CHAPTER 18

Configuring IP Routing Protocols

This chapter describes how to configure the various Internet Protocol (IP) routing protocols. For a complete description of the commands listed in this chapter, refer to the "IP Routing Protocols Commands" chapter of the *Access and Communication Servers Command Reference* publication. For information on configuring the IP protocol, refer to the "Configuring IP" chapter of this publication. For historical background and a technical overview of IP routing protocols, see the *Internetworking Technology Overview* publication.

Cisco's Implementation of IP Routing Protocols

Cisco's implementation of each IP routing protocol is discussed in detail at the beginning of the individual protocol configuration sections throughout this chapter.

IP routing protocols are divided into two classes: Interior Gateway Protocols (IGPs) and Exterior Gateway Protocols (EGPs). The IGPs and EGPs that Cisco supports are listed in the next two sections.

Note Many routing protocol specifications refer to communication servers as *gateways*, so the word *gateway* often appears as part of routing protocol names. However, a communication server usually is defined as a Layer 3 internetworking device, whereas a protocol translation gateway usually is defined as a Layer 7 internetworking device. The reader should understand that regardless of whether a routing protocol name contains the word "gateway," routing protocol activities occur at Layer 3 of the OSI reference model.

The Interior Gateway Protocols

Interior protocols are used for routing networks that are under a common network administration. All IP Interior Gateway Protocols must be specified with a list of associated networks before routing activities can begin. A routing process listens to updates from other communication servers on these networks and broadcasts its own routing information on those same networks. The interior routing protocols supported are as follows:

- Internet Gateway Routing Protocol (IGRP)
- Enhanced Internet Gateway Routing Protocol (EIGRP)
- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP)

The Exterior Gateway Protocols

Exterior protocols exchange routing information between networks that do not share a common administration. The supported exterior routing protocols are as follows:

- Border Gateway Protocol (BGP)
- Exterior Gateway Protocol (EGP)

IP Exterior Gateway Protocols require three pieces of information before routing can begin:

- A list of neighbor (or peer) communication servers with which to exchange routing information
- A list of networks to advertise as directly reachable
- The autonomous system number of the local communication server

Router Discovery Protocols

Our communication servers also support two communication server discovery protocols, Gateway Discovery Protocol (GDP) and ICMP Router Discovery Protocol (IRDP), which allow hosts to locate communication servers.

GDP was developed by Cisco and is not an industry standard. Examples of unsupported GDP clients can be obtained upon request from Cisco. Our IRDP implementation fully conforms to the communication server discovery protocol outlined in RFC 1256.

Multiple Routing Protocols

You can configure multiple routing protocols in a single communication server to connect networks that use different routing protocols. You can, for example, run RIP on one subnetted network, IGRP on another subnetted network, and exchange routing information between them in a controlled fashion. The available routing protocols were not designed to interoperate with one another, so each protocol collects different types of information and reacts to topology changes in its own way. For example, RIP uses a hop-count metric and IGRP uses a five-element vector of metric information. In the case where routing information is being exchanged between different networks that use different routing protocols, there are many configuration options that allow you to filter the exchange of routing information.

Our communication servers can handle simultaneous operation of up to 30 dynamic IP routing processes.The combination of routing processes on a communication server can consist of the following protocols (with the limits noted):

- Up to 30 IGRP routing processes
- Up to 30 OSPF routing processes
- One RIP routing process
- One BGP routing process
- Up to 30 EGP routing processes

IP Routing Protocols Task List

With any of the IP routing protocols, you need to create the routing process, associate networks with the routing process, and customize the routing protocol for your particular network.

You will need to perform some combination of the tasks in the following sections to configure IP routing protocols:

- Determine a Routing Process
- Configure IGRP
- Configure Enhanced IGRP
- Configure OSPF
- Configure RIP
- Configure BGP
- Configure EGP
- Configure GDP
- Configure IRDP
- Configure IP Multicast Routing
- Configure Routing Protocol-Independent Features
- Monitor and Maintain the IP Network

See the end of this chapter for IP routing protocol configuration examples.

Determine a Routing Process

Choosing a routing protocol is a complex task. When choosing a routing protocol, consider (at least) the following:

- Internetwork size and complexity
- Support for variable-length subnet masks (VLSM) for OSPF
- Internetwork traffic levels
- Security needs
- Reliability needs
- Internetwork delay characteristics
- Organizational policies
- Organizational acceptance of change

The following sections describe the configuration tasks associated with each supported routing protocol. This publication does not provide in-depth information on how to choose routing protocols; you must choose routing protocols that best suit your needs. For detailed information on the technology behind the major routing protocols, see the *Internetworking Technology Overview* manual or other internetworking publications.

Configure IGRP

The Interior Gateway Routing Protocol (IGRP) is a dynamic distance-vector routing protocol designed by Cisco Systems in the mid-1980s for routing in an autonomous system that contains large, arbitrarily complex networks with diverse bandwidth and delay characteristics.

Cisco's IGRP Implementation

IGRP uses a combination of user-configurable metrics, including internetwork delay, bandwidth, reliability, and load.

IGRP also advertises three types of routes: interior, system, and exterior, as shown in Figure 18-1. Interior routes are routes between subnets in the network attached to a communication server interface. If the network attached to a communication server is not subnetted, IGRP does not advertise interior routes.

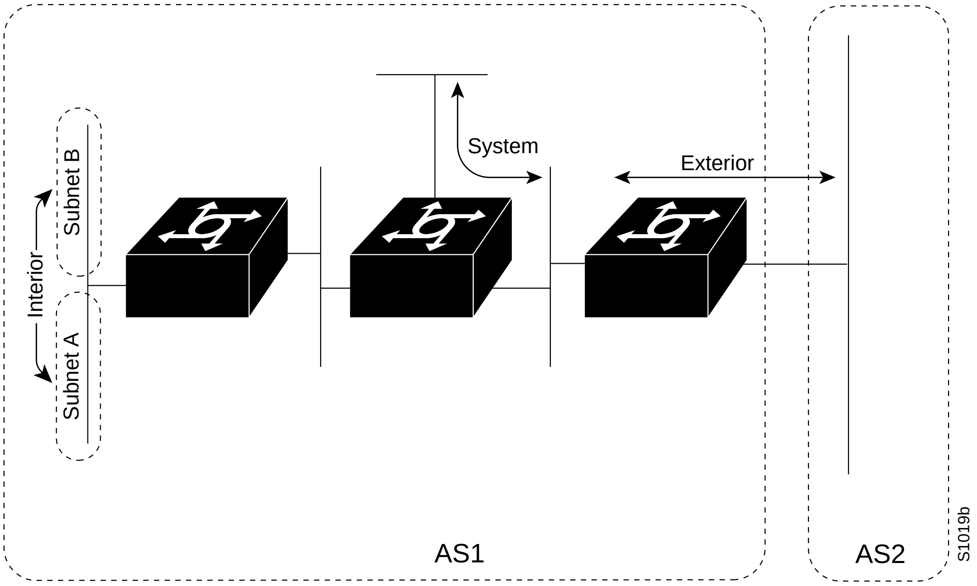

Figure 18-1 Interior, System, and Exterior Routes

System routes are routes to networks within an autonomous system. The communication server derives system routes from directly connected network interfaces and system route information provided by other IGRP-speaking communication servers. System routes do not include subnet information.

Exterior routes are routes to networks outside the autonomous system that are considered when identifying a *gateway of last resort*. The communication server chooses a gateway of last resort from the list of exterior routes that IGRP provides. The communication server uses the gateway (communication server) of last resort if it does not have a better route for a packet and the destination is not a connected network. If the autonomous system has more than one connection to an external network, different communication servers can choose different exterior communication servers as the gateway of last resort.

IGRP Updates

By default, a communication server running IGRP sends an update broadcast every 90 seconds. It declares a route inaccessible if it does not receive an update from the first communication server in the route within three update periods (270 seconds). After seven update periods (630 seconds), the communication server removes the route from the routing table.

IGRP uses *flash update* and *poison reverse updates* to speed up the convergence of the routing algorithm. Flash update is the sending of an update sooner than the standard periodic update interval of notifying other communication servers of a metric change. Poison reverse updates are intended to defeat larger routing loops caused by increases in routing metrics. The poison reverse updates are sent to remove a route and place it in *holddown*, which keeps new routing information from being used for a certain period of time.

IGRP Configuration Task List

To configure IGRP, perform the tasks in the following sections. Creating the IGRP routing process is mandatory; the other tasks described are optional.

- Create the IGRP Routing Process
- Allow Point-to-Point Updates for IGRP
- Define Unequal-Cost Load Balancing
- Control Traffic Distribution
- Adjust the IGRP Metric Weights
- Disable Holddown
- Enforce a Maximum Network Diameter
- Validate Source IP Addresses

Create the IGRP Routing Process

To create the IGRP routing process, perform the following required tasks:

Task		Command
Step 1	Enter global configuration mode.	See Table 2-1 earlier in this manual.
Step 2	Enable an IGRP routing process, which places you in router configuration mode.	**router igrp** *process-number*
Step 3	Associate networks with an IGRP routing process.	**network** *network-number*

IGRP sends updates to the interfaces in the specified networks. If an interface's network is not specified, it will not be advertised in any IGRP update.

Allow Point-to-Point Updates for IGRP

Because IGRP is normally a broadcast protocol, in order for IGRP routing updates to reach point-to-point or nonbroadcast networks, you must configure the communication server to permit this exchange of routing information.

To permit information exchange, perform the following task in router configuration mode:

Task	Command
Define a neighboring communication server with which to exchange point-to-point routing information.	**neighbor** *ip-address*

To control the set of interfaces that you want to exchange routing updates with, you can disable the sending of routing updates on specified interfaces by configuring the **passive-interface** command. See the “Filter Routing Information” section later in this chapter.

Define Unequal-Cost Load Balancing

IGRP can simultaneously use an asymmetric set of paths for a given destination. This feature is known as *unequal-cost load balancing*. Unequal-cost load balancing allows traffic to be distributed among multiple (up to four) unequal-cost paths to provide greater overall throughput and reliability. Alternate path variance (that is, the difference in desirability between the primary and alternate paths) is used to determine the *feasibility* of a potential route. An alternate route is *feasible* if the next communication server in the path is *closer* to the destination (has a lower metric value) than the current communication server and if the metric for the entire alternate path is *within* the variance. Only paths that are feasible can be used for load balancing and included in the routing table. These conditions limit the number of cases in which load balancing can occur, but ensure that the dynamics of the network will remain stable.

The following general rules apply to IGRP unequal-cost load balancing:

- IGRP will accept up to four paths for a given destination network.
- The local best metric must be greater than the metric learned from the next communication server; that is, the next-hop communication server must be closer (have a smaller metric value) to the destination than the local best metric.
- The alternative path metric must be within the specified *variance* of the local best metric. The multiplier times the local best metric for the destination must be greater than or equal to the metric through the next communication server.

If these conditions are met, the route is deemed feasible and can be added to the routing table.

By default, the amount of variance is set to one (equal-cost load balancing). You can define how much worse an alternate path can be before that path is disallowed by performing the following task in router configuration mode:

Task	Command
Define the variance associated with a particular path.	**variance** *multiplier*

See the "IP Routing Protocol Configuration Examples" section at the end of this chapter for an example of configuring IGRP feasible successor.

Note By using the variance feature, the communication server can balance traffic across all feasible paths and can immediately converge to a new path if one of the paths should fail.

Control Traffic Distribution

By default, if IGRP or Enhanced IGRP have multiple routes of unequal cost to the same destination, the communication server will distribute traffic among the different routes by giving each route a share of the traffic in inverse proportion to its metric. If you want to have faster convergence to alternate routes but you do not want to send traffic across inferior routes in the normal case, you might prefer to have no traffic flow along routes with higher metrics.

To control how traffic is distributed among multiple routes of unequal cost, perform the following task in router configuration mode:

Task	Command
Distribute traffic proportionately to the ratios of metrics, or by the minimum-cost route.	**traffic-share** {**balanced** | **min**}

Adjust the IGRP Metric Weights

You have the option of altering the default behavior of IGRP routing and metric computations. For example, you can tune system behavior to enable transmissions via satellite. Although IGRP metric defaults were carefully selected to provide excellent operation in most networks, you can adjust the IGRP metric. Adjusting IGRP metric weights can dramatically affect network performance, however, so ensure you make all metric adjustments carefully.

To adjust the IGRP metric weights, perform the following task in router configuration mode. Due to the complexity of this task, we recommend that you only perform it with guidance from an experienced system designer.

Task	Command
Adjust the IGRP metric.	**metric weights** *tos k1 k2 k3 k4 k5*

By default, the IGRP composite metric is a 24-bit quantity that is a sum of the segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media (FDDI, Ethernet, and serial lines running from 9600 bps to T1 rates), the route with the lowest metric reflects the most desirable path to a destination.

Disable Holddown

When a communication server learns that a network is at a greater distance than was previously known, or it learns the network is down, the route to that network is placed into holddown. During the holddown period, the route is advertised, but incoming advertisements about that network from any communication server other than the one that originally advertised the network's new metric will be ignored. This mechanism is often used to help avoid routing loops in the network, but has the effect of increasing the topology convergence time. To disable holddowns with IGRP, perform the following task in router configuration mode. All communication servers in an IGRP autonomous system must be consistent in their use of holddowns.

Task	Command
Disable the IGRP holddown period.	**no metric holddown**

Enforce a Maximum Network Diameter

The communication server enforces a maximum diameter to the IGRP network. Routes whose hop counts exceed this diameter will not be advertised. The default maximum diameter is 100 hops. The maximum diameter is 255 hops.

To configure the maximum diameter, perform the following task in router configuration mode:

Task	Command
Configure the maximum network diameter.	**metric maximum-hops** *hops*

Validate Source IP Addresses

To disable the default function that validates the source IP addresses of incoming routing updates, perform the following task in router configuration mode:

Task	Command
Disable the checking and validation of the source IP address of incoming routing updates.	**no validate-update-source**

Configure Enhanced IGRP

Enhanced IGRP is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco Systems, Inc. Enhanced IGRP uses the same distance vector algorithm and distance information as IGRP. However, the convergence properties and the operating efficiency of Enhanced IGRP have improved significantly over IGRP.

The convergence technology is based on research conducted at SRI International and employs an algorithm referred to as the Diffusing Update Algorithm (DUAL). This algorithm guarantees loop-free operation at every instant throughout a route computation and allows all communication servers involved in a topology change to synchronize at the same time. Communication servers that are not affected by topology changes are not involved in recomputations. The convergence time with DUAL rivals that of any other existing routing protocol.

Cisco's Implementation of Enhanced IGRP

IP Enhanced IGRP provides the following features:

- Automatic redistribution. IP IGRP routes can be automatically redistributed into Enhanced IGRP, and IP Enhanced IGRP routes can be automatically redistributed into IGRP. If desired, you can turn off redistribution. You can also completely turn off IP Enhanced IGRP and IP IGRP on the communication server or on individual interfaces.
- Increased network width. With IP RIP, the largest possible width of your network is 15 hops. When IP Enhanced IGRP is enabled, the largest possible width is 224 hops. Because the Enhanced IGRP metric is large enough to support thousands of hops, the only barrier to expanding the network is the transport layer hop counter. Cisco works around this problem by incrementing the transport control field only when an IP packet has traversed 15 routers and the next hop to the destination was learned via Enhanced IGRP. When a RIP route is being used as the next hop to the destination, the transport control field is incremented as usual.

Enhanced IGRP offers the following features:

- Fast convergence. The DUAL algorithm allows routing information to converge as quickly as any currently available routing protocol.
- Partial updates. Enhanced IGRP sends incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table. This feature minimizes the bandwidth required for Enhanced IGRP packets.
- Less CPU usage than IGRP. This occurs because full update packets do not have to be processed each time they are received.
- Neighbor discovery mechanism. This is a simple hello mechanism used to learn about neighboring routers. It is protocol-independent.
- Variable-length subnet masks.
- Arbitrary route summarization.
- Scaling. Enhanced IGRP scales to large networks.

Enhanced IGRP has four basic components:

- Neighbor discovery/recovery
- Reliable transport protocol
- DUAL finite state machine
- Protocol-dependent modules

Neighbor discovery/recovery is the process that routers use to dynamically learn of other routers on their directly attached networks. Routers must also discover when their neighbors become unreachable or inoperative. Neighbor discovery/recovery is achieved with low overhead by periodically sending small hello packets. As long as hello packets are received, a router can determine that a neighbor is alive and functioning. Once this status is determined, the neighboring routers can exchange routing information.

The reliable transport protocol is responsible for guaranteed, ordered delivery of Enhanced IGRP packets to all neighbors. It supports intermixed transmission of multicast and unicast packets. Some Enhanced IGRP packets must be transmitted reliably and others need not be. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities, such as Ethernet, it is not necessary to send hellos reliably to all neighbors individually. Therefore, Enhanced IGRP sends a single multicast hello with an indication in the packet informing the receivers that the packet need not be acknowledged. Other types of packets, such as updates, require acknowledgment, and this is indicated in the packet. The reliable transport has a provision to send multicast packets quickly when there are unacknowledged packets pending. Doing so helps ensure that convergence time remains low in the presence of varying speed links.

The DUAL finite state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses the distance information, known as a metric, to select efficient, loop-free paths. DUAL selects routes to be inserted into a routing table based on feasible successors. A successor is a neighboring router used for packet forwarding that has a least-cost path to a destination that is guaranteed not to be part of a routing loop. When there are no feasible successors but there are neighbors advertising the destination, a recomputation must occur. This is the process whereby a new successor is determined. The amount of time it takes to recompute the route affects the convergence time. Even though the recomputation is not processor intensive, it is advantageous to avoid recomputation if it is not necessary. When a topology change occurs, DUAL will test for feasible successors. If there are feasible successors, it will use any it finds in order to avoid unnecessary recomputation.

The protocol-dependent modules are responsible for network layer protocol-specific tasks. An example is the IP Enhanced IGRP module, which is responsible for sending and receiving Enhanced IGRP packets that are encapsulated in IP. It is also responsible for parsing Enhanced IGRP packets and informing DUAL of the new information received. IP Enhanced IGRP asks DUAL to make routing decisions, but the results are stored in the IP routing table. Also, IP Enhanced IGRP is responsible for redistributing routes learned by other IP routing protocols.

Enhanced IGRP Configuration Task List

To configure IP Enhanced IGRP, complete the tasks in the following sections. At a minimum, you must enable IP Enhanced IGRP. The remaining tasks are optional.

- Enable IP Enhanced IGRP
- Transition from IGRP to Enhanced IGRP
- Configure IP Enhanced IGRP-Specific Parameters
- Monitor IP Enhanced IGRP on an IP Network

See the "IP Routing Protocol Configuration Examples" section at the end of this chapter for configuration examples.

Enable IP Enhanced IGRP

To create an IP Enhanced IGRP routing process, perform the following tasks:

Task		Command
Step 1	Enable an IP Enhanced IGRP routing process in global configuration mode.	**router eigrp** *process-number*
Step 2	Associate networks with an IP Enhanced IGRP routing process in router configuration mode.	**network** *network-number*

IP Enhanced IGRP sends updates to the interfaces in the specified network(s). If you do not specify an interface's network, it will not be advertised in any IP Enhanced IGRP update.

Transition from IGRP to Enhanced IGRP

If you have communication servers on your network that are configured for IGRP and you want to make a transition to routing Enhanced IGRP, you need to designate transition communication servers that have both IGRP and Enhanced IGRP configured. In these cases, perform the tasks as noted in the previous section, "Create the IP Enhanced IGRP Routing Process," and also read the section "Configure IGRP" earlier in this chapter. You must use the same autonomous system number in order for routes to be redistributed automatically.

