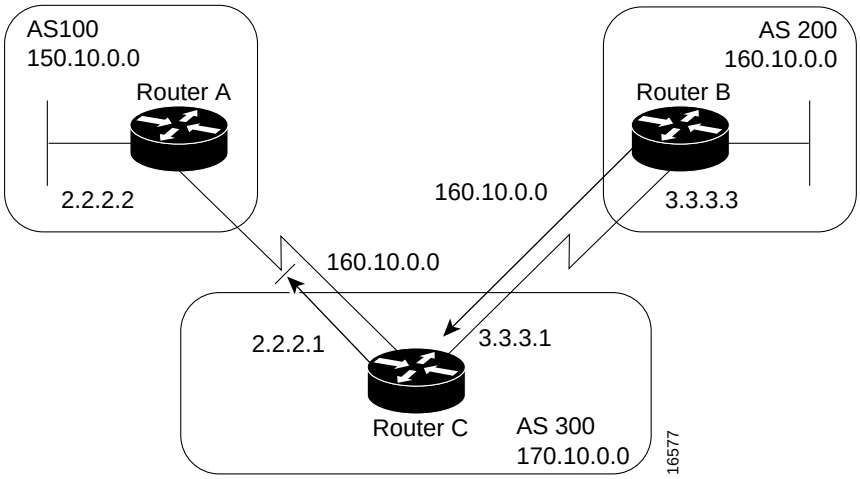
- Prefix Filtering
- AS_path Filtering
- Route Map Filtering
- Community Filtering

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 3-32 demonstrates the usefulness of prefix filtering.

Figure 3-32 Prefix route filtering.



In Figure 3-32, 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 3-32 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
```

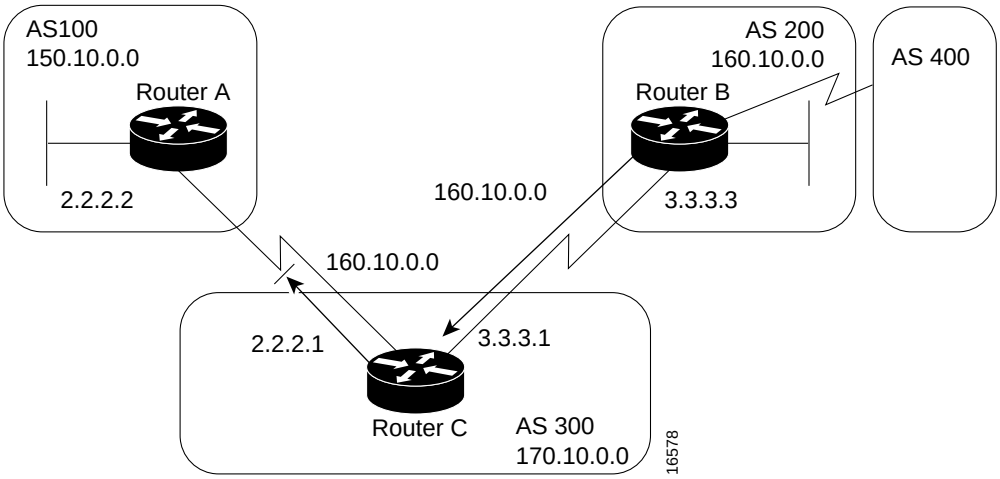
To restrict the update to 160.0.0.0/8 only, you have to use an extended access list, such as the following:

```
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 3-33 demonstrates the usefulness of AS_path filters.

Figure 3-33 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:

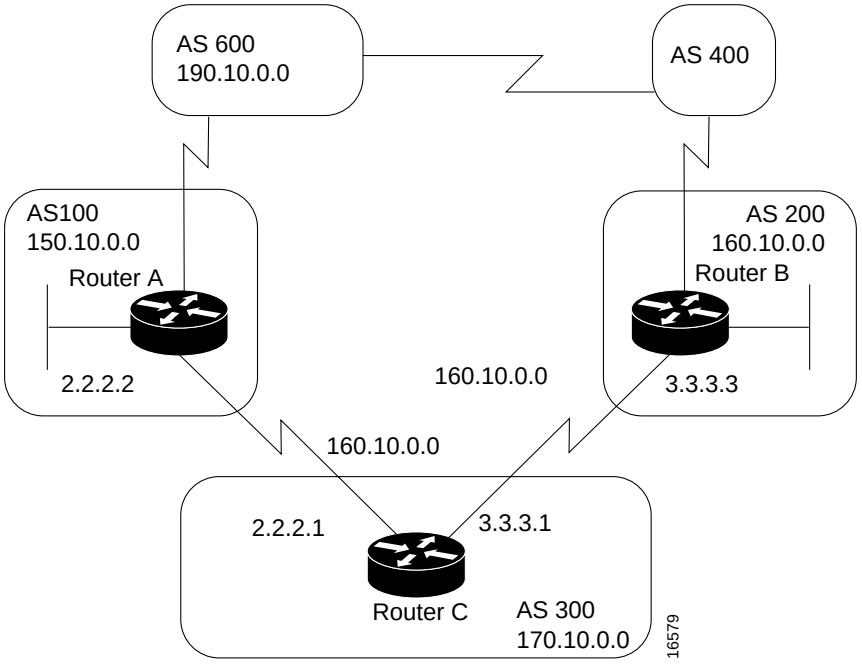
```
show ip bgp regexp regular-expression
```

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. The network shown in Figure 3-34 demonstrates using route maps to filter BGP updates.

Figure 3-34 BGP route map filtering.



Assume that in Figure 3-34, 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:

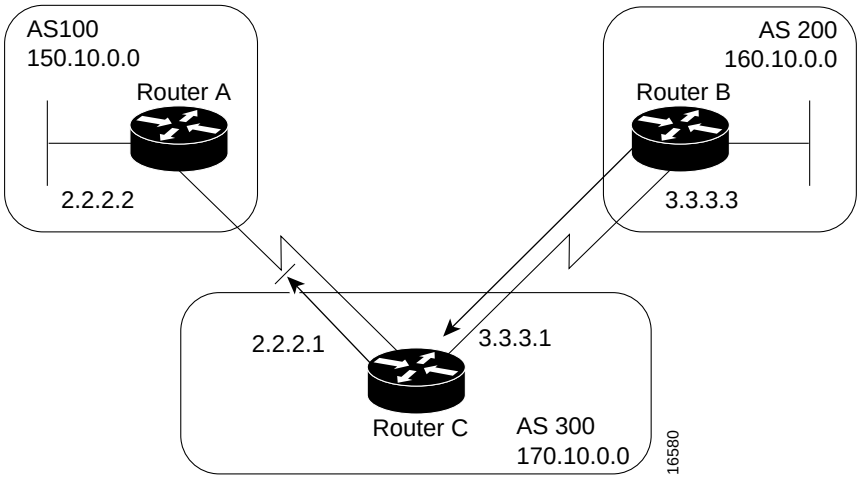
```
!Router C
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map STAMP in
!
route-map STAMP permit 10
match as-path 1
set weight 20
!
ip as-path access-list 1 permit ^200$
```

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.

Community Filtering

The network shown in Figure 3-35 demonstrates the usefulness of community filters.

Figure 3-35 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
```

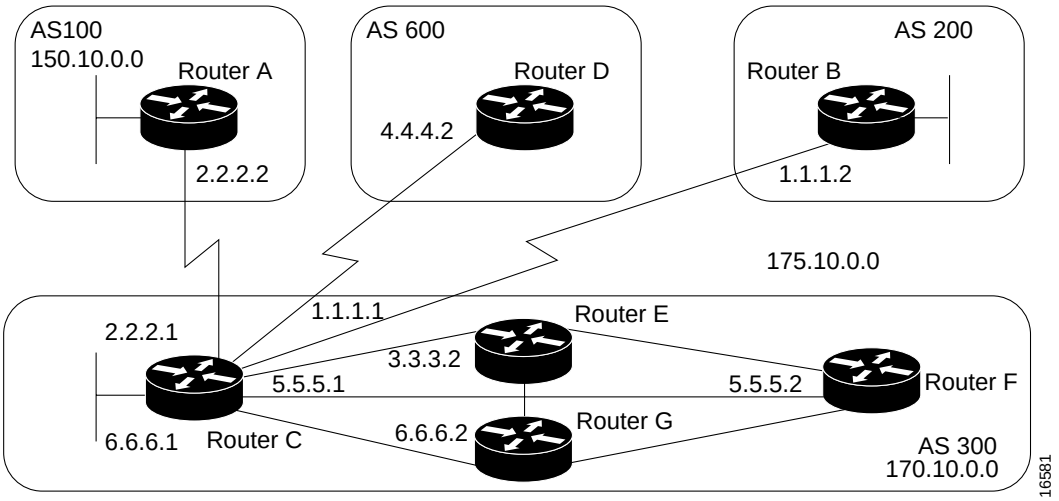
In the preceding configuration, any route that has 100 in its community attribute matches community list 1 and has its weight set to 20. Any route whose community attribute is only 200 (by virtue of the exact keyword) matches community list 2 and has its weight set to 10. In the last community list (list 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 override options that are set only for incoming updates. The use of BGP peer groups is demonstrated by the network shown in Figure 3-36

Figure 3-36 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 be used only 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
```