Configure IP Enhanced IGRP-Specific Parameters

To configure IP Enhanced IGRP-specific parameters, perform one or more of the following tasks:

- Define unequal-cost load balancing
- Adjust the IP Enhanced IGRP metric
- Disable route summarization
- Configure summary aggregate addresses

Define Unequal-Cost Load Balancing

IP Enhanced IGRP can simultaneously use an asymmetric set of paths for a given destination. This feature is known as *unequal-cost load balancing*. Unequal-cost load balancing allows traffic to be distributed among up to four unequal-cost paths to provide greater overall throughput and reliability. Alternate path variance (the difference in desirability between the primary and alternate paths) is used to determine the feasibility of a potential route. An alternate route is feasible if the next communication server in the path is closer to the destination (has a lower metric value) than the current communication server and if the metric for the entire alternate path is within the variance. Only paths that are feasible can be used for load balancing and included in the routing table. These conditions limit the number of cases in which load balancing can occur, but ensure that the dynamics of the network will remain stable.

The following general rules apply to IP Enhanced IGRP unequal-cost load balancing:

- IP Enhanced IGRP will accept up to four paths for a given destination network.
- The local best metric must be greater than the metric learned from the next communication server; that is, the next-hop communication server must be closer (have a smaller metric value) to the destination than the local best metric.
- The alternative path metric must be within the specified *variance* of the local best metric. The multiplier times the local best metric for the destination must be greater than or equal to the metric through the next communication server.

If these conditions are met, the route is deemed feasible and can be added to the routing table.

By default, the amount of variance is set to one (equal-cost load balancing). To change the variance to define how much worse an alternate path can be before that path is disallowed, perform the following task in router configuration mode:

Task	Command
Define the variance associated with a particular path.	**variance** *multiplier*

See the "IP Routing Protocol Configuration Examples" section at the end of this chapter for an example of configuring an IP Enhanced IGRP feasible successor.

Note By using the variance feature, the communication server can balance traffic across all feasible paths and can immediately converge to a new path if one of the paths should fail.

Adjust the IP Enhanced IGRP Metric Weights

You can adjust the default behavior of IP Enhanced IGRP routing and metric computations. For example, you can tune system behavior to allow for satellite transmission. Although IP Enhanced IGRP metric defaults have been carefully selected to provide excellent operation in most networks, you can adjust the IP Enhanced IGRP metric. Adjusting IP Enhanced IGRP metric weights can dramatically affect network performance, so be careful if you adjust them.

To adjust the IP Enhanced IGRP metric weights, perform the following task in router configuration mode:

Task	Command
Adjust the IP Enhanced IGRP metric.	**metric weights** *tos k1 k2 k3 k4 k5*

Note Because of the complexity of this task, it is not recommended unless it is done with guidance from an experienced network designer.

By default, the IP Enhanced IGRP composite metric is a 32-bit quantity that is a sum of the segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media (FDDI, Ethernet, and serial lines running from 9600 bps to T1 rates), the route with the lowest metric reflects the most desirable path to a destination.

Disable Route Summarization

You can configure IP Enhanced IGRP to perform automatic summarization of subnet routes into network-level routes. For example, you can configure subnet 131.108.1.0 to be advertised as 131.108.0.0 over interfaces that have subnets of 192.31.7.0 configured. Automatic summarization is performed when there are two or more **network** router configuration commands configured for the IP Enhanced IGRP process. By default, this feature is enabled.

To disable automatic summarization, perform the following task in router configuration mode:

Task	Command
Disable automatic summarization.	**no auto-summary**

Route summarization works in conjunction with the **ip summary-address eigrp** interface configuration command, in which additional summarization can be performed. If auto-summary is in effect, there usually is no need to configure network level summaries using the **ip summary-address eigrp** command.

Configure Summary Aggregate Addresses

You can configure a summary aggregate address for a specified interface. If there are any more-specific routes in the routing table, IP Enhanced IGRP will advertise the summary address out the interface with a metric equal to the minimum of all more-specific routes.

To configure a summary aggregate address, perform the following task in interface configuration mode:

Task	Command
Configure a summary aggregate address.	**ip summary-address eigrp** *autonomous-system-number address mask*

Monitor IP Enhanced IGRP on an IP Network

You can display communication server statistics such as the contents of IP routing tables, caches, and databases. You can use the information displayed to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path that your communication server's packets are taking through the network.

To display various statistics, perform one or more of the following tasks at the EXEC prompt:

Task	Command
Delete neighbors from the neighbor table.	**clear ip eigrp neighbors** [*ip-address* | *interface*]
Display the IP Enhanced IGRP discovered neighbors.	**show ip eigrp neighbors** [*interface number*]
Display the IP Enhanced IGRP topology table for a given process.	**show ip eigrp topology** [*autonomous-system-number* | [[*ip-address*] *mask*]]
Display the number of packets sent and received for all or a specified IP Enhanced IGRP process.	**show ip eigrp traffic** [*autonomous-system-number*]
Display the parameters and current state of the active routing protocol process.	**show ip protocols**
Display the current state of the routing table.	**show ip route** [*ip-address* [*mask*]] | [*protocol* [*process-id*]]
Display the current state of the routing table in summary form.	**show ip route summary**

Configure OSPF

Open Shortest Path First (OSPF) is an IGP developed by the OSPF working group of the Internet Engineering Task Force (IETF). Designed expressly for IP networks, OSPF supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending/receiving packets.

Cisco's OSPF Implementation

Cisco's implementation of OSPF conforms to the OSPF Version 2 specifications detailed in Internet RFC 1583. The list that follows outlines key features supported in Cisco's OSPF implementation:

- Stub areas—Definition of stub areas is supported.
- Route redistribution—Routes learned via any IP routing protocol can be redistributed into any other IP routing protocol. At the intradomain level, this means that OSPF can import routes learned via IGRP and RIP. OSPF routes also can be exported into IGRP and RIP. At the interdomain level, OSPF can import routes learned via EGP and BGP. OSPF routes can be exported into EGP and BGP.
- Authentication—Authentication among neighboring communication servers within an area is supported.
- Routing interface parameters—Configurable parameters supported include interface output cost, retransmission interval, interface transmit delay, communication server priority, communication server "dead" and hello intervals, and authentication key.
- Virtual links—Virtual links are supported.

Note In order to take advantage of the OSPF stub area support, *default routing* must be used in the stub area.

OSPF Configuration Task List

OSPF typically requires coordination among many internal communication servers, communication servers acting as *area border routers* (routers connected to multiple areas), and communication servers acting as autonomous system boundary routers. At a minimum, OSPF-based communication servers can be configured with all default parameter values, no authentication, and interfaces assigned to areas. If you intend to customize your environment, you must ensure coordinated configurations of all communication servers.

To configure OSPF, complete the tasks in the following sections. Enabling OSPF is mandatory; the other tasks are optional but might be required for your application.

- Enable OSPF
- Configure OSPF Interface Parameters
- Configure OSPF over Different Physical Networks
- Configure OSPF Area Parameters
- Configure Route Summarization between OSPF Areas
- Create Virtual Links
- Generate a Default Route
- Configure Lookup of DNS Names
- Force the Router ID Choice with a Loopback Interface

In addition, you can specify route redistribution; see the "Redistribute Routing Information" section later in this chapter for information on how to configure route redistribution.

Enable OSPF

As with other routing protocols, enabling OSPF requires that you create an OSPF routing process, specify the range of IP addresses to be associated with the routing process, and assign area IDs to be associated with that range of IP addresses. Perform the following tasks, starting in global configuration mode:

Task		Command
Step 1	Enable OSPF routing, which places you in router configuration mode.	**router ospf** *process-id*
Step 2	Define an interface on which OSPF runs and define the area ID for that interface.	**network** *address wildcard-mask* **area** *area-id*

Configure OSPF Interface Parameters

Our OSPF implementation allows you to alter certain interface-specific OSPF parameters, as needed.You are not required to alter any of these parameters, but some interface parameters must be consistent across all communication servers in an attached network. Therefore, be sure that if you do configure any of these parameters, the configurations for all communication servers on your network have compatible values.

In interface configuration mode, specify any of the following interface parameters as needed for your network:

Task	Command
Explicitly specify the cost of sending a packet on an OSPF interface.	**ip ospf cost** *cost*
Specify the number of seconds between link state advertisement retransmissions for adjacencies belonging to an OSPF interface.	**ip ospf retransmit-interval** *seconds*
Set the estimated number of seconds it takes to transmit a link state update packet on an OSPF interface.	**ip ospf transmit-delay** *seconds*
Set communication server priority to help determine the OSPF designated communication server for a network.	**ip ospf priority** *number*
Specify the length of time, in seconds, between the hello packets that a communication server sends on an OSPF interface.	**ip ospf hello-interval** *seconds*
Set the number of seconds that a communication server's hello packets must not have been seen before its neighbors declare the OSPF communication server down.	**ip ospf dead-interval** *seconds*
Assign a specific password to be used by neighboring OSPF communication servers on a network segment that is using OSPF's simple password authentication.	**ip ospf authentication-key** *password*

Configure OSPF over Different Physical Networks

OSPF classifies different media into three types of networks by default:

- Broadcast networks (Ethernet, Token Ring)
- Nonbroadcast, multiaccess networks (SMDS, Frame Relay, X.25)
- Point-to-point networks (HDLC, PPP)

You can configure your network as either a broadcast or a nonbroadcast multiaccess network.

X.25 and Frame Relay provide an optional broadcast capability that can be configured in the map to allow OSPF to run as a broadcast network. See the **x25 map** and **frame-relay map** command descriptions in the *Access and Communication Servers Command Reference* publication for more detail.

Configure Your OSPF Network Type

You have the choice of configuring your OSPF network type to either broadcast or nonbroadcast multiaccess, regardless of the default media type. Using this feature, you can configure broadcast networks as nonbroadcast multiaccess networks when, for example, you have communication servers in your network that do not support multicast addressing. You also can configure nonbroadcast multiaccess networks, such as X.25, Frame Relay, and SMDS, as broadcast networks. This feature saves you from having to configure neighbors, as described in "Configure OSPF for Nonbroadcast Networks," following this section.

To configure your OSPF network type, perform the following task in interface configuration mode:

Task	Command
Configure the OSPF network type for a specified interface.	**ip ospf network** {**broadcast** | **non-broadcast**}

Configure OSPF for Nonbroadcast Networks

Because there might be many communication servers attached to an OSPF network, a *designated communication server* is selected for the network. It is necessary to use special configuration parameters in the designated communication server selection if broadcast capability is not configured.

These parameters need only be configured in those communication servers that are themselves eligible to become the designated communication server or backup designated communication server (in other words, communication servers with a nonzero communication server priority value).

To configure communication servers that interconnect to nonbroadcast networks, perform the following task in router configuration mode

Task	Command
Configure communication servers interconnecting to nonbroadcast networks.	**neighbor** *ip-address* [**priority** *number*] [**poll-interval** *seconds*]

You can specify the following neighbor parameters, as required:

- Priority for a neighboring communication server
- Nonbroadcast poll interval
- Interface through which the neighbor is reachable

Configure OSPF Area Parameters

Our OSPF software allows you to configure several area parameters. These area parameters, shown in the following table, include authentication, defining stub areas, and assigning specific costs to the default summary route. *Authentication* allows password-based protection against unauthorized access to an area. *Stub areas* are areas into which information on external routes is not sent. Instead, there is a default external route generated by the area border router into the stub area for destinations outside the autonomous system.

In router configuration mode, specify any of the following area parameters as needed for your network:

Task	Command
Enable authentication for an OSPF area.	**area** *area-id* **authentication**
Define an area to be a stub area.	**area** *area-id* **stub**
Assign a specific cost to the default summary route used for the stub area.	**area** *area-id* **default-cost** *cost*

Configure Route Summarization between OSPF Areas

Route summarization is the consolidation of advertised addresses. This feature causes a single summary route to be advertised to other areas by a communication server acting as an area border router. In OSPF, a communication server acting as an area border router will advertise networks in one area into another area. If the network numbers in an area are assigned in a way such that they are contiguous, you can configure the communication server acting as an area border router to advertise a summary route that covers all the individual networks within the area that fall into the specified range.

To specify an address range, perform the following task in router configuration mode:

Task	Command
Specify an address range for which a single route will be advertised.	**area** *area-id* **range** *address mask*

Create Virtual Links

In OSPF, all areas must be connected to a backbone area. If there is a break in backbone continuity, or the backbone is purposefully partitioned, you can establish a *virtual link*. The two end points of a virtual link are communication servers acting as area border routers. The virtual link must be configured in both communication servers. The configuration information in each communication server consists of the other virtual endpoint (the other Area Border Router), and the nonbackbone area that the two communication servers have in common (called the *transit area*). Note that virtual links cannot be configured through stub areas.

To establish a virtual link, perform the following task in router configuration mode:

Task	Command
Establish a virtual link.	**area** *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] **transmit-delay** *seconds*] [**dead-interval** *seconds*] [**authentication-key** *password*]

To display information about virtual links, use the **show ip ospf virtual-links** EXEC command. To display the router ID of an OSPF router, use the **show ip ospf** EXEC command.

Generate a Default Route

You can force an autonomous system boundary communication server to generate a default route into an OSPF routing domain. Whenever you specifically configure redistribution of routes into an OSPF routing domain, the communication server automatically becomes an autonomous system boundary communication server. However, an autonomous system boundary communication server does not, by default, generate a *default route* into the OSPF routing domain.

To force the communication server acting as an autonomous system boundary router to generate a default route, perform the following task in router configuration mode:

Task	Command
Force the autonomous system boundary router to generate a default route into the OSPF routing domain.	**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*]

See also the discussion of redistribution of routes in the "Configure Routing Protocol-Independent Features" section later in this chapter.

Configure Lookup of DNS Names

You can configure OSPF to look up Domain Name System (DNS) names for use in all OSPF **show** command displays. This feature makes it easier to identify a communication server, because it is displayed by name rather than by its communication server ID or neighbor ID.

To configure DNS name lookup, perform the following task in global configuration mode:

Task	Command
Configure DNS name lookup.	**ip ospf-name-lookup**

Force the Router ID Choice with a Loopback Interface

OSPF uses the largest IP address configured on the communication server's interfaces as its router ID. If the interface associated with this IP address is ever brought down, or if the address is removed, the OSPF process must recalculate a new router ID and resend all of its routing information out its interfaces.

If a loopback interface is configured with an IP address, the communication server will use this IP address as its router ID, even if other interfaces have larger IP addresses. Because loopback interfaces never go down, greater stability in the routing table is achieved.

OSPF automatically prefers a loopback interface over any other kind, and it chooses the first loopback interface found. If no loopback interfaces are present, the highest IP address in the communication server is chosen. You cannot tell OSPF to use any particular interface.

To configure an IP address on a loopback interface, perform the following tasks, starting in global configuration mode:

Task		Command
Step 1	Create a loopback interface, which places you in interface configuration mode.	**interface loopback 0**[1]
Step 2	Assign an IP address to this interface.	**ip address** *address mask*

[1] This command is documented in the "Interface Commands" chapter of the *Access and Communication Servers Command Reference* publication.

Configure RIP

The Routing Information Protocol (RIP) is a relatively old but still commonly used IGP created for use in small, homogeneous networks. It is a classical distance-vector routing protocol.

RIP uses broadcast User Datagram Protocol (UDP) data packets to exchange routing information. Each communication server sends routing information updates every 30 seconds; this process is termed *advertising.* If a communication server does not receive an update from another communication server for 180 seconds or more, it marks the routes served by the nonupdating communication server as being unusable. If there is still no update after 240 seconds, the communication server removes all routing table entries for the nonupdating communication server.

The measure, or metric, that RIP uses to rate the value of different routes is the *hop count.* The hop count is the number of communication servers that can be traversed in a route. A directly connected network has a metric of zero; an unreachable network has a metric of 16. This small range of metrics makes RIP unsuitable as a routing protocol for large networks. If the communication server has a default network path, RIP advertises a route that links the communication server to the pseudonetwork 0.0.0.0. The network 0.0.0.0 does not exist; RIP treats 0.0.0.0 as a network to implement the default routing feature. Our systems will advertise the default network if a default was learned by RIP or if the communication server has a gateway of last resort and RIP is configured with a default metric.

RIP sends updates to the interfaces in the specified networks. If an interface's network is not specified, it will not be advertised in any RIP update.

For information about filtering RIP information, see the "Filter Routing Information" section later in this chapter. RIP is documented in RFC 1058.

To configure RIP, perform the following tasks, starting in global configuration mode:

Task		Command
Step 1	Enable a RIP routing process, which places you in router configuration mode.	**router rip**
Step 2	Associate a network with a RIP routing process.	**network** *network-number*

Running IGRP and RIP Concurrently

It is possible to run IGRP and RIP concurrently. The IGRP information will override the RIP information by default because of IGRP's administrative distance.

However, running IGRP and RIP concurrently does not work well when the network topology changes. Because IGRP and RIP have different update timers and because they require different amounts of time to propagate routing updates, one part of the network will end up believing IGRP routes and another part will end up believing RIP routes. This will result in routing loops. Even though these loops do not exist for very long, the time to live (TTL) will quickly reach zero, and ICMP will send a "TTL exceeded" message. This message will cause most applications to stop attempting network connections.

Validate Source IP Addresses

To disable the default function that validates the source IP addresses of incoming routing updates, perform the following task in router configuration mode:

Task	Command
Disable the checking and validation of the source IP address of incoming routing updates.	**no validate-update-source**

Allow Point-to-Point Updates for RIP

Because RIP is normally a broadcast protocol, in order for RIP routing updates to reach point-to-point or nonbroadcast networks, you must configure the communication server to permit this exchange of routing information.

You configure the communication server to permit this exchange of routing information by performing the following task in router configuration mode:

Task	Command
Define a neighboring communication server with which to exchange point-to-point routing information.	**neighbor** *ip-address*

To control the set of interfaces that you want to exchange routing updates with, you can disable the sending of routing updates on specified interfaces by configuring the **passive-interface** command. See the discussion on filtering in the “Filter Routing Information” section later in this chapter.

Configure BGP

The Border Gateway Protocol (BGP), as defined in RFCs 1163 and 1267, allows you to set up an interdomain routing system that automatically guarantees the loop-free exchange of routing information between autonomous systems.

Cisco’s BGP Implementation

In BGP, each route consists of a network number, a list of autonomous systems that information has passed through (called the *autonomous system path* or *AS path*), and a list of other *path attributes*. We support BGP Versions 2, 3, and 4. This section describes our implementation of BGP.

The primary function of a BGP system is to exchange network reachability information with other BGP systems, including information about the list of AS paths. This information can be used to construct a graph of autonomous system connectivity from which routing loops can be pruned and with which autonomous system-level policy decisions can be enforced.

You can configure the value for the multiple exit discriminator (MULTI_EXIT_DISC, or MED) metric attribute using route maps. (The name of this metric for BGP Versions 2 and 3 is INTER_AS.) When an update is sent to an IBGP peer, the MED will be passed along without any change. This will enable all the peers in the same autonomous system to make a consistent path selection.

A third-party next-hop communication server address is used in the NEXT_HOP attribute, regardless of the autonomous system path of that third-party communication server. The communication server automatically calculates the value for this attribute.

Transitive, optional path attributes are passed along to other BGP-speaking communication servers. The current BGP implementation does not generate such attributes.

BGP Version 4 (BGP4) supports classless interdomain routing (CIDR), which lets you reduce the size of your routing tables by creating aggregate routes, resulting in *supernets*. CIDR eliminates the concept of network classes within BGP and supports the advertising of IP prefixes. CIDR routes can be carried by OSPF.

See the “Using Route Maps with BGP” section later in this chapter for examples of how to use route maps to redistribute BGP4 routes.

How BGP Selects Paths

The BGP process selects a single autonomous system path to use and to pass along to other BGP-speaking communication servers. Cisco’s BGP implementation has a reasonable set of factory defaults that can be overridden by administrative weights. The algorithm for path selection is as follows:

- If the next hop is inaccessible, do not consider it.
- Consider larger BGP administrative weights first.
- If the routes have the same weight, consider the route with higher local preference.
- If the routes have the same local preference, prefer the route that the specified communication server originated.
- If no route was originated, prefer the shorter AS path.
- If the AS paths are of the same length, prefer external paths over internal paths.
- If all paths are external, prefer the lowest origin code (IGP < EGP < INCOMPLETE).
- If origin codes are the same and all the paths are from the same AS, prefer the path with the lowest MULTI_EXIT_DISC METRIC. A missing metric is treated as zero.
- If IGP synchronization is disabled and only internal paths remain, prefer the path through the closest neighbor.
- Prefer the route with the lowest IP address value for the BGP router ID.

BGP Configuration Task List

To configure BGP, complete the tasks in the following sections:

- Enable BGP Routing
- Configure BGP Neighbors
- Reset BGP Connections

The tasks in the following sections are optional:

- Configure BGP Route Filtering by Neighbor
- Configure BGP Path Filtering by Neighbor
- Disable Next-Hop Processing on BGP Updates
- Configure BGP Administrative Weights
- Configure BGP Interactions with IGPs
- Configure Miscellaneous BGP Parameters

Enable BGP Routing

To enable BGP routing, establish a BGP routing process on the communication server and specify those networks within the communication server's autonomous system to be advertised. Perform the following steps. There is a limit of 200 networks that can be advertised from one autonomous system.

Task		Command
Step 1	Enter global configuration mode.	See Table 2-1 earlier in this manual.
Step 2	Enable a BGP routing process, which places you in router configuration mode.	**router bgp** *autonomous-system*
Step 3	Flag a network as local to this autonomous system.	**network** *network-number* **mask** *network-mask*

Note For Exterior Gateway Protocols, a reference to an IP network from the **network** router configuration command only controls which networks are advertised. This is in contrast to Interior Gateway Protocols, such as IGRP, which also use the **network** command to determine where to send updates.

Configure BGP Neighbors

Like other Exterior Gateway Protocols (EGPs), BGP must completely understand the relationships it has with its neighbors. BGP supports two kinds of neighbors: internal and external. Internal neighbors are in the same autonomous system path; external neighbors are in different autonomous system paths. Normally, external neighbors are adjacent to each other and share a subnet, while internal neighbors may be anywhere in the same autonomous system.

To configure BGP neighbors, perform the following task in router configuration mode:

Task	Command
Specify a BGP neighbor.	**neighbor** *ip-address* **remote-as** *number*

You also can configure neighbor templates that use a word argument rather than an IP address to configure BGP neighbors. This is an advanced feature requiring a well-thought-out network architecture. Do not use this feature without thoroughly understanding its application.

Perform the following tasks in router configuration mode to configure BGP neighbor templates:

Task	Command
Support anonymous neighbor peers by configuring a neighbor template.	**neighbor** *template-name* **neighbor-list** *access-list-number*
Treat neighbors that have been accepted by a template as if they were configured by hand.	**neighbor** *template-name* **configure-neighbors**

Reset BGP Connections

Once you have defined two communication servers to be BGP neighbors, they will form a BGP connection and exchange routing information. If you subsequently change a BGP filter, weight, distance, version, or timer, or make a similar configuration change, you need to reset BGP connections for the configuration change to take effect. Perform either of the following tasks in EXEC mode to reset BGP connections:

Task	Command
Reset a particular BGP connection.	**clear ip bgp** *address*
Reset all BGP connections.	**clear ip bgp ***

To automatically reset BGP sessions, perform the following task in router configuration mode:

Task	Command
Automatically reset BGP sessions of any directly adjacent external peer if the link used to reach it goes down.	**bgp fast-external-fallover**

Configure BGP Route Filtering by Neighbor

If you want to restrict the routing information that the communication server learns or advertises, you can filter BGP routing updates to and from particular neighbors. To do this, define an access list and apply it to the updates. Distribute-list filters are applied to network numbers and not AS paths.

To filter BGP routing updates, perform the following task in router configuration mode:

Task	Command
Filter BGP routing updates to/from neighbors as specified in an access list.	**neighbor** *ip-address* **distribute-list** *access-list-number* {**in** | **out**}

Configure BGP Path Filtering by Neighbor

In addition to filtering routing updates based on network numbers, you can specify an access list filter on both incoming and outbound updates based on the BGP AS paths. Each filter is an access list based on regular expressions. To do this, define an AS path access list and apply it to updates to and from particular neighbors. See the “Regular Expressions” appendix in the *Access and Communication Servers Command Reference* publication for more information on forming regular expressions.

Perform the following tasks to configure BGP path filtering:

Task		Command
Step 1	Enter global configuration mode.	See Table 2-1 earlier in this chapter.
Step 2	Define a BGP-related access list.	**ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expression*
Step 3	Enter router configuration mode.	See Table 2-1 earlier in this manual.
Step 4	Establish a BGP filter.	**neighbor** *ip-address* **filter-list** *access-list-number* {**in** | **out** | **weight** *weight*}

Disable Next-Hop Processing on BGP Updates

You can configure the communication server to disable next-hop processing for BGP updates to a neighbor. This is useful in non-meshed networks such as Frame Relay or X.25 where BGP neighbors might not have direct access to all other neighbors on the same IP subnet.

To disable next-hop processing, perform the following task in router configuration mode:

Task	Command
Disable next-hop processing on BGP updates to a neighbor.	**neighbor** *ip-address* **next-hop-self**

Configure BGP Administrative Weights

An administrative weight is a number that you can assign to a path so that you can control the path selection process.The administrative weight is local to the communication server. A weight can be a number from 0 to 65535. Paths that the communication server originates have weight 32768 by default, other paths have weight zero. If you have particular neighbors that you want to prefer for most of your traffic, you can assign a weight to all paths learned from a neighbor.

Perform the following task in router configuration mode to configure BGP administrative weights:

Task	Command
Specify a weight for all paths from a neighbor.	**neighbor** *ip-address* **weight** *weight*

In addition, you can assign weights based on autonomous system path access lists. A given weight becomes the weight of the path if the AS path is accepted by the access list. Any number of weight filters are allowed.

Perform the following tasks to assign weights based on AS path access lists:

Task		Command
Step 1	Enter global configuration mode.	See Table 2-1 earlier in this manual.
Step 2	Define a BGP-related access list.	**ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expression*
Step 3	Enter router configuration mode.	See Table 2-1 earlier in this manual.
Step 4	Configure set administrative weight on all incoming routes matching an autonomous system path filter.	**neighbor** *ip-address* **filter-list** *access-list-number* **weight** *weight*

Configure BGP Interactions with IGPs

If your autonomous system will be passing traffic through it from another autonomous system to a third autonomous system, it is very important that your autonomous system be consistent about the routes that it advertises. For example, if your BGP were to advertise a route before all communication servers in your network had learned about the route through your IGP, your autonomous system could receive traffic that some communication servers cannot yet route. To prevent this from happening, BGP must wait until the IGP has propagated routing information across your autonomous system. This causes BGP to be *synchronized* with the IGP. Synchronization is enabled by default.

In some cases, you do not need synchronization. If you will not be passing traffic from a different autonomous system through your autonomous system, or if all communication servers in your autonomous system will be running BGP, you can disable synchronization. Disabling this feature can allow you to carry fewer routes in your IGP, increase the number of paths that BGP can select, and allow BGP to converge more quickly, however you must run BGP on all communication servers in your autonomous system and there must be a full IBGP connectivity mesh between these communication servers. To disable synchronization, perform the following task in router configuration mode:

Task	Command
Disable synchronization between BGP and an IGP.	**no synchronization**

When you disable synchronization, you should also clear BGP routes using the **clear ip bgp** command.

In general, you will not want to redistribute most BGP routes into your IGP. A common design is to redistribute one or two routes and to make them exterior routes in IGRP or have your BGP speakers generate a default route for your autonomous system. When redistributing from BGP into IGP, only the routes learned using EBGP get redistributed.

In most circumstances, you also will not want to redistribute your IGP into BGP. Just list the networks in your autonomous system with **network** router configuration commands and your networks will be advertised. Networks that are listed this way are referred to as *local networks* and have a BGP origin attribute of “IGP.” They must appear in the main IP routing table and can have any source; for example, they can be directly connected or learned via an IGP. The BGP routing process periodically scans the main IP routing table to detect the presence or absence of local networks, updating the BGP routing table as appropriate.

If you do perform redistribution into BGP, you must be very careful about the routes that can be in your IGP, especially if the routes were redistributed from BGP into the IGP elsewhere. This creates a situation where BGP is potentially injecting information into the IGP and then sending such information back into BGP and vice versa.

Networks that are redistributed into BGP from the EGP protocol will be given the BGP origin attribute “EGP.” Other networks that are redistributed into BGP will have the BGP origin attribute of “incomplete.” The origin attribute in our implementation is only used in the path selection process.

See the “IP Routing Protocol Configuration Examples” section at the end of this chapter for an example of synchronization.

Configure Aggregate Addresses

Classless Interdomain Routing (CIDR) lets you create aggregate routes, or *supernets*, to minimize the size of routing tables. You can configure aggregate routes in BGP either by redistributing an aggregate route into BGP or by using the conditional aggregation feature described in the next task table.

To create an aggregate address in the routing table, perform one or more of the following tasks in router configuration mode:

Task	Command
Create an aggregate entry in the BGP routing table. Advertise general information.	**aggregate-address** *address mask*
Advertised information will include all elements of all paths.	**aggregate-address** *address mask* **as-set**
Advertise summary addresses only.	**aggregate-address** *address mask* **summary-only**
Suppress selected more-specific routes.	**aggregate-address** *address mask* **suppress-map** *map-tag*

Specify Automatic Summarization of Network Numbers

To disable automatic network number summarization when redistributing to BGP from IGPs, perform the following task in router configuration mode:

Task	Command
Disable automatic network summarization.	**no auto-summary**

Configure Miscellaneous BGP Parameters

You can adjust several miscellaneous BGP parameters, as indicated in the following subsections.

Configure Neighbor Options

If you would like to provide BGP routing information to a large number of neighbors, you can configure BGP to accept neighbors based on an access list. If a neighbor attempts to initiate a BGP connection, its address must be accepted by the access list for the connection to be accepted. If you do this, the communication server will not attempt to initiate a BGP connection to these neighbors, so the neighbors must be explicitly configured to initiate the BGP connection. If no access list is specified, all connections are accepted.

If a neighbor is running a different version of BGP, you should configure the version of BGP that the neighbor is speaking.

External BGP peers normally must reside on a directly connected network. Sometimes it is useful to relax this restriction in order to test BGP; do so by specifying the **neighbor ebgp-multihop** command

For internal BGP, you might want to allow your BGP connections to stay up regardless of which interfaces are available on the communication server. To do this, you first configure a *loopback* interface and assign it an IP address. Next, configure the BGP update source to be the loopback interface. Finally, configure your neighbor to use the address on the loopback interface.

You can also set the minimum interval of time between BGP routing updates and apply a route map to incoming and outgoing routes.

Configure any of the following neighbor options in router configuration mode:

Task	Command
Specify an access list of BGP neighbors.	**neighbor any** [*access-list-number*]
Specify the BGP version to use when communicating with a neighbor.	**neighbor** *ip-address* **version** *value*

Allow internal BGP sessions to use any operational interface for TCP connections.	**neighbor** *ip-address* **update-source** *interface*
Allow BGP sessions even when the neighbor is not on a directly connected segment.	**neighbor** *ip-address* **ebgp-multihop**
Set the minimum interval between sending BGP routing updates.	**neighbor** {*address* | *tag*} **advertisement-interval** *seconds*
Apply a route map to incoming or outgoing routes.	**neighbor** {*address* | *tag*} **route-map** *route-map-name* {**in** | **out**}

See the "IP Routing Protocol Configuration Examples" section at the end of this chapter for examples of configuring BGP neighbor options.

Set the Network Weight

To set the absolute weight for a network, perform the following task in router configuration mode:

Task	Command
Set the weight for a network.	**network** *address* **weight** *weight*

Indicate Backdoor Routes

You can indicate which networks are reachable using a *backdoor* route that the border communication server should use. A backdoor network is treated as a local network, except that it is not advertised. To configure backdoor routes, perform the following task in router configuration mode:

Task	Command
Indicate reachable networks through backdoor routes.	**network** *address* **backdoor**

Update IP Routing Table

To modify metric and tag information when the IP routing table is updated with BGP learned routes, perform the following task in router configuration mode:

Task	Command
Apply route-map to routes when updating the IP routing table.	**table-map** *route-map-name*

Set Administrative Distance

Administrative distance is a measure of the ability of a routing protocol to provide optimal routes. BGP uses three different administrative distances—external, internal, and local. Routes learned through external BGP are given the external distance, routes learned with internal BGP are given the internal distance, and routes that are part of this autonomous system are given the local distance. To assign a BGP administrative distance, perform the following task in router configuration mode:

Task	Command
Assign a BGP administrative distance.	**distance bgp** *external-distance internal-distance local-distance*

Changing the administrative distance of BGP routes is considered dangerous and generally is not recommended. The external distance should be lower than any other dynamic routing protocol, and the internal and local distances should be higher than any other dynamic routing protocol.

Adjust BGP Timers

BGP uses certain timers to control periodic activities such as the sending of keepalive messages and the interval after not receiving a keepalive message after which the communication server declares a peer dead. You can adjust these timers. When a connection is started, BGP will negotiate the hold time with the neighbor. The smaller of the two hold times will be chosen. The keepalive timer is then set based on the negotiated holdtime and the configured keepalive time. To adjust BGP timers, perform the following task in router configuration mode:

Task	Command
Adjust BGP timers.	**timers bgp** *keepalive holdtime*

Configure the MULTI_EXIT_DISC METRIC

BGP uses the MULTI_EXIT_DISC METRIC as a hint to external neighbors about preferred paths. (The name of this metric for BGP Versions 2 and 3 is INTER_AS.) If you have a communication server that traffic should avoid, you can configure that communication server with a higher MULTI_EXIT_DISC METRIC. Doing this sets the MULTI_EXIT_DISC METRIC on all paths that the communication server advertises. Perform the following task in router configuration mode:

Task	Command
Set a MULTI_EXIT_DISC METRIC.	**default-metric** *number*

Change the Local Preference Value

You can define a particular path as more or less preferable than other paths by changing the default local preference value of 100. To assign a different default local preference value, perform the following task in router configuration mode:

Task	Command
Change the default local preference value.	**bgp default local-preference** *value*

You can use route maps to change the default local preference of specific paths. See the “Using Route Maps with BGP” section for examples.

Redistribute Network 0.0.0.0

To redistribute network 0.0.0.0, perform the following task in router configuration mode:

Task	Command
Allow the redistribution of network 0.0.0.0 into BGP.	**default-information originate**

Configure EGP

The Exterior Gateway Protocol (EGP), specified in RFC 904, is an older EGP used for communicating with certain communication servers in the Defense Data Network (DDN) that the U.S. Department of Defense designates as *core communication servers*. EGP also was used extensively when attaching to the National Science Foundation Network (NSFnet) and other large backbone networks.

An exterior communication server uses EGP to advertise its knowledge of routes to networks within its autonomous system. It sends these advertisements to the core communication servers, which then readvertise their collected routing information to the exterior communication server. A neighbor or peer communication server is any communication server with which the communication server communicates using EGP.

Cisco's EGP Implementation

Cisco's implementation of EGP supports three primary functions, as specified in RFC 904:

- Routers running EGP establish a set of neighbors, and these neighbors share reachability information.
- EGP communication servers poll their neighbors periodically to see if they are "alive."
- EGP communication servers send update messages containing information about the reachability of networks within their autonomous systems.

EGP Configuration Task List

To enable EGP routing on your communication server, complete the tasks in the following sections. The tasks in the first two sections are mandatory; the tasks in the other sections are optional.

- Enable EGP Routing
- Configure EGP Neighbor Relationships
- Adjust EGP Timers
- Configure Third-Party EGP Support
- Configure Backup Communication Servers
- Configure Default Routes
- Define a Central Routing Information Manager (Core Gateway)

Enable EGP Routing

To enable EGP routing, you must specify an autonomous system number, generate an EGP routing process, and indicate the networks for which the EGP process will operate.

Perform these required tasks in the order given as shown in the following table:

Task		Command
Step 1	Enter global configuration mode.	See Table 2-1 earlier in this manual.
Step 2	Specify the autonomous system that the communication server resides in for EGP.	**autonomous-system** *local-as*
Step 3	Enable an EGP routing process, which places you in router configuration mode.	**router egp** *remote-as*
Step 4	Specify a network to be advertised to the EGP peers of an EGP routing process.	**network** *network-number*

Note For Exterior Gateway Protocols, a reference to an IP network from the **network** router configuration command that is learned by another routing protocol does not require a **redistribute** router configuration command. This is in contrast to interior gateway protocols, such as IGRP, which require the use of the **redistribute** command.

Configure EGP Neighbor Relationships

A communication server using EGP cannot dynamically determine its neighbor or peer communication servers. You must therefore provide a list of neighbor communication servers.

To specify an EGP neighbor, perform the following task in router configuration mode:

Task	Command
Specify an EGP neighbor.	**neighbor** *ip-address*

Adjust EGP Timers

The EGP timers consist of a hello timer and a poll time interval timer. The hello timer determines the frequency in seconds with which the communication server sends hello messages to its peer. The poll time specifies how frequently to exchange updates. Our implementation of EGP allows these timers to be adjusted by the user.

To adjust EGP timers, perform the following task in router configuration mode:

Task	Command
Adjust EGP timers.	**timers egp** *hello polltime*

Configure Third-Party EGP Support

EGP supports a *third-party mechanism* in which EGP tells an EGP peer that another communication server (the third party) on the shared network is the appropriate communication server for some set of destinations.

To specify third-party communication servers in updates, perform the following task in router configuration mode:

Task	Command
Specify a third-party through which certain destinations can be achieved.	**neighbor** *ip-address* **third-party** *third-party-ip-address* [**internal** | **external**]

See the "IP Routing Protocol Configuration Examples" section later in this chapter for an example of configuring third-party EGP support.

Configure Backup Communication Servers

You might want to provide backup in the event of site failure by having a second communication server belonging to a different autonomous system act as a backup to the EGP communication server for your autonomous system. To differentiate between the primary and secondary EGP communication servers, the two communication servers will advertise network routes with differing EGP distances or metrics. A network with a low metric is generally favored over a network with a high metric.

Networks declared as local are always announced with a metric of zero. Networks that are redistributed will be announced with a metric specified by the user. If no metric is specified, redistributed routes will be advertised with a metric of three. All redistributed networks will be advertised with the same metric. The redistributed networks can be learned from static or dynamic routes. See also the "Redistribute Routing Information" section later in this chapter.

See the "IP Routing Protocol Configuration Examples" section later in this chapter for an example of configuring backup communication servers.

Configure Default Routes

You also can designate network 0.0.0.0 as a default route. If the next hop for the default route can be advertised as a third party, it will be included as a third party.

To enable the use of default EGP routes, perform the following task in router configuration mode:

Task	Command
Configure EGP to generate a default route.	**default-information originate**

Define a Central Routing Information Manager (Core Gateway)

Normally, an EGP process expects to communicate with neighbors from a single autonomous system. Because all neighbors are in the same autonomous system, the EGP process assumes that these neighbors all have consistent internal information. Therefore, if the EGP process is informed about a route from one of its neighbors, it will not send it out to other neighbors.

With *core EGP*, the assumption is that all neighbors are from different autonomous systems, and all have inconsistent information. In this case, the EGP process distributes routes from one neighbor to all others (but not back to the originator). This allows the EGP process to be a central clearinghouse for information with a single, central manager of routing information (sometimes called a *core gateway*). To this end, one core gateway process can be configured for each communication server.

To define a core gateway process, perform the following steps in the order in which they appear:

Task		Command
Step 1	Enter global configuration mode.	See Table 2-1 earlier in this manual.
Step 2	Allow a specific communication server to act as a peer with any reachable autonomous system.	**router egp 0**
Step 3	Define how an EGP process determines which neighbors will be treated as peers. or	**neighbor any** [*access-list-number*]
	Allow the specified address to be used as the next hop in EGP advertisements.	**neighbor any third-party** *ip-address* [**internal** | **external**]

The EGP process defined in this way can act as a peer with any autonomous system, and information is interchanged freely between autonomous systems.