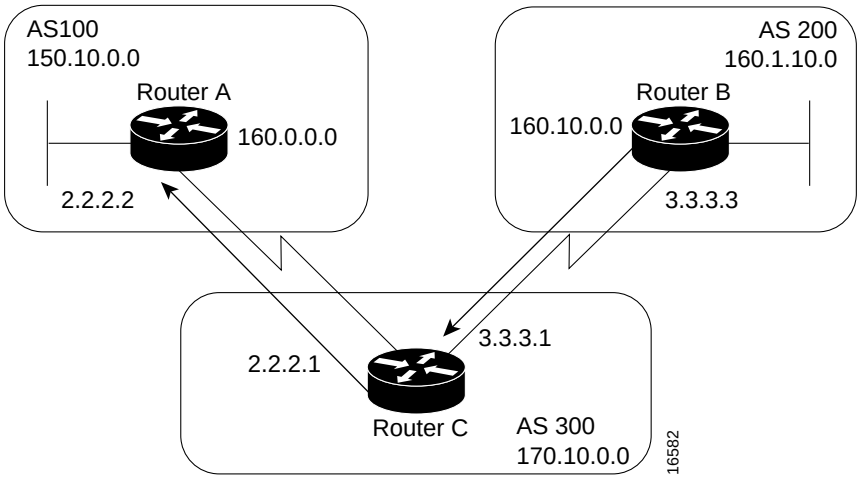
In the preceding configuration, the **neighbor remote-as router** configuration commands are placed outside of the **neighbor peer-group router** configuration commands because different external ASs have to be defined. Also note that this configuration defines filter list 3, which can be used to override 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). 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 3-37.

Figure 3-37 Aggregation example.



In Figure 3-37, 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. 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. 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 3-37 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:

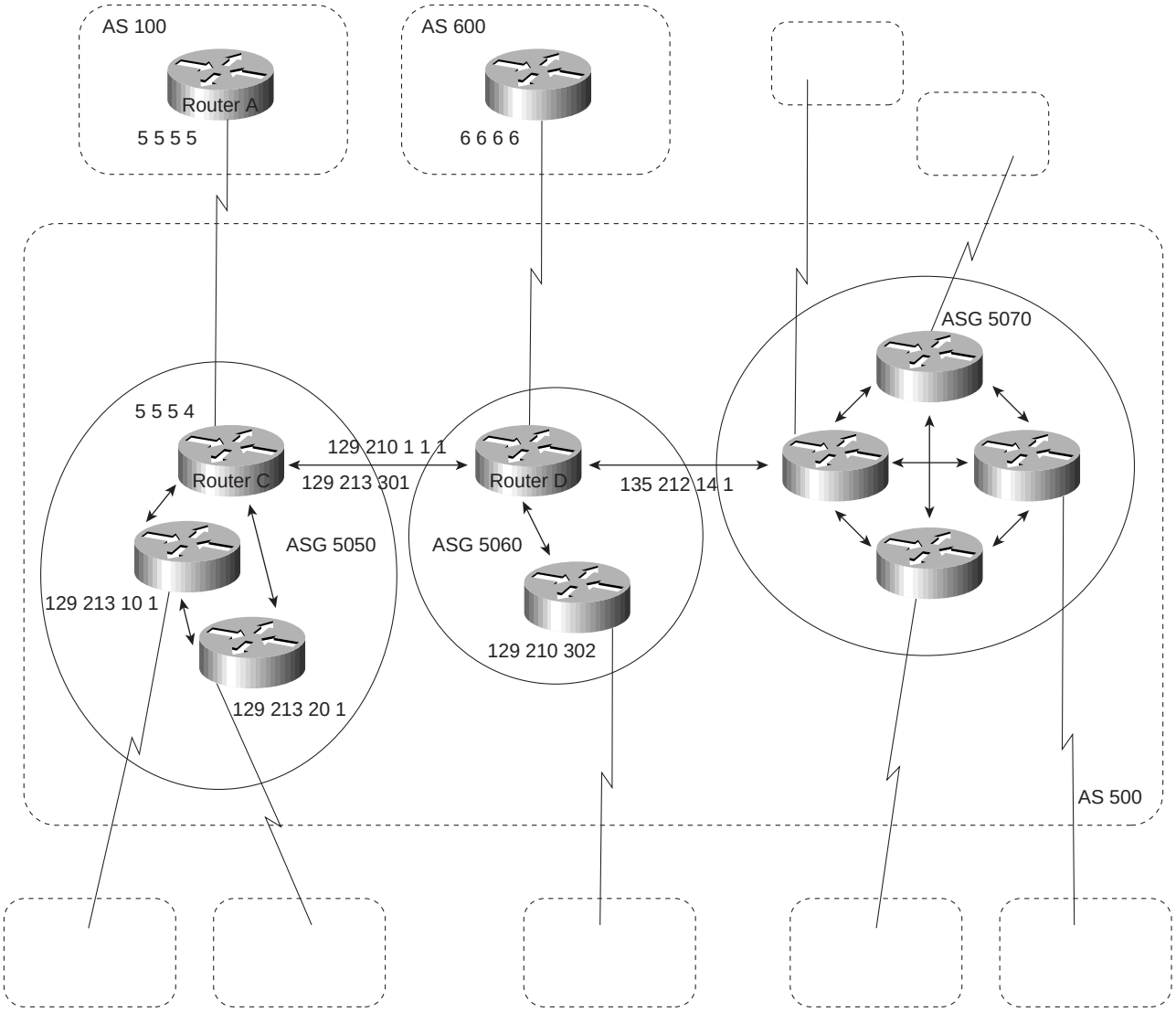
```
route-map SETORIGIN permit 10
set origin igp
!
aggregate-address 160.0.0.0 255.0.0.0 attribute-map SETORIGIN
```