See the "IP Routing Protocol Configuration Examples" section at the end of this chapter for an example of configuring an EGP core gateway.

Note Split horizon is performed only on a *per-gateway* basis (in other words, if an external communication server informs the communication server about a specific network, and that communication server is the *best* path, the communication server will *not* inform the originating external communication server about that path). Our communication servers can also perform per-gateway split horizon on third-party updates.

Configure GDP

The Gateway Discovery Protocol (GDP), designed by Cisco to address customer needs, allows hosts to dynamically detect the arrival of new communication servers, as well as determine when a communication server goes down. You must have host software to take advantage of this protocol.

For ease of implementation on a variety of host software, GDP is based on the User Datagram Protocol (UDP). The UDP source and destination ports of GDP datagrams are both set to 1997 (decimal).

There are two types of GDP messages: *report* and *query.* On broadcast media, report message packets are periodically sent to the IP broadcast address announcing that the communication server is present and functioning. By listening for these report packets, a host can detect a vanishing or appearing communication server. If a host issues a query packet to the broadcast address, the communication servers each respond with a report sent to the host's IP address. On nonbroadcast media, communication servers send report message packets only in response to query message packets. The protocol provides a mechanism for limiting the rate at which query messages are sent on nonbroadcast media.

Figure 18-2 shows the format of the GDP report message packet format. A GDP query message packet has a similar format, except that the count field is always zero and no address information is present.

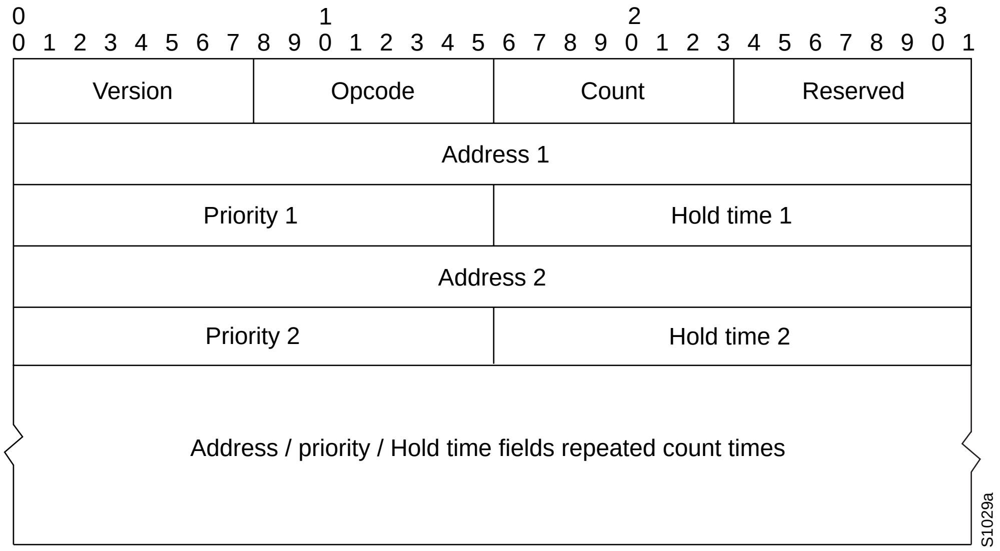

Figure 18-2 GDP Report Message Packet Format

The fields in the report and query messages are as follows:

- Version—8-bit field containing the protocol version number. The current GDP version number is 1. If an unrecognized version number is found, the GDP message must be ignored.
- Opcode—8-bit field that describes the GDP message type. Unrecognized opcodes must be ignored. Opcode 1 is a report message and opcode 2 is a query message.
- Count—8-bit field that contains the number of address, priority, and hold time tuples in this message. A query message has a Count field value of zero. A report message has a count field value of 1 or greater.
- Reserved—8-bit reserved field; it must be set to zero.
- Address—32-bit fields containing the IP address of a communication server on the local network segment. There are no other restrictions on this address. If a host encounters an address that it believes is not on its local network segment, it should ignore that address.
- Priority—16-bit fields that indicate the relative quality of the associated address. The numerically larger the value in the priority field, the better the address should be considered.
- Hold Time—16-bit fields. On broadcast media, the number of seconds the associated address should be used as a communication server without hearing further report messages regarding that address. On nonbroadcast media such as X.25, this is the number of seconds the requester should wait before sending another query message.

Numerous actions can be taken by the host software listening to GDP packets. One possibility is to flush the host’s ARP cache whenever a communication server appears or disappears. A more complex possibility is to update a host routing table based on the coming and going of communication servers. The particular course of action taken depends on the host software and your network requirements.

To enable GDP routing and other optional GDP tasks as required for your network, perform the following tasks in interface configuration mode:

Task	Command
Enable GDP processing on an interface.	**ip gdp**
Set the relative quality of the associated address.	**ip gdp priority** *number*
Set the GDP report period.	**ip gdp reporttime** *seconds*
Set the length of time the associated address should be used as a communication server without hearing further report messages regarding that address.	**ip gdp holdtime** *seconds*

Configure IRDP

Like GDP, the ICMP Router Discovery Protocol (IRDP) allows hosts to locate communication servers. When operating as a client, communication server discovery packets are generated, and when operating as a host, communication server discovery packets are received.

The only required task for configuring IRDP routing on a specified interface is to enable IRDP processing on an interface. Perform the following task in interface configuration mode:

Task	Command
Enable IRDP processing on an interface.	**ip irdp**

When you enable IRDP processing, the default parameters will apply. You can optionally change any of these IRDP parameters. Perform the following tasks in interface configuration mode:

Task	Command
Send IRDP advertisements to the all-systems multicast address (224.0.0.1) on a specified interface.	**ip irdp multicast**
Set the IRDP period for which advertisements are valid.	**ip irdp holdtime** *seconds*
Set the IRDP maximum interval between advertisements.	**ip irdp maxadvertinterval** *seconds*
Set the IRDP minimum interval between advertisements.	**ip irdp minadvertinterval** *seconds*
Set a communication server's IRDP preference level.	**ip irdp preference** *number*
Specify an IRDP address and preference to proxy-advertise.	**ip irdp address** *address* [*number*]

A communication server can proxy-advertise other machines that use IRDP; however, this is not recommended because it is possible to advertise nonexistent machines or machines that are down.

Configure IP Multicast Routing

Traditional IP communication allows a host to send packets to a single host (unicast transmission) or to all hosts (broadcast transmission). IP multicast provides a third scheme, allowing a host to send packets to a subset of all hosts (group transmission). These hosts are known as group members.

Packets delivered to group members are identified by a single multicast group address. Multicast packets are delivered to a group using best-effort reliability just like IP unicast packets.

The multicast environment consists of senders and receivers. Any host, whether it is a member of a group or not, can send to a group. However, only the members of a group receive the message.

A multicast address is chosen for the receivers in a multicast group. Senders use that address as the destination address of a datagram to reach all members of the group.

Membership in a multicast group is dynamic: hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.

How active a multicast group is and what members it has can vary from group to group and from time to time. A multicast group can be active for a very long while, or it may be very short-lived. Membership in a group can change constantly. A group that has members may have no activity.

Routers executing a multicast routing protocol, such as PIM, maintain forwarding tables to forward multicast datagrams. Routers use the Internet Group Management Protocol (IGMP) to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending IGMP report messages.

Cisco's Implementation of IP Multicast Routing

The IOS software supports two protocols to implement IP multicast routing:

- Internet Group Management Protocol (IGMP). IGMP is used between hosts on a LAN and the router(s) on that LAN to track which multicast groups the hosts are members of.
- Protocol-Independent Multicast (PIM). PIM is used between routers so they can track which multicast packets to forward to each other and to their directly connected LANs.

Internet Group Management Protocol (IGMP)

IGMP is used by IP hosts to report their group membership to directly connected multicast routers. IGMP is an integral part of IP. IGMP is defined in RFC 1112, *Host Extensions for IP Multicasting*.

IGMP uses group addresses, which are class D IP addresses. The high-order four bits of a Class D address are 1110. This means that host group addresses can be in the range 224.0.0.0 to 239.255.255.255. The address 224.0.0.0 is guaranteed not to be assigned to any group. The address 224.0.0.1 is assigned to all systems on a subnet; this address is used to address all multicast hosts on the directly connected network. The address 224.0.0.2 is assigned to all routers on a subnet.

Protocol-Independent Multicast (PIM) Protocol

The PIM protocol maintains the current IP multicast service mode of receiver-initiated membership. It is not dependent on a specific unicast routing protocol.

PIM is defined in the following IETF Internet drafts: *Protocol Independent Multicast (PIM): Motivation and Architecture*; *Protocol Independent Multicast (PIM), Dense Mode Protocol Specification*; *Protocol Independent Multicast (PIM), Sparse Mode Protocol Specification*; *IGMP Router Extensions for Routing to Dense Multicast Groups*; and *IGMP Router Extensions for Routing to Sparse Multicast Groups*.

PIM can operate in two modes, dense mode and sparse mode.

In dense mode, a router assumes that all other routers do want to forward multicast packets for a group. If a router receives a multicast packet and has no directly connected members or PIM neighbors present, a prune message is sent back to the source. Subsequent multicast packets are not flooded to this router on this pruned branch. PIM builds source-based.

In sparse mode, a router assumes that other routers do not want to forward multicast packets for a group unless there is an explicit request for the traffic. When hosts join a multicast group, the directly connected routers send PIM Join messages to the rendezvous point (RP). The RP keeps track of multicast groups. Hosts that send multicast packets are registered with the RP by that host's first-hop router. The RP then sends joins toward the source. At this point, packets are forwarded on a shared distribution tree. If the multicast traffic from a specific source is sufficient, the receiver's first-hop router may send joins toward the source to build a source-based distribution tree.

IP Multicast Routing Configuration Task List

To configure IP multicast routing, perform the required tasks described in the following sections:

- Enable IP Multicast Routing on the Communication Server
- Enable PIM on an Interface

You can also perform the optional tasks described in the following sections:

- Configure a Communication Server to Be a Member of a Group
- Configure the Host-Query Message Interval
- Control Access to IP Multicast Groups
- Modify PIM Message Timers
- Configure the TTL Threshold
- Configure DVMRP Interoperability
- Configure a DVMRP Tunnel
- Monitor and Manage IP Multicast Routing

Configuration examples are provided at the end of this chapter.

Enable IP Multicast Routing on the Communication Server

Enabling IP multicast routing on the communication server allows the communication server to forward multicast packets.

To enable IP multicast routing on the communication server, perform the following task in global configuration mode:

Task	Command
Enable IP multicast routing on the communication server.	**ip multicast-routing**

Enable PIM on an Interface

Enabling PIM on an interface also enables IGMP operation on that interface. An interface can be configured to be in dense mode or sparse mode. The mode describes how the router populates its multicast routing table and how the router forwards multicast packets it receives from its directly connected LANs. In populating the multicast routing table, dense-mode interfaces are always added to the table. Sparse-mode interfaces are added to the table only when periodic join messages are received from downstream routers or there is a directly connected member on the interface. When forwarding from a LAN, sparse-mode operation occurs if there is an RP known for the group. If so, the packets are encapsulated and sent toward the RP. When no RP is known, the packet is flooded in a dense-mode fashion. If the multicast traffic from a specific source is sufficient, the receiver's first-hop router may send joins toward the source to build a source-based distribution tree.

There is no default mode setting. By default, multicast routing is disabled on an interface.

To configure PIM on an interface to be in dense mode, perform the following task in interface configuration mode:

Task	Command
Enable dense-mode PIM on the interface.	**ip pim dense-mode**

For an example of how to configure a PIM interface in dense mode, see the section "Configure a Communication Server to Operate in Dense Mode Example" later in this chapter.

To configure PIM on an interface to be in sparse mode, perform the following task in interface configuration mode:

Task	Command
Enable sparse-mode PIM on the interface.	**ip pim sparse-mode**

If you configure the router to operate in sparse mode, you must also choose one or more routers to be RPs. You do not have to configure the routers to be RPs; they learn this themselves. RPs are used by senders to a multicast group to announce their existence and by receivers of multicast packets to learn about new senders. A router can be configured so that packets for a single multicast group can use one or more RPs.

You must configure the IP address of RPs in leaf routers only. Leaf routers are those routers that are directly connected either to a multicast group member or to sender of multicast messages.

The RP address is used by first-hop routers to send PIM register messages on behalf of a host sending a packet to the group. The RP address is also used by last-hop routers to send PIM join/prune messages to the RP to inform it about group membership. The RP does not need to know it is an RP. You need to configure the RP address only on first-hop and last-hop routers (leaf routers).

A PIM communication server can be an RP for more than one group. A group can have more than one RP. The conditions specified by the access-list determine for which groups the communication server is an RP.

To configure the address of the RP, perform the following task in global configuration mode:

Task	Command
Configure the address of a PIM RP.	**ip pim rp-address** *ip-address* [*access-list-number*]

For an example of how to configure a PIM interface in sparse mode, see the section "Configure a Communication Server to Operating in Spare Mode Example" later in this chapter.

Configure a Communication Server to Be a Member of a Group

Cisco communication servers can be configured to members of a multicast group. This is useful for determining multicast reachability in a network. If a communication server is configured to be a group member and supports the protocol that is being transmitted to the group, it can respond. An example is **ping**. A communication server will respond to ICMP echo request packets addressed to a group it is a member of. Another example is the multicast traceroute tools provided in the IOS software.

To have a communication server join a multicast group, perform the following task in interface configuration mode:

Task	Command
Join a multicast group.	**ip igmp join-group** *group-address*

Configure the Host-Query Message Interval

Multicast communication servers send IGMP host-query messages to discover which multicast groups are present on attached local networks. These messages are sent to the all-systems group address of 224.0.0.1 with a TTL of 1.

Multicast communication servers send host-query messages periodically to refresh their knowledge of memberships present on their networks. If, after some number of queries, the communication server discovers that no local hosts are members of a multicast group, the communication server stops forwarding onto the local network multicast packets from remote origins for that group and sends a prune message upstream to the source.

Multicast communication servers elect a PIM designated router for the LAN (subnet). This is the router with the highest IP address. The designated router is responsible for sending IGMP host-query messages to all hosts on the LAN. In sparse mode, the designated router also sends PIM register and PIM join messages towards the RP router.

By default, the designated router sends IGMP host-query messages once a minute in order to keep the IGMP overhead on hosts and networks very low. To modify this interval, perform the following task in interface configuration mode:

Task	Command
Configure the frequency at which the designated router sends IGMP host-query messages.	**ip igmp query-interval** *seconds*

Control Access to IP Multicast Groups

Multicast communication servers send IGMP host-query messages to determine which multicast groups have members of the communication server's attached local networks. The communication servers then forward to these group members all packets addressed to the multicast group. You can place a filter on each interface that restricts the multicast groups that hosts on the subnet serviced by the interface can join.

To filter multicast groups allowed on an interface, perform the following task in interface configuration mode:

Task	Command
Control the multicast groups that hosts on the subnet serviced on an interface can join.	**ip igmp access-group** *access-list-number*

Modify PIM Message Timers

Communication servers operating in sparse-mode PIM send periodic PIM join and prune messages to maintain multicast group membership. By default, communication servers send these messages every 60 seconds (1 minute). A communication server is pruned from a multicast group if a join message from the communication server is not detected at least three times during the periodic interval, or 3 minutes.

All communication servers and routers in the internetwork must be set to use the same timer values.

To modify the frequency at which a sparse-mode PIM communication server sends periodic sparse-mode join and prune PIM messages, perform the following task in global configuration mode:

Task	Command
Modify the frequency at which a sparse-mode PIM communication server sends join and prune messages.	**ip pim message-interval** *seconds*

By default, multicast communication servers send PIM query messages every 30 seconds. To modify this interval, perform the following task in interface configuration mode:

Task	Command
Configure the frequency at which multicast communication servers send PIM communication server-query messages.	**ip pim query-interval** *seconds*

Configure the TTL Threshold

The time-to-live (TTL) value controls whether packets are forwarded out of an interface. You specify the TTL value in hops. Any multicast packet with a TTL less than the interface TTL threshold is not forwarded on the interface. The default value is 0, which means that all multicast packets are forwarded on the interface. To change the default TTL threshold value, perform the following task in interface configuration mode:

Task	Command
Configure the TTL threshold of packets being forwarded out an interface.	**ip multicast-threshold** *ttl*

Configure DVMRP Interoperability

Multicast communication servers using PIM can interoperate with multicast communication servers using the Distance Vector Multiprotocol Routing Protocol (DVMRP).

PIM communication servers dynamically discover DVMRP multicast communication servers on attached networks. Once a DVMRP neighbor has been discovered, the communication server periodically transmits DVMRP report messages advertising the unicast sources reachable in the PIM domain. By default, directly connected subnets and networks are advertised. The communication server forwards multicast packets that have been forwarded by DVMRP communication servers and in turn forwards multicast packets to DVMRP communication servers.

You can configure what sources are advertised and what metrics are used by using the **ip dvmrp metric** command. You can also direct all sources learned via a particular unicast routing process to be advertised into DVMRP.

It is necessary to use mrouted version 2.2 (which implements a nonpruning version of DVMRP) or version 3.2 (which implements a pruning version of DVMRP) when Cisco routers are directly connected to DVMRP routers or interoperate with DVMRP routers over an MBONE tunnel. DVMRP advertisements produced by Cisco routers can cause older versions of mrouted to corrupt their routing tables and those of their neighbors.

Our communication servers answer mrinfo requests sent by mrouted systems. The communication server returns information about neighbors on DVMRP tunnels. This information includes the metric, which is always set to 1, plus the configured TTL threshold and the status of the tunnel.

To configure the metric associated with a set of destinations for DVMRP reports, perform the following task in interface configuration mode:

Task	Command
Configure the metric associated with a set of destinations for DVMRP reports.	**ip dvmrp metric** *metric* [*access-list-number*] [*protocol process-id*]

For an example of how to configure a PIM router to interoperate with a DVMRP router, see the section “Configure DVMRP Interoperability Examples” later in this chapter.

Configure a DVMRP Tunnel

Our communication servers can support DVMRP tunnels to the MBONE. (The MBONE is the multicast backbone of the Internet.) You can configure a DVMRP tunnel on a communication server if the other end is running DVMRP. The communication server then sends and receives multicast packets over the tunnel. This allows a PIM domain to connect to the DVMRP router in the case where all routers on the path do not support multicast routing.

When our communication server runs DVMRP over a tunnel, it advertises sources in DVMRP report messages much as it does on real networks. In addition, DVMRP report messages received are cached on the communication server and are used as part of its Reverse Path Forwarding (RPF) calculation. This allows multicast packets received over the tunnel to be forwarded by the communication server.

When you configure a DVMRP tunnel, you should assign a tunnel an address for two reasons:

- To enable the sending of IP packets over the tunnel
- To indicate whether the Cisco IOS software should perform DVMRP summarization

You can assign an IP address either by using the **ip address** interface configuration command or by using the **ip unnumbered** interface configuration command to configure the tunnel to be unnumbered. Either of these two methods allows IP multicast packets to flow over the tunnel. The Cisco IOS software will not advertise subnets over the tunnel if the tunnel has a different network number from the subnet. In this case, the software advertises only the network number over the tunnel.

To configure a DVMRP tunnel, perform the following tasks starting in global configuration mode:

Task		Command
Step 1	Specify a tunnel interface in global configuration mode. This places the communication server into interface configuration mode.	**interface tunnel** *number*

Task		Command
Step 2	Set the tunnel interface's source address. This is the IP address of the interface on the communication server.	**tunnel source** *ip-address*
Step 3	Set the tunnel interface's destination address. This is the IP address of the mrouted multitask router.	**tunnel destination** *ip-address*
Step 4	Configure a DVRMP tunnel.	**tunnel mode dvmrp**
Step 5	Assign an IP address to the interface. or Configure the interface as unnumbered.	**ip address** *address mask* **ip unnumbered**
Step 6	Configure PIM on the interface.	**ip pim** [**dense-mode** | **sparse-mode**]
Step 7	Configure an acceptance filter for incoming DMVRP reports.	**ip dvmrp accept-filter** *access-list-number administrative-distance*

Monitor and Manage IP Multicast Routing

To monitor IP multicast routing information and to clear IP multicast routing caches, perform one or more of the following tasks at the EXEC prompt:

Task	Command
Delete entries from the IGMP cache.	**clear ip igmp group** [*group-name* | *group-address* | *type number*]
Delete entries from the IP multicast routing table.	**clear ip mroute** ***** | {*group-name* [*source-address*] | *group-address* [*source-address*]}
Trace a branch of a multicast tree for a specific group.	**mbranch** *group-address branch-address* [*ttl*]
Trace a branch of a multicast tree for a group in the reverse direction.	**mrbranch** *group-address branch-address* [*ttl*]
Display the entries in the DVMRP routing table.	**show ip dvmrp route** [*ip-address*]
Display the multicast groups that are directly connected to the router and that were learned via IGMP.	**show ip igmp groups** [*group-name* | *group-address* | *type number*]
Display multicast-related information about an interface.	**show ip igmp interface** [*type number*]
Display the contents of the IP multicast routing table.	**show ip mroute** [*group-name* | *group-address*] [**summary**] [**count**] **show ip mroute** [*group-name* [*source-address*] | *group-address* [*source-address*]]
Display information about interfaces configured for PIM.	**show ip pim interface** [*type number*]
List the PIM neighbors discovered by the router.	**show ip pim neighbor** [*type number*]
Display the RP routers associated with a sparse-mode multicast group.	**show ip pim rp** [*group-name* | *group-address*]

Configure Routing Protocol-Independent Features

Previous sections addressed configurations of specific routing protocols. Complete the protocol-independent tasks described in the following sections as needed:

- Use Variable-Length Subnet Masks
- Configure Static Routes
- Specify Default Routes
- Redistribute Routing Information
- Filter Routing Information
- Adjust Timers
- Adjust the Interval between Hello Packets and the Hold Time (EIGRP)
- Enable or Disable Split Horizon

Use Variable-Length Subnet Masks

OSPF and static routes support variable-length subnet masks (VLSMs). With VLSMs, you can use different masks for the same network number on different interfaces, which allows you to conserve IP addresses and more efficiently use available address space. However, using VLSMs also presents address assignment challenges for the network administrator and ongoing administrative challenges.

Refer to RFC 1219 for detailed information about VLSMs and how to correctly assign addresses.

Note Consider your decision to use VLSMs carefully. It is easy to make mistakes in address assignments and it is generally more difficult to monitor your network using VLSMs.

The best way to implement VLSMs is to keep your existing numbering plan in place and gradually migrate some networks to VLSMs to recover address space. See the "IP Routing Protocol Configuration Examples" section at the end of this chapter for an example of using VLSMs.

Configure Static Routes

Static routes are user-defined routes that cause packets moving between a source and a destination to take a specified path. Static routes can be important if the communication server cannot build a route to a particular destination. They are also useful for specifying a gateway of last resort to which all unroutable packets will be sent.

To configure static routes, perform the following task in global configuration mode:

Task	Command
Establish a static route.	**ip route** *network* [*mask*] {*address* | *interface*} [*distance*]

See the "IP Routing Protocol Configuration Examples" section at the end of this chapter for an example of configuring static routes.

The communication server remembers static routes until you remove them (using the **no** form of the **ip route** global configuration command). However, you can override static routes with dynamic routing information through prudent assignment of administrative distance values. Each dynamic routing protocol has a default administrative distance, as listed in Table 18-1. If you would like a static route to be overridden by information from a dynamic routing protocol, simply ensure that the administrative distance of the static route is higher than that of the dynamic protocol.

Static routes that point to an interface will be advertised via RIP, IGRP, and other dynamic routing protocols, regardless of whether **redistribute static** commands were specified for those routing protocols. This is because static routes that point to an interface are considered in the routing table to be connected and hence lose their static nature. However, if you define a static route to an interface that is not one of the networks defined in a **network** command, no dynamic routing protocols will advertise the route unless a **redistribute static** command is specified for these protocols.

When an interface goes down, all static routes through that interface are removed from the IP routing table. Also, when the communication server can no longer find a valid next hop for the address specified as the forwarding communication server's address in a static route, the static route is removed from the IP routing table.

Table 18-1 Default Administrative Distances

Route Source	Default Distance
Connected interface	0
Static route	1
External BGP	20
IGRP	100
OSPF	110
RIP	120
EGP	140
Internal BGP	200
Unknown	255

Specify Default Routes

A communication server might not be able to determine the routes to all other networks. To provide complete routing capability, the common practice is to use some communication servers as "smart communication servers" and give the remaining communication servers default routes to the smart communication server. (Smart communication servers have routing table information for the entire internetwork.) These default routes can be passed along dynamically or can be configured into the individual communication servers.

Most dynamic interior gateway routing protocols include a mechanism for causing a smart communication server to generate dynamic default information that is then passed along to other communication servers.

Specify a Default Network

If a communication server has a directly connected interface onto the specified default network, the dynamic routing protocols running on that communication server will generate or source a default route. RIP advertises the pseudonetwork 0.0.0.0. For IGRP, the network itself is advertised and flagged as an exterior route.

A communication server that is generating the default for a network may also need a default of its own. One way of doing this is to specify a static route to the network 0.0.0.0 through the appropriate communication server.

To define a static route to a network as the static default route, perform the following task in global configuration mode:

Task	Command
Specify a default network.	**ip default-network** *network-number*

The Gateway of Last Resort

When default information is being passed along through a dynamic routing protocol, no further configuration is required. The system will periodically scan its routing table to choose the optimal default network as its default route. In the case of RIP, it will be only one choice, network 0.0.0.0. In the case of IGRP, there might be several networks that can be candidates for the system default. The communication server uses both administrative distance and metric information to determine the default route (gateway of last resort). The selected default route appears in the gateway of last resort display of the **show ip route** EXEC command.

If dynamic default information is not being passed to the communication server, candidates for the default route can be specified with the **ip default-network** command. In this usage, **ip default-network** takes a nonconnected network as an argument. If this network appears in the routing table from any source (dynamic or static), it is flagged as a candidate default route and is a possible choice as the default route for the communication server.

If the communication server has no interface on the default network but does have a route to it, it will consider this network as a candidate default path. The route candidates will be examined and the best one will be chosen based on administrative distance and metric. The gateway to the best default path will become the gateway of last resort for the communication server.

Redistribute Routing Information

In addition to running multiple routing protocols simultaneously, the communication server can redistribute information from one routing protocol to another. For example, you can instruct the communication server to readvertise IGRP-derived routes using the RIP protocol, or to readvertise static routes using the IGRP protocol or IP Enhanced IGRP protocol. This capability applies to all of the IP-based routing protocols.

You may also conditionally control the redistribution of routes between routing domains by defining a method known as *route maps* between the two domains.

The following four tables list tasks associated with route redistribution.

To define a route map for redistribution, perform the following task in global configuration mode:

Task	Command
Define any route maps needed to control redistribution.	**route-map** *map-tag* [[**permit** | **deny**] | [*sequence-number*]]

A pair of **match** and **set** commands are required to follow a **route-map** command. To define conditions for redistributing routes from one routing protocol into another, perform at least one of the following tasks in route-map configuration mode:

Task	Command
Match a BGP autonomous system path access list.	**match as-path** *path-list-number*
Match a standard access list.	**match ip address** *access-list-number...access-list-number*
Match the specified metric.	**match metric** *metric-value*
Match a next-hop communication server address passed by one of the access lists specified.	**match ip next-hop** *access-list-number...access-list-number*
Match the specified tag value.	**match tag** *tag-value...tag-value*
Match the specified next hop route out one of the interfaces specified.	**match interface** *type number...type number*
Match the address specified by the specified advertised access lists.	**match ip route-source** *access-list-number...access-list-number*
Match the specified route type.	**match route-type** {**internal** | **external** | **level-1** | **level-2**}

A pair of **match** and **set** commands are required to follow a **route-map** command. To define conditions for redistributing routes from one routing protocol into another, perform at least one of the following tasks in route-map configuration mode:

Task	Command
Assign a value to a local BGP path.	**set local-preference** *value*
Specify the BGP weight for the routing table.	**set weight** *weight*
Set the BGP origin code.	**set origin** {**igp** | **egp** *remote-as* | **incomplete**}
Specify the address of the next hop.	**set next-hop** *next-hop*
Enable automatic computing of tag table.	**set automatic-tag**
For routes that are advertised into the specified area of the routing domain.	**set level** {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**}
Set the metric value to give the redistributed routes.	**set metric** *metric-value*
Set the metric type to give redistributed routes.	**set metric-type** {**internal** | **external** | **type-1** | **type-2**}
Set a tag value to associate with the redistributed routes.	**set tag** *tag-value*

To distribute routes from one routing domain into another and to control route redistribution, perform the following tasks in router configuration mode:

Task	Command
Redistribute routes from one routing protocol to another routing protocol.	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**match internal** | **external** *type-value*] [**tag** *tag-value*] [**route-map** *map-tag*] [**weight** *weight*] [**subnets**]
Cause the current routing protocol to use the same metric value for all redistributed routes.	**default-metric** *number*
Cause the IGRP routing protocol to use the same metric value for all redistributed routes.	**default-metric** *bandwidth delay reliability loading mtu*
Disable the redistribution of default information between IGRP processes. This is enabled by default.	**no default-information allowed** {**in** | **out**}

The metrics of one routing protocol do not necessarily translate into the metrics of another. For example, the RIP metric is a hop count and the IGRP metric is a combination of five quantities. In such situations, an artificial metric is assigned to the redistributed route. Because of this unavoidable tampering with dynamic information, carelessly exchanging routing information between different routing protocols can create routing loops, which can seriously degrade network operation.

See the “IP Routing Protocol Configuration Examples” section at the end of this chapter for examples of configuring redistribution and route maps.

Supported Metric Translations

This section describes supported automatic metric translations between the routing protocols. The following descriptions assume that you have not defined a default redistribution metric that replaces metric conversions.

- RIP can automatically redistribute static routes. It assigns static routes a metric of 1 (directly connected).
- EGP can automatically redistribute static routes and all dynamically derived routes. EGP assigns the metric 3 to all static and derived routes.
- BGP does not normally send metrics in its routing updates.
- IGRP can automatically redistribute static routes and information from other IGRP-routed autonomous systems. IGRP assigns static routes a metric that identifies them as directly connected. IGRP does not change the metrics of routes derived from IGRP updates from other autonomous systems.
- Note that any protocol can redistribute other routing protocols if a default metric is in effect.

Filter Routing Information

You can filter routing protocol information by performing the following tasks:

- Prevent the transmission of routing updates on a particular communication server interface. Do this to prevent other systems on an interface from learning about routes dynamically.

- Prevent networks from being advertised in routing updates. do this to prevent other communication servers from learning a particular communication server's interpretation of one or more routes.
- Prevent networks listed in updates from being accepted and acted upon by a routing process. Do this to keep a communication server from using certain routes.
- Filter on the source of routing information. Do this to prioritize routing information from different sources, because some pieces of routing information may be more accurate than others.
- Apply an offset to routing metrics. Do this to provide a local mechanism for increasing the value of routing metrics.

Note When routes are redistributed between OSPF processes, no OSPF metrics are preserved.

The following sections describe these tasks.

Prevent Routing Updates through an Interface

To prevent a routing process from learning about routes dynamically, you can keep routing update messages from being sent through a communication server interface. This feature applies to all IP-based routing protocols except BGP and EGP.

OSPF behavior is somewhat different. In OSPF, the interface address you specify as passive appears as a stub network in the OSPF domain. OSPF routing information is neither sent nor received through the specified communication server interface.

To prevent routing updates through a specified interface, perform the following task in router configuration mode:

Task	Command
Suppress the transmission of routing updates through the specified communication server interface.	**passive-interface** *type number*

See the "IP Routing Protocol Configuration Examples" section at the end of this chapter for examples of configuring passive interfaces.

Control the Advertising of Routes in Routing Updates

To prevent other communication servers from learning one or more routes, you can prevent the advertising of routes in routing updates. You cannot specify an interface name in OSPF. When used for OSPF, this feature applies only to external routes.

To prevent routes from being advertised in routing updates, perform the following task in router configuration mode:

Task	Command
Permit or deny advertising of routes in routing updates depending upon the action listed in the access list.	**distribute-list** *access-list-number* **out** [*type number* | *routing-process*]

Control the Processing of Routing Updates

You might want to avoid processing certain routes listed in incoming updates. This feature does not apply to OSPF.

Perform this task in router configuration mode:

Task	Command
Suppress routes listed in updates from being processed.	**distribute-list** *access-list-number* **in** [*type number*]

Apply Offsets to Routing Metrics

To provide a local mechanism for increasing the value of routing metrics, you can apply an offset to routing metrics. This feature applies to IGRP and RIP.

To apply an offset to routing metrics, perform the following task in router configuration mode:

Task	Command
Apply an offset to routing metrics.	**offset-list** {**in** | **out**} *offset* [*access-list-number*]

Filter Sources of Routing Information

An *administrative distance* is a rating of the trustworthiness of a routing information source, such as an individual communication server or a group of communication servers. In a large network, some routing protocols and some communication servers can be more reliable than others as sources of routing information. Also, when multiple routing processes are running in the same communication server for IP, it is possible for the same route to be advertised by more than one routing process. By specifying administrative distance values, you enable the communication server to discriminate intelligently between sources of routing information. The communication server will always pick the route whose routing protocol has the lowest administrative distance.

There are no general guidelines for assigning administrative distances, because each network has its own requirements. You must determine a reasonable matrix of administrative distances for the network as a whole. Table 18-2 shows the default administrative distance for various sources of routing information.

Table 18-2 Default Administrative Distances

Route Source	Default Distance
Connected interface	0
Static route	1
External BGP	20
IGRP	100
OSPF	110
RIP	120
EGP	140
Internal BGP	200
Unknown	255

For example, consider a communication server using IGRP and RIP. Suppose you trust the IGRP-derived routing information more than the RIP-derived routing information. In this example, because the default IGRP administrative distance is lower than the default RIP administrative distance, the communication server uses the IGRP-derived information and ignores the RIP-derived information. However, if you lose the source of the IGRP-derived information (to a power shutdown in another building, for example), the communication server uses the RIP-derived information until the IGRP-derived information reappears.

Note You can also use administrative distance to rate the routing information from communication servers running the same routing protocol. This application is generally discouraged if you are unfamiliar with this particular use of administrative distance, because it can result in inconsistent routing information, including forwarding loops.

To filter sources of routing information, perform the following task in router configuration mode:

Task	Command
Filter on routing information sources.	**distance** *weight* [*address-mask* [*access-list-number*]] [**ip**]

See the “IP Routing Protocol Configuration Examples” section at the end of this chapter for examples of setting administrative distances.

Adjust Timers

Routing protocols use a variety of timers that determine such variables as the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune routing protocol performance to better suit your internetwork needs.

For IGRP and RIP, you can make the following timer adjustments:

- The rate (time in seconds between updates) at which routing updates are sent
- The interval of time (in seconds) after which a route is declared invalid
- The interval (in seconds) during which routing information regarding better paths is suppressed
- The amount of time (in seconds) that must pass before a route is removed from the routing table
- The amount of time for which routing updates will be postponed

EGP and BGP have their own **timers** commands, although some EGP timers might be set with the timers basic command. See the EGP and BGP sections, respectively.

It also is possible to tune the IP routing support in the software to enable faster convergence of the various IP routing algorithms and hence, quicker fallback to redundant communication servers. The total effect is to minimize disruptions to end users of the network in situations where quick recovery is essential.

The following two tables list tasks associated with adjusting routing protocol timers and the keepalive interval.

Perform the following task in router configuration mode:

Task	Command
Adjust routing protocol timers.	**timers basic** *update invalid holddown flush* [*sleeptime*]

Perform the following the following task in interface configuration mode:

Task	Command
Adjust the frequency with which the communication server sends messages to itself (Ethernet and Token Ring) or to the other end (HDLC-serial and PPP-serial links) to ensure that a network interface is alive for a specified interface.	**keepalive** [*seconds*] [1]

[1] This command is documented in the "Interface Commands" chapter of the *Access and Communication Servers Command Reference* publication.

You can also configure the *keepalive* interval, the frequency at which the communication server sends messages to itself (Ethernet and Token Ring) or to the other end (hdlc-serial, ppp-serial) to ensure that a network interface is alive. The interval in some previous software versions was 10 seconds; it is now adjustable in 1-second increments down to one second. An interface is declared down after three update intervals have passed without receiving a keepalive packet.

When adjusting the keepalive timer for a very low bandwidth serial interface, large packets can delay the smaller keepalive packets long enough to cause the line protocol to go down. You might need to experiment to determine the best value.

Adjust the Interval between Hello Packets and the Hold Time (EIGRP)

You can adjust the interval between hello packets and the hold time.

Routers periodically send hello packets to each other to learn dynamically about other communication servers or routers on their directly attached networks. The communication servers use this information to discover who their neighbors are and to learn when their neighbors become unreachable or inoperative. By default, hello packets are sent every 5 seconds.

You can configure the hold time on a specified interface for the IP Enhanced IGRP routing process designated by the autonomous system number. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is three times the hello interval, or 15 seconds.

To change the interval between hello packets, perform the following task in interface configuration mode:

Task	Command
Configure the hello interval for an IP Enhanced IGRP routing process.	**ip hello-interval eigrp** *autonomous-system-number seconds*

On very congested and large networks, 15 seconds may not be sufficient time for all communication servers to receive hello packets from their neighbors. In this case, you may want to increase the hold time.

To change the hold time, perform the following task in interface configuration mode:

Task	Command
Configure the hold time for an IP Enhanced IGRP routing process.	**ip hold-time eigrp** *autonomous-system-number seconds*

Note Do not adjust the hold time without advising technical support.

Enable or Disable Split Horizon

Normally, communication servers that are connected to broadcast-type IP networks and that use distance-vector routing protocols employ the *split horizon* mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a communication server out any interface from which that information originated. This behavior usually optimizes communications among multiple communication servers, particularly when links are broken. However, with nonbroadcast networks, such as Frame Relay and SMDS, situations can arise for which this behavior is less than ideal. For these situations, you might want to disable split horizon. This applies to IGRP and RIP.

If an interface is configured with secondary IP addresses, split horizon rules can affect whether or not routing updates are sourced by these secondary addresses. If the primary and secondary IP address network numbers belong to the same network class, routing updates source by the secondary address are suppressed unless split horizon is disabled. If the primary and secondary addresses do not belong to the same network class, routing updates sourced by the secondary address are not suppressed.

To enable or disable split horizon, perform the following tasks in interface configuration mode:

Task	Command
Enable split horizon.	**ip split-horizon**
Disable split horizon.	**no ip split-horizon**

Split horizon for Frame Relay and SMDS encapsulation is disabled by default. Split horizon is not disabled by default for interfaces using any of the X.25 encapsulations. For all other encapsulations, split horizon is enabled by default.

See the “IP Routing Protocol Configuration Examples” section at the end of this chapter for an example of using split horizon.

Note In general, changing the state of the default is not recommended unless you are certain that your application requires making a change in order to advertise routes properly. Remember: If split horizon is disabled on a serial interface (and that interface is attached to a packet-switched network), you *must* disable split horizon for all communication servers in any relevant multicast groups on that network.

Monitor and Maintain the IP Network

You can remove all contents of a particular cache, table, or database. You also can display specific communication server statistics. The following sections describe each of these tasks.

Clear Caches, Tables, and Databases

You can remove all contents of a particular cache, table, or database. Clearing a cache, table, or database can become necessary when the contents of the particular structure have become or are suspected to be invalid.