Note **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.

Confederations

A *confederation* is a technique for reducing the IBGP mesh inside the AS. Consider the network shown in Figure 3-38.

Figure 3-38 Example of confederations.



In Figure 3-38, 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 3-38. 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:

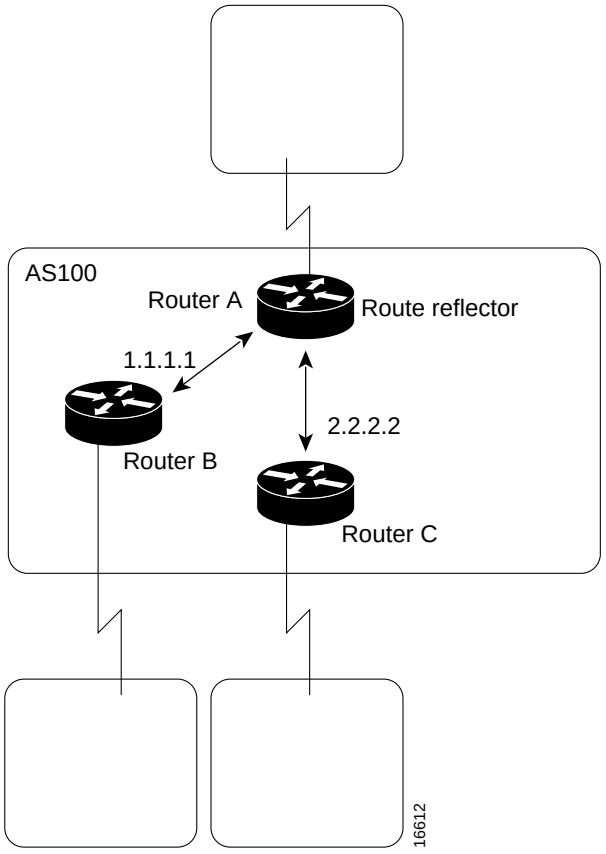
```
!Router A
router bgp 100
neighbor 5.5.5.4 remote-as 500
```

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 3-39 demonstrates how route reflectors work.

Figure 3-39 imple route reflector example.



Without a route reflector, the network shown in Figure 3-39 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.

Route Flap Dampening

Route flap dampening (introduced in 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).

Summary of BGP

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.

Summary

Recall the following design implications of the Enhanced Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF) protocols, and the BGP protocol:

- Network topology
- Addressing and route summarization
- Route selection
- Convergence
- Network scalability
- Security

This chapter outlined these general routing protocol issues and focused on design guidelines for the specific IP protocols.