The following table lists the tasks associated with clearing caches, tables, and databases for IP routing protocols. Perform these tasks in EXEC mode:

Task	Command
Clear the IP ARP cache and the fast-switching cache.	**clear arp-cache**
Reset a particular BGP connection.	**clear ip bgp** *address*
Reset all BGP connections.	**clear ip bgp ***
Clear one or more routes from the IP routing table.	**clear ip route** {*network* [*mask*] | *****}

Display System and Network Statistics

You can display specific communication server statistics such as the contents of IP routing tables, caches, and databases. Information provided can be used to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path your communication server's packets are taking through the network.

To display various communication server statistics, perform the following tasks in EXEC mode:

Task	Command
Display all BGP routes that contain subnet and supernet network masks.	**show ip bgp cidr-only**
Display supernets.	**show ip route supernets-only**
Display routes that are matched by the specified AS path access list.	**show ip bgp filter-list** *access-list-number*
Display the routes that match the specified regular expression entered on the command line.	**show ip bgp regexp** *regular-expression*
Display the contents of the BGP routing table.	**show ip bgp** [*network*] [*network-mask*] [**subnets**]
Display detailed information on the TCP and BGP connections to individual neighbors.	**show ip bgp neighbors** [*address*]
Display routes learned from a particular BGP neighbor.	**show ip bgp neighbors** *address* [**routes** | **paths**]
Display all BGP paths in the database.	**show ip bgp paths**
Display the status of all BGP connections.	**show ip bgp summary**
Display statistics on EGP connections and neighbors.	**show ip egp**
Display the IP Enhanced IGRP topology table for a given process.	**show ip eigrp topology** [*autonomous-system-number* | [[*ip-address*] *mask*]]
Display the number of packets sent and received for all or a specified IP Enhanced IGRP process.	**show ip eigrp traffic** [*autonomous-system-number*]
Display IRDP values.	**show ip irdp**
Display general information about OSPF routing processes in a particular communication server.	**show ip ospf** [*process-id*]

Task	Command
Display lists of information related to the OSPF database for a specific communication server.	**show ip ospf** [*process-id area-id*] **database** **show ip ospf** [*process-id area-id*] **database** [**router**] [*link-state-id*] **show ip ospf** [*process-id area-id*] **database** [**network**] [*link-state-id*] **show ip ospf** [*process-id area-id*] **database** [**summary**] [*link-state-id*] **show ip ospf** [*process-id area-id*] **database** [**asb-summary**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**external**] [*link-state-id*]
Display OSPF-related interface information.	**show ip ospf interface** [*type number*]
Display OSPF-neighbor information on a per-interface basis.	**show ip ospf neighbor** [*type number*] [*neighbor-id*] **detail**
Display OSPF-related virtual links information.	**show ip ospf virtual-links**
Display the parameters and current state of the active routing protocol process.	**show ip protocols**
Display the current state of the routing table.	**show ip route** [*address* [*mask*]] | [*protocol*]
Display the current state of the routing table in summary form.	**show ip route summary**
Display all route maps configured or only the one specified.	**show route-map** [*map-name*]
Display the internal OSPF routing table entries to Area Border Router (ABR) and Autonomous System Boundary Router (ASBR).	**show ip ospf border-routers**

IP Routing Protocol Configuration Examples

The following sections provide IP routing protocol configuration examples:

- Variable-Length Subnet Masks Example
- Overriding Static Routes with Dynamic Protocols Example
- Static Routing Redistribution Examples
- IGRP Redistribution Examples
- IP Enhanced IGRP Redistribution Examples
- RIP and IGRP Redistribution Example
- RIP and IP Enhanced IGRP Redistribution Examples
- OSPF Routing and Route Redistribution Examples
- BGP Route Advertisement and Redistribution Examples
- Default Metric Values Redistribution Examples
- Route-Map Examples
- Route Summarization Example

- IGRP Feasible Successor Relationship Example
- BGP Synchronization Example
- BGP Basic Neighbor Specification Examples
- BGP Aggregate Route Examples
- Third-Party EGP Support Example
- Backup EGP Communication Server Example
- EGP Core Gateway Example
- Autonomous System within EGP Example
- Passive Interface Examples
- Administrative Distance Examples
- Split Horizon Examples
- IP Multicast Routing Configuration Examples

Variable-Length Subnet Masks Example

OSPF and static routes support variable-length subnet masks (VLSMs). With VLSMs, you can use different masks for the same network number on different interfaces, which allows you to conserve IP addresses and more efficiently use available address space.

In the following example, a 14-bit subnet mask is used, leaving two bits of address space reserved for serial line host addresses. There is sufficient host address space for two host endpoints on a point-to-point serial link.

```
interface ethernet 0
ip address 131.107.1.1 255.255.255.0
! 8 bits of host address space reserved for ethernets

interface serial 0
ip address 131.107.254.1 255.255.255.252
! 2 bits of address space reserved for serial lines

! System is configured for OSPF and assigned AS 107
router ospf 107
! Specifies network directly connected to the system
network 131.107.0.0 0.0.255.255 area 0.0.0.0
```

Overriding Static Routes with Dynamic Protocols Example

In the following example, packets for network 10.0.0.0 from Router B, where the static route is installed, will be routed through 131.108.3.4 if a route with an administrative distance less than 110 is not available. Figure 18-3 illustrates this point. The route learned by a protocol with an administrative distance of less than 110 might cause Router B to send traffic destined for network 10.0.0.0 via the alternate path—through Router D.

```
ip route 10.0.0.0 255.0.0.0 131.108.3.4 110
```

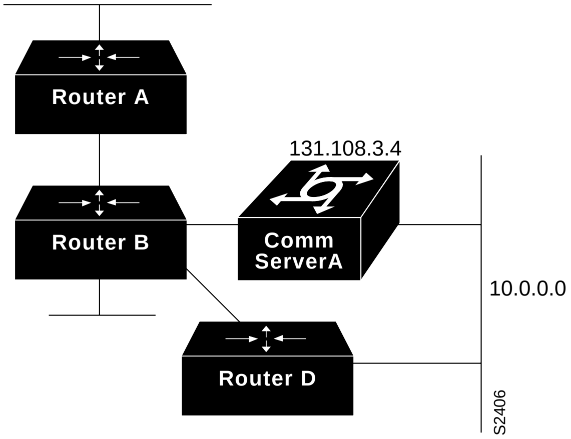

Figure 18-3 Overriding Static Routes

Static Routing Redistribution Examples

In the example that follows, three static routes are specified, two of which are to be advertised. Do this by specifying the **redistribute static** router configuration command, then specifying an access list that allows only those two networks to be passed to the IGRP process. Any redistributed static routes should be sourced by a single communication server to minimize the likelihood of creating a routing loop.

```
ip route 192.1.2.0 255.255.255.0 192.31.7.65
ip route 193.62.5.0 255.255.255.0 192.31.7.65
ip route 131.108.0.0 255.255.255.0 192.31.7.65
access-list 3 permit 192.1.2.0
access-list 3 permit 193.62.5.0
!
router igrp 109
network 192.31.7.0
default-metric 10000 100 255 1 1500
redistribute static
distribute-list 3 out static
```

The same basic example can be adapted for IP Enhanced IGRP operation:

```
ip route 192.1.2.0 255.255.255.0 192.31.7.65
ip route 193.62.5.0 255.255.255.0 192.31.7.65
ip route 131.108.0.0 255.255.0.0 192.31.7.65
access-list 3 permit 192.1.2.0
access-list 3 permit 193.62.5.0
!
router eigrp 109
network 192.31.7.0
redistribute static route-map static-to-eigrp

route-map static-to-eigrp
match ip address 3
set metric 10000 100 255 1 1500
```

IGRP Redistribution Examples

Each IGRP routing process can provide routing information to only one autonomous system; the communication server must run a separate IGRP process and maintain a separate routing database for each autonomous system it services. However, you can transfer routing information between these routing databases.

Suppose the communication server has one IGRP routing process for network 15.0.0.0 in autonomous system 71 and another for network 192.31.7.0 in autonomous system 109, as the following commands specify:

```
router igrp 71
network 15.0.0.0
router igrp 109
network 192.31.7.0
```

To transfer a route to 192.31.7.0 into autonomous system 71 (without passing any other information about autonomous system 109), use the command in the following example:

```
router igrp 71
redistribute igrp 109
distribute-list 3 out igrp 109
access-list 3 permit 192.31.7.0
```

IP Enhanced IGRP Redistribution Examples

Each IP Enhanced IGRP routing process can provide routing information to only one autonomous system; the communication server must run a separate IP Enhanced IGRP process and maintain a separate routing database for each autonomous system it services. However, you can transfer routing information between these routing databases.

Suppose the communication server has one IP Enhanced IGRP routing process for network 15.0.0.0 in autonomous system 71 and another for network 192.31.7.0 in autonomous system 109, as the following commands specify:

```
router eigrp 71
network 15.0.0.0
router eigrp 109
network 192.31.7.0
```

To transfer a route from 192.31.7.0 into autonomous system 71 (without passing any other information about autonomous system 109), use the command in the following example:

```
router eigrp 71
redistribute eigrp 109 route-map 109-to-71
route-map 109-to-71 permit
match ip address 3
set metric 10000 100 1 255 1500
access-list 3 permit 192.31.7.0
```

The following example is an alternative way to transfer a route to 192.31.7.0 into autonomous system 71. Unlike the previous configuration, this one does not allow you to arbitrarily set the metric.

```
router eigrp 71
redistribute eigrp 109
distribute-list 3 out eigrp 109
access-list 3 permit 192.31.7.0
```

RIP and IGRP Redistribution Example

Consider a WAN at a university that uses RIP as an interior routing protocol. Assume that the university wants to connect its wide area network to a regional network, 128.1.0.0, which uses IGRP as the routing protocol. The goal in this case is to advertise the networks in the university network to the communication servers on the regional network. The commands for the interconnecting communication server are listed in the example that follows:

```
router igrp 109
network 128.1.0.0
redistribute rip
default-metric 10000 100 255 1 1500
distribute-list 10 out rip
```

In this example, the **router** global configuration command starts an IGRP routing process. The **network** router configuration command specifies that network 128.1.0.0 (the regional network) is to receive IGRP routing information. The **redistribute** router configuration command specifies that RIP-derived routing information be advertised in the routing updates. The **default-metric** router configuration command assigns an IGRP metric to all RIP-derived routes.

The **distribute-list** router configuration command instructs the communication server to use access list 10 (not defined in this example) to limit the entries in each outgoing update. The access list prevents unauthorized advertising of university routes to the regional network.

RIP and IP Enhanced IGRP Redistribution Examples

This section provides a simple example and a complex example.

Example 1: Simple Redistribution

Consider a wide-area network at a university that uses RIP as an interior routing protocol. Assume that the university wants to connect its wide-area network to a regional network, 128.1.0.0, which uses IP Enhanced IGRP as the routing protocol. The goal is to advertise the networks in the university network to the routers on the regional network. The following example shows the commands for the interconnecting router:

```
router eigrp 109
network 128.1.0.0
redistribute rip
default-metric 10000 100 255 1 1500
distribute-list 10 out rip
```

The **router** global configuration command starts an IP Enhanced IGRP routing process. The **network** router configuration command specifies that network 128.1.0.0 (the regional network) is to send and receive IP Enhanced IGRP routing information. The **redistribute** router configuration command specifies that RIP-derived routing information be advertised in the routing updates. The **default-metric** router configuration command assigns an IP Enhanced IGRP metric to all RIP-derived routes.

The **distribute-list** router configuration command instructs the router to use access list 10 (not defined in this example) to limit the entries in each outgoing update. The access list prevents unauthorized advertising of university routes to the regional network.

Example 2: Complex Redistribution

The most complex redistribution case is one in which *mutual* redistribution is required between an IGP (in this case, IP Enhanced IGRP) and BGP.

Suppose that EGP is running on a router somewhere else in autonomous system 1, and that the EGP routes are injected into IP Enhanced IGRP routing process 1. You must use filters to ensure that the proper routes are advertised. The example configuration for router R1 illustrates use of access filters and a distribution list to filter routes advertised to BGP neighbors. This example also illustrates configuration commands for redistribution between BGP and IP Enhanced IGRP.

```
! Configuration for router R1:
router bgp 1
network 131.108.0.0
neighbor 192.5.10.1 remote-as 2
neighbor 192.5.10.15 remote-as 1
! 131.108.2.1 an internal peer
neighbor 192.5.10.24 remote-as 3
! 131.108.3.1 is an external peer
redistribute eigrp 1
distribute-list 1 out eigrp 1
!
! All networks that should be advertised from R1 are controlled with access lists:
!
access-list 1 permit 131.108.0.0
access-list 1 permit 150.136.0.0
access-list 1 permit 128.125.0.0
!
router eigrp 1
network 131.108.0.0
network 192.5.10.0
redistribute bgp 1
```

OSPF Routing and Route Redistribution Examples

OSPF typically requires coordination among many internal communication servers, communication servers acting as area border routers, and communication servers acting as autonomous system boundary routers. At a minimum, OSPF-based communication servers can be configured with all default parameter values, with no authentication, and with interfaces assigned to areas.

Three examples follow:

- The first is a simple configuration illustrating basic OSPF commands.
- The second example illustrates a configuration for communication servers acting as internal, area border, and autonomous system boundary routers within a single, arbitrarily assigned, OSPF autonomous system.
- The third example illustrates a more complex configuration and the application of various tools available for controlling OSPF-based routing environments.

Example 1: Basic OSPF Configuration

The following example illustrates a simple OSPF configuration that enables OSPF routing process 9000, attaches Ethernet 0 to area 0.0.0.0, and redistributes RIP into OSPF, and OSPF into RIP:

```
interface ethernet 0
ip address 130.93.1.1 255.255.255.0
ip ospf cost 1
!
interface ethernet 1
ip address 130.94.1.1 255.255.255.0
!
router ospf 9000
network 130.93.0.0 0.0.255.255 area 0.0.0.0
redistribute rip metric 1 subnets
!
router rip
network 130.94.0.0
redistribute ospf 9000
default-metric 1
```

Example 2: Another Basic OSPF Configuration

The following example illustrates the assignment of four area IDs to four IP address ranges. In the example, OSPF routing process 109 is initialized, and four OSPF areas are defined: 10.9.50.0, 2, 3, and 0. Areas 10.9.50.0, 2, and 3 mask specific address ranges, while Area 0 enables OSPF for *all other* networks.

```
router ospf 109
network 131.108.20.0  0.0.0.255 area 10.9.50.0
network 131.108.0.0  0.0.255.255 area 2
network 131.109.10.0  0.0.0.255 area 3
network 0.0.0.0  255.255.255.255 area 0
!
! Interface Ethernet0 is in area 10.9.50.0:
interface ethernet 0
ip address 131.108.20.5 255.255.255.0
!
! Interface Ethernet1 is in area 2:
interface ethernet 1
ip address 131.108.1.5 255.255.255.0
!
! Interface Ethernet2 is in area 2:
interface ethernet 2
ip address 131.108.2.5 255.255.255.0
!
! Interface Ethernet3 is in area 3:
interface ethernet 3
ip address 131.109.10.5 255.255.255.0
!
! Interface Ethernet4 is in area 0:
interface ethernet 4
ip address 131.109.1.1 255.255.255.0
!
! Interface Ethernet5 is in area 0:
interface ethernet 5
ip address 10.1.0.1 255.255.0.0
```

Each **network** router configuration command is evaluated sequentially, so the specific order of these commands in the configuration is important. The communication server sequentially evaluates the *address*/*wildcard-mask* pair for each interface. See the "IP Routing Protocols Commands" chapter of the *Access and Communication Servers Command Reference* publication for more information.

Consider the first **network** command. Area ID 10.9.50.0 is configured for the interface on which subnet 131.108.20.0 is located. Assume that a match is determined for Ethernet interface 0. Ethernet interface 0 is attached to Area 10.9.50.0 only.

The second **network** command is evaluated next. For Area 2, the same process is then applied to all interfaces (except Ethernet interface 0). Assume that a match is determined for Ethernet interface 1. OSPF is then enabled for that interface and Ethernet 1 is attached to Area 2.

This process of attaching interfaces to OSPF areas continues for all **network** commands. Note that the last **network** command in this example is a special case. With this command all available interfaces (not explicitly attached to another area) are attached to Area 0.

Example 3: Internal, Area Border, and Autonomous System Boundary Routers

The following example outlines a configuration for several communication servers within a single OSPF autonomous system. Figure 18-4 provides a general network map that illustrates this example configuration.

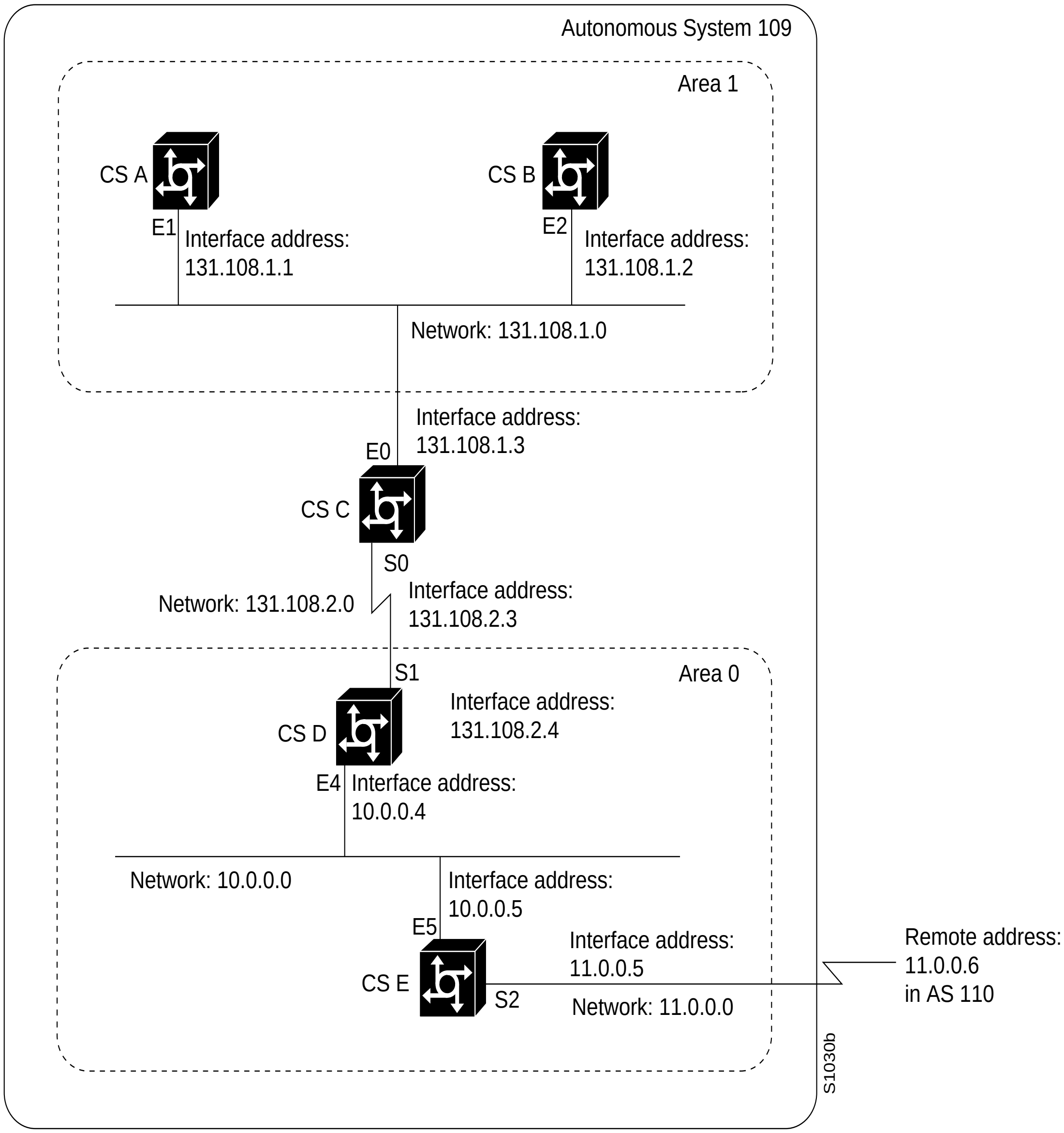

Figure 18-4 Sample OSPF Autonomous System Network Map

In this configuration, five communication servers are configured in OSPF autonomous system 109:

- CS A and CS B are both acting as internal routers within Area 1.
- CS C acts as an OSPF area border router; note that for CS C, Area 1 is assigned to Ethernet interface 0 and Area 0 is assigned to serial interface 0.
- CS D acts as an internal router in Area 0 (backbone area); in this case, both **network** router configuration commands specify the same area (Area 0, or the backbone area).
- CS E acts as an OSPF autonomous system boundary router; note that BGP routes are redistributed into OSPF and that these routes are advertised by OSPF.

Note It is not necessary to include definitions of all areas in an OSPF autonomous system in the configuration of all communication servers in the autonomous system. You need only define the *directly* connected areas. In the example that follows, routes in Area 0 are learned by the communication servers in Area 1 (CS A and CS B) when the area border router (CS C) injects summary link state advertisements (LSAs) into Area 1.

Autonomous System 109 is connected to the outside world via the BGP link to the external peer at IP address 11.0.0.6.

Configuration for Communication Server A - Internal Router

```
interface ethernet 1
ip address 131.108.1.1 255.255.255.0

router ospf 109
network 131.108.0.0 0.0.255.255 area 1
```

Configuration for Communication Server B - Internal Router

```
interface ethernet 2
ip address 131.108.1.2 255.255.255.0

router ospf 109
network 131.108.0.0 0.0.255.255 area 1
```

Configuration for Communication Server C - Area Border Router

```
interface ethernet 0
ip address 131.108.1.3 255.255.255.0

interface serial 0
ip address 131.108.2.3 255.255.255.0

router ospf 109
network 131.108.1.0 0.0.0.255 area 1
network 131.108.2.0 0.0.0.255 area 0
```

Configuration for Communication Server D - Internal Router

```
interface ethernet 4
ip address 10.0.0.4 255.0.0.0

interface serial 1
ip address 131.108.2.4 255.255.255.0

router ospf 109
network 131.108.2.0 0.0.0.255 area 0
network 10.0.0.0 0.255.255.255 area 0
```

Configuration for Communication Server E - Autonomous System Boundary Router

```
interface ethernet 5
ip address 10.0.0.5 255.0.0.0

interface serial 2
ip address 11.0.0.5 255.0.0.0

router ospf 109
network 10.0.0.0 0.255.255.255 area 0
redistribute bgp 109 metric 1 metric-type 1

router bgp 109
network 131.108.0.0
network 10.0.0.0
neighbor 11.0.0.6 remote-as 110
```

Example 4: Complex OSPF Configuration

The following example configuration accomplishes several tasks in setting up a communication server to act as an area border router. These tasks can be split into two general categories:

- Basic OSPF configuration
- Route redistribution

The specific tasks outlined in this configuration are detailed briefly in the following descriptions. Figure 18-5 illustrates the network address ranges and area assignments for the interfaces.

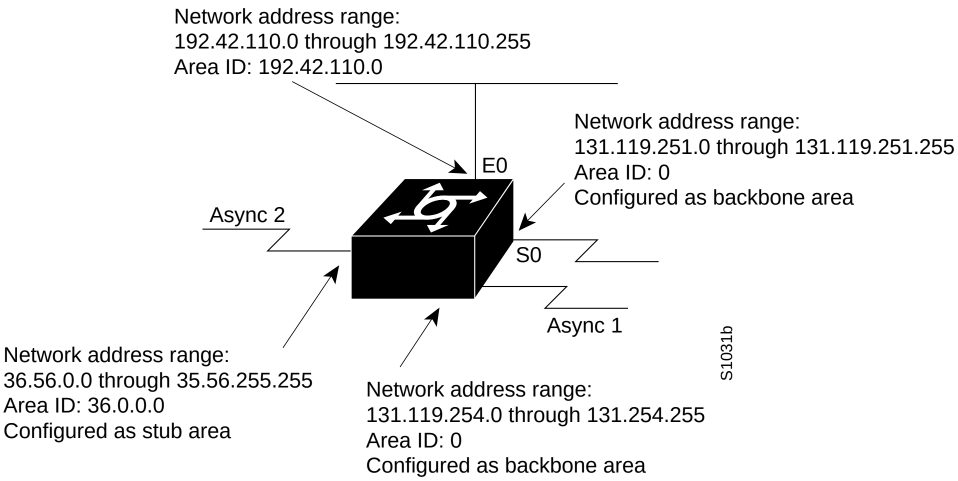

Figure 18-5 Interface and Area Specifications for OSPF Example Configuration

The basic configuration tasks in this example are as follows:

- Configure address ranges for Ethernet 0 through Ethernet 3 interfaces.
- Enable OSPF on each interface.
- Set up an OSPF authentication password for each area and network.
- Assign link state metrics and other OSPF interface configuration options.
- Create a *stub area* with area id 36.0.0.0. (Note that the **authentication** and **stub** options of the **area** router configuration command are specified with separate **area** command entries, but can be merged into a single **area** command.)
- Specify the backbone area (Area 0).

Configuration tasks associated with redistribution are as follows:

- Redistribute IGRP and RIP into OSPF with various options set (including **metric-type**, **metric**, **tag**, and **subnet**).
- Redistribute IGRP and OSPF into RIP.

The following is an example OSPF configuration:

```
interface ethernet 0
ip address 192.42.110.201 255.255.255.0
ip ospf authentication-key abcdefgh
ip ospf cost 10
!
interface ethernet 1
ip address 131.119.251.201 255.255.255.0
ip ospf authentication-key ijklmnop
ip ospf cost 20
ip ospf retransmit-interval 10
ip ospf transmit-delay 2
ip ospf priority 4
!
interface ethernet 2
ip address 131.119.254.201 255.255.255.0
ip ospf authentication-key abcdefgh
ip ospf cost 10
!
interface ethernet 3
ip address 36.56.0.201 255.255.0.0
ip ospf authentication-key ijklmnop
ip ospf cost 20
ip ospf dead-interval 80
```

OSPF is on network 131.119:

```
router ospf 201
network 36.0.0.0 0.255.255.255 area 36.0.0.0
network 192.42.110.0 0.0.0.255 area 192.42.110.0
network 131.119.0.0 0.0.255.255 area 0
area 0 authentication
area 36.0.0.0 stub
area 36.0.0.0 authentication
area 36.0.0.0 default-cost 20
area 192.42.110.0 authentication
area 36.0.0.0 range 36.0.0.0 255.0.0.0
area 192.42.110.0 range 192.42.110.0 255.255.255.0
area 0 range 131.119.251.0 255.255.255.0
area 0 range 131.119.254.0 255.255.255.0

redistribute igrp 200 metric-type 2 metric 1 tag 200 subnets
redistribute rip metric-type 2 metric 1 tag 200
```

IGRP AS 200 is on 131.119.0.0:

```
router igrp 200
network 131.119.0.0
!
! RIP for 192.42.110
!
router rip
network 192.42.110.0
redistribute igrp 200 metric 1
redistribute ospf 201 metric 1
```

BGP Route Advertisement and Redistribution Examples

The following examples illustrate configurations for advertising and redistributing BGP routes. The first example details the configuration for two neighboring routers that run IGRP within their respective autonomous systems and that are configured to advertise their respective BGP routes to each other. The second example illustrates route redistribution of BGP into IGRP, and IGRP into BGP.

Example 1: Simple BGP Route Advertisement

This example provides the required configuration for two routers (R1 and R2) that are intended to advertise BGP routes to each other and to redistribute BGP into IGRP.

Configuration for Router R1

```
! Assumes autonomous system 1 has network number 131.108.0.0
router bgp 1
network 131.108.0.0
neighbor 192.5.10.1 remote-as 2
!
router igrp 1
network 131.108.0.0
network 192.5.10.0
redistribute bgp 1
! Note that IGRP is not redistributed into BGP
```

Configuration for Router R2

```
router bgp 2
network 150.136.0.0
neighbor 192.5.10.2 remote-as 1
!
router igrp 2
network 150.136.0.0
network 192.5.10.0
redistribute bgp 2
```

Example 2: Mutual Route Redistribution

The most complex redistribution case is one in which *mutual* redistribution is required between an IGP (in this case IGRP) and BGP.

Suppose that EGP is running on a router somewhere else in AS 1, and that the EGP routes are injected into IGRP routing process 1. You must filter to ensure that the proper routes are advertised. The example configuration for router R1 illustrates use of access filters and a distribution list to filter routes advertised to BGP neighbors. This example also illustrates configuration commands for redistribution between BGP and IGRP. Only routes learned using the EBGP session with neighbors 192.5.10.1 and 192.5.10.24 are redistributed into IGRP.

Configuration for Router R1

```
router bgp 1
network 131.108.0.0
neighbor 192.5.10.1 remote-as 2
! External peer or neighbor
neighbor 131.108.2.1 remote-as 1
! Same AS; therefore internal neighbor
neighbor 192.5.10.24 remote-as 3
! A second external neighbor: 192.5.10.24 is an external peer
redistribute igrp 1
distribute-list 1 out igrp 1
!
! All networks that should be
! advertised from R1 are
! controlled with access lists:
!
access-list 1 permit 131.108.0.0
access-list 1 permit 150.136.0.0
access-list 1 permit 128.125.0.0
!
router igrp 1
network 131.108.0.0
network 192.5.10.0
redistribute bgp 1
```

Default Metric Values Redistribution Examples

The following example shows a communication server in autonomous system 109 using both the RIP and the IGRP routing protocols. The example advertises IGRP-derived routes using the RIP protocol and assigns the IGRP-derived routes a RIP metric of 10.

```
router rip
default-metric 10
redistribute igrp 109
```

Route-Map Examples

The examples in this section illustrate the use of redistribution, with and without route maps. Examples from both the IP and CLNS routing protocols are given.

The following example redistributes all OSPF routes into IGRP:

```
router igrp 109
redistribute ospf 110
```

The following example redistributes all OSPF routes into IP Enhanced IGRP:

```
router eigrp 109
redistribute ospf 110
default-metric 1000 100 255 1 1500
```

The following example redistributes RIP routes with a hop count equal to 1 into OSPF. These routes will be redistributed into OSPF as external link state advertisements with a metric of 5, metric type of Type 1, and a tag equal to 1.

```
router ospf 109
redistribute rip route-map rip-to-ospf
!
route-map rip-to-ospf permit
match metric 1
set metric 5
set metric-type type1
set tag 1
```

The following example redistributes OSPF learned routes with tag 7 as a RIP metric of 15:

```
router rip
redistribute ospf 109 route-map 5
!
route-map 5 permit
match tag 7
set metric 15
```

The following example redistributes OSPF intra-area and interarea routes with next-hop communication servers on serial interface 0 into BGP with an INTER_AS metric of 5:

```
router bgp 109
redistribute ospf 109 route-map 10
!
route-map 10 permit
match route-type internal
match interface serial 0
set metric 5
```

With the following configuration, OSPF external routes with tags 1, 2, 3, and 5 are redistributed into RIP with metrics of 1, 1, 5, and 5, respectively. The OSPF routes with a tag of 4 are not redistributed.

```
router rip
redistribute ospf 109 route-map 1
!
route-map 1 permit
match tag 1 2
set metric 1
!
route-map 1 permit
match tag 3
set metric 5
!
route-map 1 deny
match tag 4
!
route map 1 permit
match tag 5
set metric 5
```

The following configuration sets the condition that if there is an OSPF route to network 140.222.0.0, generate the default network 0.0.0.0 into RIP with a metric of 1:

```
router rip
redistribute ospf 109 route-map default
!
route-map default permit
match ip address 1
set metric 1
!
access-list 1 permit 140.222.0.0 0.0.255.255
access-list 2 permit 0.0.0.0 0.0.0.0
```

The following configuration example illustrates how a route map is referenced by the **default-information** router configuration command. This is called *conditional default origination*. OSPF will originate the default route (network 0.0.0.0) with a Type 2 metric of 5 if 140.222.0.0, with network 0.0.0.0 in the routing table.

```
route-map ospf-default permit
match ip address 1
set metric 5
set metric-type type-2
!
access-list 1 140.222.0.0 0.0.255.255
!
router ospf 109
default-information originate route-map ospf-default
```

Using Route Maps with BGP

The following example shows how you can use route maps to modify incoming data from a neighbor. Any route received from 140.222.1.1 that matches the filter parameters set in autonomous system access list 200 will have its weight set to 200 and its local preference set to 250 and will be accepted.

```
router bgp 100
!
neighbor 140.222.1.1 route-map fix-weight in
neighbor 140.222.1.1 remote-as 1
!
route-map fix-weight permit 10
match as-path 200
set local-preference 250
set weight 200
!
ip as-path access-list 200 permit ^690$
ip as-path access-list 200 permit ^1800
```

The following example shows how you can use route maps to modify outbound data to a neighbor:

```
router bgp 100
neighbor 198.92.68.23 route-map oscar out
!
route-map oscar
set metric 150
match as-path 1
!
ip as-path access-list 1 permit ^2200_
```

In the following example, route map freddy marks all paths originating from autonomous system 690 with a multiple exit discriminator (MULTI_EXIT_DISC) metric attribute of 127. The second permit clause is required so that routes not matching autonomous system path list 1 will still be accepted from neighbor 1.1.1.1.

```
router bgp 100
neighbor 1.1.1.1 route-map freddy in
!
ip as-path access-list 1 permit ^690_
ip as-path access-list 2 permit .*
!
route-map freddy permit 10
match as-path 1
set metric 127
!
route-map freddy permit 20
match as-path 2
```

The following example shows how you can use route maps to modify incoming data from the IP forwarding table:

```
router bgp 100
redistribute igrp 109 route-map igrp2bgp
!
route-map igrp2bgp
match ip address 1
set local-preference 25
set metric 127
set weight 30000
set next-hop 192.92.68.24
set origin igp
!
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 1 permit 160.89.0.0 0.0.255.255
access-list 1 permit 198.112.0.0 0.0.127.255
```

It is proper behavior not to accept any autonomous system path not matching the **match** clause of the route map. This means that you will not set the metric and the communication server will not accept the route. However, you can configure the communication server to accept autonomous system paths not matched in the **match** clause of the route map command by using multiple maps of the same name, some without accompanying **set** commands.

```
route-map fnord permit 10
match as-path 1
set local-preference 5
!
route-map fnord permit 20
match as-path 2
```

The following example shows how you can use route maps in a reverse operation to set the route tag (as defined by the BGP/OSPF interaction document, RFC 1403) when exporting routes from BGP into the main IP routing table:

```
router bgp 100
table-map set_ospf_tag
!
route-map set_ospf_tag
match as-path 1
set automatic-tag
!
ip as-path access-list 1 permit .*
```

Route Summarization Example

The following example configures route summarization on the interface and also configures the auto-summary feature. This configuration causes IP Enhanced IGRP to summarize network 10.0.0.0 out Ethernet interface 0 only. In addition, this example disables auto summarization.

```
interface Ethernet 0
ip summary-address eigrp 1 10.0.0.0 255.0.0.0
!
router eigrp 1
network 131.108.0.0
no auto-summary
```

IGRP Feasible Successor Relationship Example

In Figure 18-6, the assigned metrics meet the conditions required for a feasible successor relationship, so the paths in this example can be included in routing tables and used for load balancing.

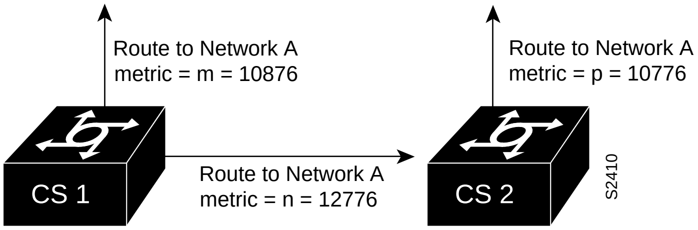

Figure 18-6 Assigning Metrics for IGRP Path Feasibility

The feasibility test would work as follows:

Assume that CS1 already has a route to Network A with metric m and has just received an update about Network A from CS2. The best metric at CS2 is p. The metric that CS1 would use through CS2 is n.

If both of the following two conditions are met, the route to network A through CS2 will be included in CS1's routing table:

- The *m* value is greater than *p*.
- The *multiplier* (value specified by the **variance** router configuration command) times *m* is greater than or equal to *n*.

The configuration for CS1 would be as follows:

```
router igrp 109
variance 10
```

A maximum of four paths can be in the routing table for a single destination. If there are more than four feasible paths, the four best feasible paths are used.

BGP Synchronization Example

In Figure 18-7, with synchronization on, CS B will not advertise network 10.0.0.0 to communication server A until an IGRP route for network 10.0.0.0 exists. If you specify the **no synchronization** router configuration command, communication server B will advertise network 10.0.0.0 as soon as possible. However, because routing information still must be sent to interior peers, you must configure a full internal BGP mesh.

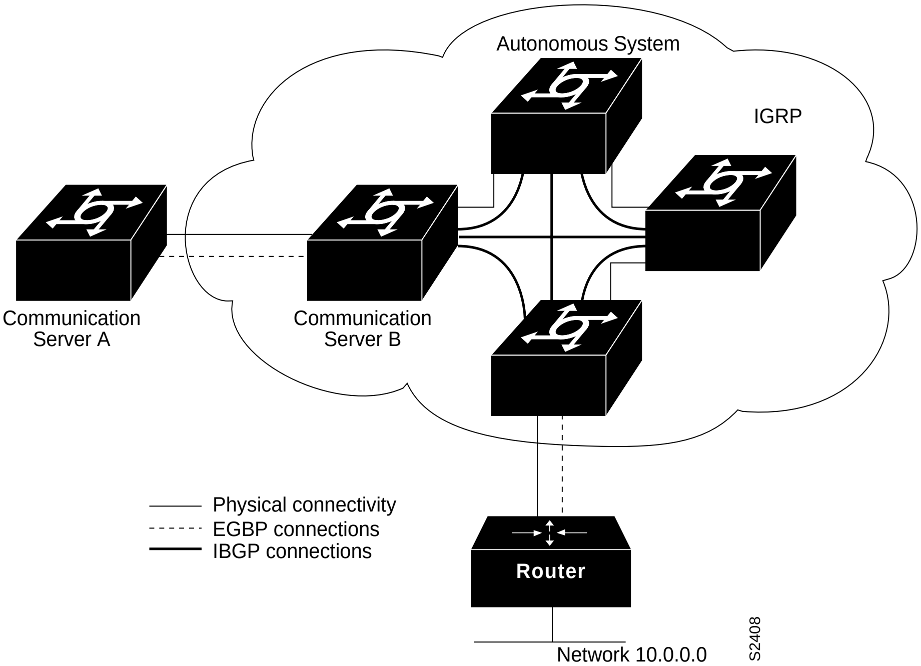

Figure 18-7 Illustration of Synchronization

BGP Basic Neighbor Specification Examples

The following example specifies that the communication server at the address 131.108.1.2 is a neighbor in autonomous system 109.

```
neighbor 131.108.1.2 remote-as 109
```

In the following example, a BGP communication server is assigned to autonomous system 109, and two networks are listed as originating in the autonomous system. Then the addresses of three remote communication servers (and their autonomous systems) are listed. The communication server being configured will share information about networks 131.108.0.0 and 192.31.7.0 with the neighbor communication servers. The first communication server listed is in the same Class B network, but in a different autonomous system; the second **neighbor** router configuration command illustrates specification of an internal neighbor (with the same autonomous system number) at address 131.108.234.2; and the last **neighbor** command specifies a neighbor on a different network.

```
router bgp 109
network 131.108.0.0
network 192.31.7.0
neighbor 131.108.200.1 remote-as 167
neighbor 131.108.234.2 remote-as 109
neighbor 150.136.64.19 remote-as  99
```

In Figure 18-8, communication server A is being configured. The internal BGP neighbor is not directly linked to communication server A. External neighbors (in autonomous system 167 and autonomous system 99) must be linked directly to communication server A.

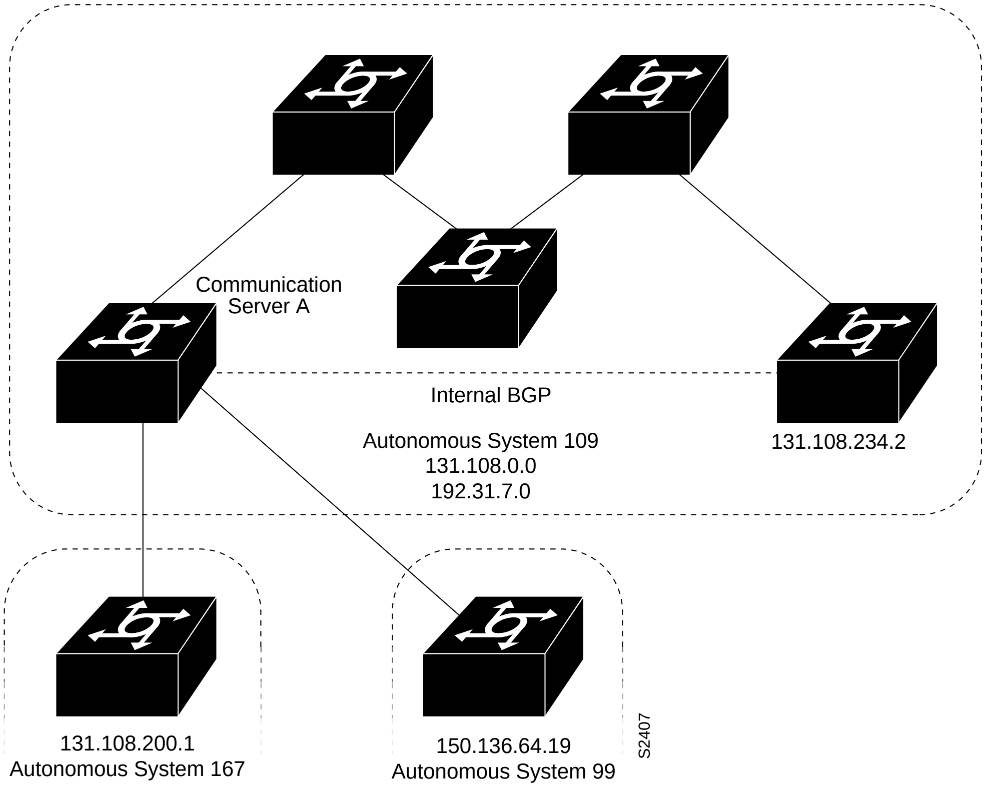

Figure 18-8 Assigning Internal and External BGP Neighbors

Using Access Lists to Specify Neighbors

In the following example, the communication server is configured to allow connections from any communication server that has an IP address in access list 1; that is, any communication server with a 192.31.7.x address. Neighbors not explicitly specified as neighbors can connect to the communication server, but the communication server will not attempt to connect to them if the connection is broken. Continuing with the preceding sample configuration, the configuration is as follows:

```
router bgp 109
network 131.108.0.0
network 192.31.7.0
neighbor 131.108.200.1  remote-as 167
neighbor 131.108.234.2 remote-as 109
neighbor 150.136.64.19  remote-as  99
neighbor internal-ethernet neighbor-list 1
access-list 1 permit 192.31.7.0 0.0.0.255
```

BGP Aggregate Route Examples

The following examples show how you can use aggregate routes in BGP either by redistributing an aggregate route into BGP or by using the conditional aggregate routing feature.

In the following example, the **redistribute static** command is used to redistribute aggregate route 193.*.*.*:

```
ip route 193.0.0.0 255.0.0.0 null 0
!
router bgp 100
redistribute static
network 193.0.0.0 255.0.0.0
! this marks route as not incomplete
```

The following configuration creates an aggregate entry in the BGP routing table when there are specific routes that fall into the specified range. The aggregate route will be advertised as coming from your autonomous system and has the atomic aggregate attribute set to show that information might be missing. (By default, atomic aggregate is set unless you use the **as-set** keyword in the **aggregate-address** command.)

```
router bgp 100
aggregate-address 193.0.0.0 255.0.0.0
```

The following example creates an aggregate entry using the same rules as in the previous example, but the path advertised for this route will be an AS_SET consisting of all elements contained in all paths that are being summarized:

```
router bgp 100
aggregate-address 193.0.0.0 255.0.0.0 as-set
```

The following example not only creates the aggegate route for 193.*.*.*, but will also suppress advertisements of more specific routes to all neighbors:

```
router bgp 100
aggegate-address 193.0.0.0 255.0.0.0 summary-only
```

Third-Party EGP Support Example

In the following example configuration, the communication server is in autonomous system 110 communicating with an EGP neighbor in autonomous system 109 with address 131.108.6.5. Network 131.108.0.0 is advertised as originating within autonomous system 110. The configuration specifies that two communication servers, 131.108.6.99 and 131.108.6.100, should be advertised as third-party sources of routing information for those networks that are accessible through those communication servers. The global configuration commands also specify that those networks should be flagged as internal to autonomous system 110.

```
autonomous-system 110
router egp 109
network 131.108.0.0
neighbor 131.108.6.5
neighbor 131.108.6.5  third-party 131.108.6.99 internal
neighbor 131.108.6.5  third-party 131.108.6.100 internal
```

Backup EGP Communication Server Example

The following example configuration illustrates a communication server that is in autonomous system 110 communicating with an EGP neighbor in autonomous system 109 with address 131.108.6.5. Network 131.108.0.0 is advertised with a distance of 1, and networks learned by RIP are being advertised with a distance of 5. Access list 3 filters which RIP-derived networks are allowed in outgoing EGP updates.

```
autonomous-system 110
router egp 109
network 131.108.0.0
neighbor 131.108.6.5
redistribute rip
default-metric 5
distribute-list 3 out rip
```

EGP Core Gateway Example

The following example illustrates how an EGP core gateway can be configured.

Figure 18-9 illustrates an environment with three communication servers (designated CS 1, CS 2, and CS 3) attached to a common X.25 network. The communication servers are intended to route information using EGP. With the following configuration (on the communication server designated core), CS 1, CS 2, and CS 3 cannot route traffic directly to each other via the X.25 network:

```
access-list 1 permit 10.0.0.0 0.255.255.255
! global access list assignment
router egp 0
neighbor any 1
```

This configuration specifies that an EGP process on any communication server on network 10.0.0.0 can act as a peer with the core communication server. All traffic in this configuration will flow through the core communication server.

Third-party advertisements allow traffic to bypass the core communication server and go directly to the communication server that advertised reachability to the core.

```
access-list 2 permit 10.0.0.0 0.255.255.255
! global access list assignment
router egp 0
neighbor any 2
neighbor any third-party 10.1.1.1
```

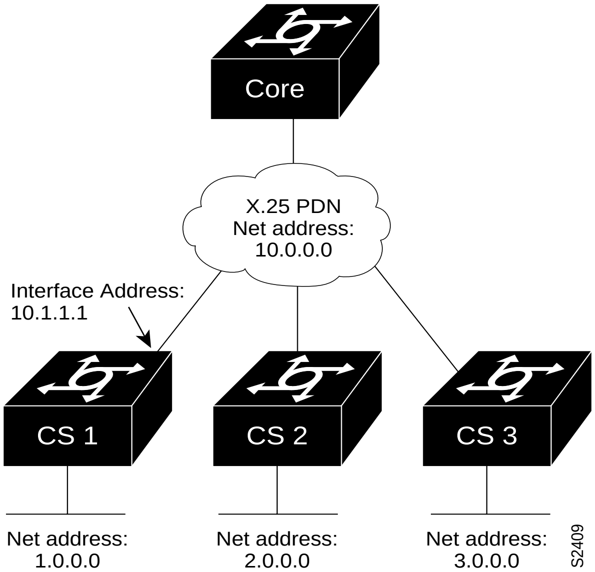

Figure 18-9 Core EGP Third-Party Update Configuration Example

Autonomous System within EGP Example

The following example illustrates a typical configuration for an EGP communication server process. The communication server is in autonomous system 109 and is peering with communication servers in autonomous system 164, as shown in Figure 18-10. It will advertise the networks 131.108.0.0 and 192.31.7.0 to the communication server in autonomous system 164, 10.2.0.2. The information sent and received from peer communication servers can be filtered in various ways, including blocking information from certain communication servers and suppressing the advertisement of specific routes.

```
autonomous-system 109
router egp 164
network 131.108.0.0
network 192.31.7.0
neighbor 10.2.0.2
```

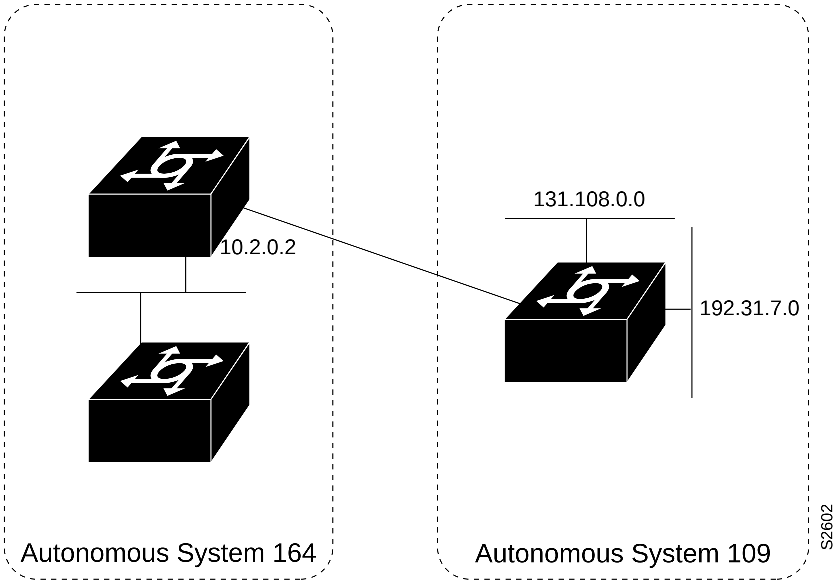

Figure 18-10 Router in AS 164 Peers with Communication Server in AS 109

Passive Interface Examples

The following example sends IGRP updates to all interfaces on network 131.108.0.0 except Ethernet interface 1. Figure 18-11 shows this configuration.

```
router igrp 109
network 131.108.0.0
passive-interface ethernet 1
```

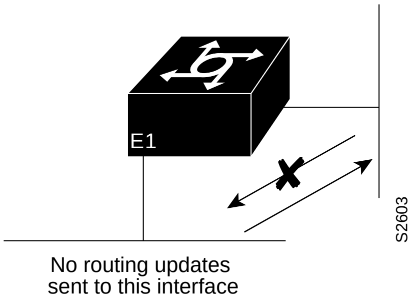

Figure 18-11 Filtering IGRP Updates

As in the first example, IGRP updates are sent to all interfaces on network 131.108.0.0 except Ethernet interface 1 in the following example. However, in this case a **neighbor** router configuration command is included. This command permits the sending of routing updates to specific neighbors. One copy of the routing update is generated per neighbor.

```
router igrp 109
network 131.108.0.0
passive-interface ethernet 1
neighbor 131.108.20.4
```

In OSPF, hello packets are not sent on an interface that is specified as passive. Hence, the communication server will not be able to discover any neighbors, and none of the OSPF neighbors will be able to see the communication server on that network. In effect, this interface will appear as a stub network to the OSPF domain. This is useful if you want to import routes associated with a connected network into the OSPF domain without any OSPF activity on that interface.

The **passive-interface** router configuration command typically is used when the wildcard specification on the **network** router configuration command configures more interfaces than is desirable. The following configuration causes OSPF to run on all subnets of 131.108.0.0:

```
interface ethernet 0
ip address 131.108.1.1 255.255.255.0
interface ethernet 1
ip address 131.108.2.1 255.255.255.0
interface ethernet 2
ip address 131.108.3.1 255.255.255.0
!
router ospf 109
network 131.108.0.0 0.0.255.255 area 0
```

If you do not want OSPF to run on 131.108.3.0, enter the following commands:

```
router ospf 109
network 131.108.0.0 0.0.255.255 area 0
passive-interface ethernet 2
```

Administrative Distance Examples

In the following example, the **router igrp** global configuration command sets up IGRP routing in autonomous system 109. The **network** router configuration commands specify IGRP routing on networks 192.31.7.0 and 128.88.0.0. The first **distance** router configuration command sets the default administrative distance to 255, which instructs the communication server to ignore all routing updates from communication servers for which an explicit distance has not been set. The second **distance** command sets the administrative distance to 90 for all communication servers on the Class C network 192.31.7.0. The third **distance** command sets the administrative distance to 120 for the communication server with the address 128.88.1.3.

```
router igrp 109
network 192.31.7.0
network 128.88.0.0
distance 255
distance 90  192.31.7.0  0.0.0.255
distance 120  128.88.1.3  0.0.0.0
```

The following example assigns the communication server with the address 192.31.7.18 an administrative distance of 100, and all other communication servers on subnet 192.31.7.0 an administrative distance of 200:

```
distance 100 192.31.7.18  0.0.0.0
distance 200 192.31.7.0  0.0.0.255
```

However, if you reverse the order of these commands, all communication servers on subnet 192.31.7.0 are assigned an administrative distance of 200, including the communication server at address 192.31.7.18:

```
distance 200 192.31.7.0  0.0.0.255
distance 100 192.31.7.18  0.0.0.0
```

Assigning administrative distances is a problem unique to each network and is done in response to the greatest perceived threats to the connected network. Even when general guidelines exist, the network manager must ultimately determine a reasonable matrix of administrative distances for the network as a whole.

In the following example, the distance value for IP routes learned is 90. Preference is given to these IP routes rather than routes with the default administrative distance value of 110.

```
router isis
distance 90 ip
```

Split Horizon Examples

Two examples of configuring split horizon are provided.

Example 1

The following sample configuration illustrates a simple example of disabling split horizon on a serial link. In this example, the serial link is connected to an X.25 network.

```
interface serial 0
encapsulation x25
no ip split-horizon
```

Example 2

In the next example, Figure 18-12 illustrates a typical situation in which the **no ip split-horizon** interface configuration command would be useful. This figure depicts two IP subnets that are both accessible via a serial interface on CS C (connected to Frame Relay network). In this example, the serial interface on CS C accommodates one of the subnets via the assignment of a secondary IP address.

The Ethernet interfaces for CS A, CS B, and CS C (connected to IP networks 12.13.50.0, 10.20.40.0, and 20.155.120.0) all have split horizon *enabled* by default, while the serial interfaces connected to networks 128.125.1.0 and 131.108.1.0 all have split horizon *disabled* by default. The partial interface configuration specifications for each communication server that follow Figure 18-12 illustrate that the **ip split-horizon** command is *not* explicitly configured under normal conditions for any of the interfaces.

In this example, split horizon must be disabled in order for network 128.125.0.0 to be advertised into network 131.108.0.0, and vice versa. These subnets overlap at CS C, serial interface 0. If split horizon were enabled on serial interface 0, it would not advertise a route back into the Frame Relay network for either of these networks.

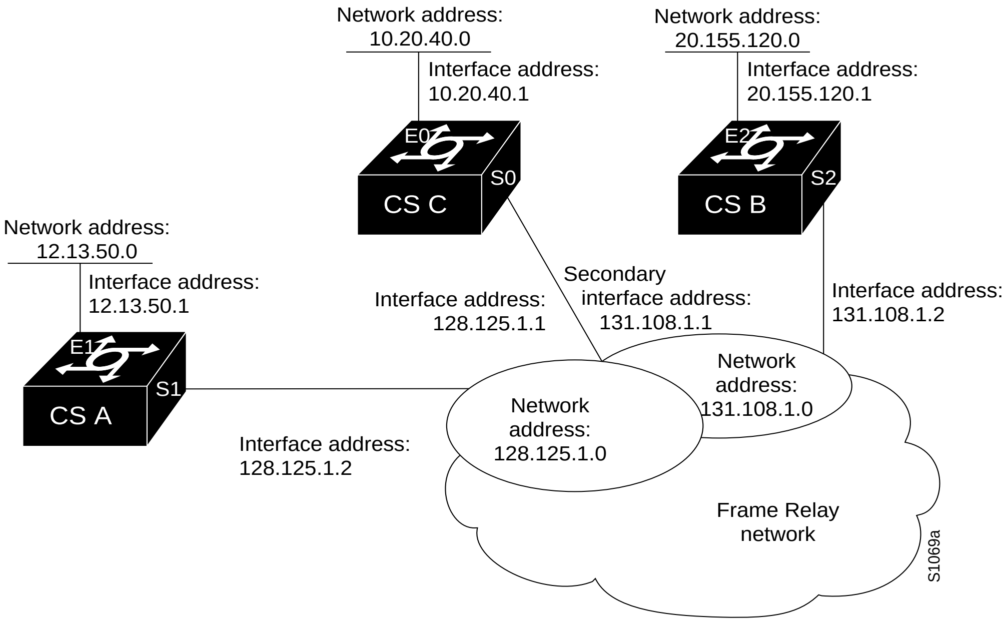

Figure 18-12 Disabled Split Horizon Example for Frame Relay Network

Configuration for CS A

```
interface ethernet 1
ip address 12.13.50.1
!
interface serial 1
ip address 128.125.1.2
encapsulation frame-relay
```

Configuration for CS B

```
interface ethernet 2
ip address 20.155.120.1
!
interface serial 2
ip address 131.108.1.2
encapsulation frame-relay
```

Configuration for CS C

```
interface ethernet 0
ip address 10.20.40.1
!
interface serial 0
ip address 128.125.1.1
ip address 131.108.1.1 secondary
encapsulation frame-relay
```

IP Multicast Routing Configuration Examples

This section provides the following IP multicast routing configuration examples:

- Configure a Communication Server to Operate in Dense Mode Example
- Configure a Communication Server to Operating in Spare Mode Example
- Configure DVMRP Interoperability Examples

Configure a Communication Server to Operate in Dense Mode Example

The following example configures dense-mode PIM on an Ethernet interface of the communication server:

```
ip multicast-routing
interface ethernet 0
ip pim dense-mode
```

Configure a Communication Server to Operating in Spare Mode Example

The following example configures the communication server to operate in sparse-mode PIM. The RP communication server is the communication server whose address is 10.8.0.20.

```
access-list 1 permit 224.2.0.1
!
ip multicast-routing
ip pim rp-address 10.8.0.20 1
interface ethernet 1
ip pim sparse-mode
```

Configure DVMRP Interoperability Examples

The following example configures DVMRP interoperability for configurations when the PIM router and the DVMRP router are on the same network segment. In this example, access list 1 advertises the networks (98.92.35.0,198.92.36.0, 198.92.37.0,131.108.0.0, and 150.136.0.0 to the DVMRP router, and access list 2 is used to prevent all other networks from being advertised (**ip dvmrp metric 0**).

```
interface ethernet 0
ip address 131.119.244.244 255.255.255.0
ip pim dense-mode
ip dvmrp metric 1 1
ip dvmrp metric 0 2

access-list 1 permit 198.92.35.0 0.0.0.255
access-list 1 permit 198.92.36.0 0.0.0.255
access-list 1 permit 198.92.37.0 0.0.0.255
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 1 permit 150.136.0.0 0.0.255.255
access-list 1 deny   0.0.0.0 255.255.255.255
access-list 2 permit 0.0.0.0 255.255.255.255
```

The following example configures DVMRP interoperability over a tunnel interface.

```
hostname Mbone(ACOnet)
!
boot system xx-k.Aug11 192.76.243.9
boot system flash
boot system rom
!
ip multicast-routing
!
interface tunnel 0
no ip address
ip pim dense-mode
tunnel source Ethernet0
tunnel destination 192.70.92.133
tunnel mode dvmrp
!
interface Ethernet0
description Universitat DMZ-ethernet
ip address 193.171.23.23 255.255.255.240 secondary
ip address 192.76.243.2 255.255.255.0
ip pim dense-mode
!
router igrp 11853
network 192.76.243.0
network 193.171.23.0
```