CHAPTER 17

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 *Router Products Command Reference* publication. For information on configuring the IP protocol, refer to the "Configuring IP" chapter of this manual. 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 of the IP routing protocols is discussed in detail at the beginning of the individual protocol 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 following sections.

Note Many routing protocol specifications refer to routers as *gateways*, so the word *gateway* often appears as part of routing protocol names. However, a router 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 whether a routing protocol name contains the word "gateway" or not, 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 routers 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)

Note Enhanced IGRP is documented in another publication.

- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP)
- Intermediate System-to-Intermediate System (IS-IS)

The Exterior Gateway Protocols

Exterior protocols are used to exchange routing information between networks that do not share a common administration. IP exterior gateway protocols require three sets of information before routing can begin:

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

The supported exterior routing protocols are as follows:

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

Router Discovery Protocols

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

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

Multiple Routing Protocols

You can configure multiple routing protocols in a single router 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 routers can handle simultaneous operation of up to 30 dynamic IP routing processes.The combination of routing processes on a router 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 IS-IS 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 OSPF
- Configure RIP
- Configure IS-IS
- Configure BGP
- Configure EGP
- Configure GDP
- Configure IRDP
- 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); IS-IS, static routes, and OSPF support VLSM.
- 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.

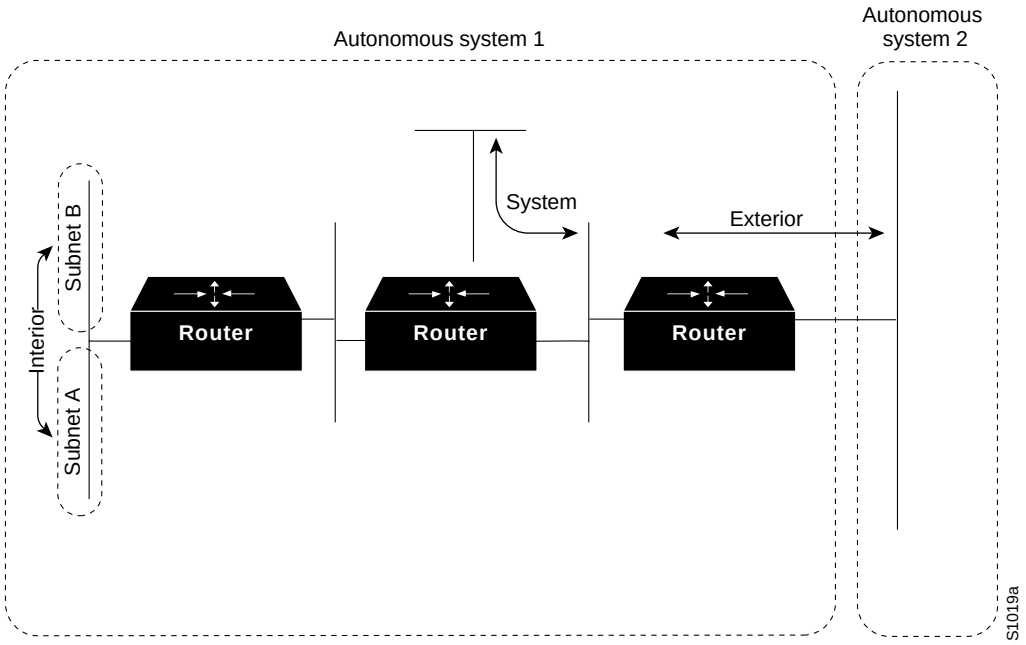
Note Enhanced IGRP is documented in another publication.

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 1-1. Interior routes are routes between subnets in the network attached to a router interface. If the network attached to a router is not subnetted, IGRP does not advertise interior routes.

Figure 1-1 Interior, System, and Exterior Routes



System routes are routes to networks within an autonomous system. The router derives system routes from directly connected network interfaces and system route information provided by other IGRP-speaking routers. 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 router chooses a gateway of last resort from the list of exterior routes that IGRP provides. The router uses the gateway (router) 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 routers can choose different exterior routers as the gateway of last resort.

IGRP Updates

By default, a router running IGRP sends an update broadcast every 90 seconds. It declares a route inaccessible if it does not receive an update from the first router in the route within three update periods (270 seconds). After seven update periods (630 seconds), the router 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 routers 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. It is only mandatory to create the IGRP routing process; 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.
Step 2	Enable an IGRP routing process, which places you in router configuration mode.	**router igrp** *autonomous-system*
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 router to permit this exchange of routing information.

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

Task	Command
Define a neighboring router 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 section in this chapter titled "Filter Routing Information."

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 router in the path is *closer* to the destination (has a lower metric value) than the current router 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 router; that is, the next-hop router 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 router.

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 router 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 router 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. This allows, for example, tuning system behavior to allow for 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 router 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 router 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 routers 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 router 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 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 conforms to the OSPF Version 2 specifications detailed in the Internet RFC 1247. 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, RIP, and IS-IS. OSPF routes also can be exported into IGRP, RIP, and IS-IS. 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 routers within an area is supported.

- Routing interface parameters—Configurable parameters supported include interface output cost, retransmission interval, interface transmit delay, router priority, router "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 routers, *area border routers* (routers connected to multiple areas), and autonomous system boundary routers. At a minimum, OSPF-based routers 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 routers.

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
- Configure OSPF on Simplex Ethernet Interfaces

In addition, you can specify route redistribution; see the task "Redistribute Routing Information" 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 routers in an attached network. Therefore, be sure that if you do configure any of these parameters, the configurations for all routers 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 router priority to help determine the OSPF designated router for a network.	**ip ospf priority** *number*
Specify the length of time, in seconds, between the hello packets that a router sends on an OSPF interface.	**ip ospf hello-interval** *seconds*
Set the number of seconds that a router's hello packets must not have been seen before its neighbors declare the OSPF router down.	**ip ospf dead-interval** *seconds*
Assign a specific password to be used by neighboring OSPF routers 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, FDDI)
- 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 *Router Products 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 routers 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 the section following this one.

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 routers attached to an OSPF network, a *designated router* is selected for the network. It is necessary to use special configuration parameters in the designated router selection if broadcast capability is not configured.

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

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

Task	Command
Configure routers 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 router
- Nonbroadcast poll interval
- Interface through which the neighbor is reachabl

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 an area border router. In OSPF, 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 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 area border routers. The virtual link must be configured in both routers. The configuration information in each router consists of the other virtual endpoint (the other area border router), and the nonbackbone area that the two routers 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*]

Use the **show ip ospf virtual-links** EXEC command to display virtual link information. Use the **show ip ospf** EXEC command to display the router ID of an OSPF router.

Generate a Default Route

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

To force the 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 router, because it is displayed by name rather than by its router 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 router'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 router will use this IP address as its router ID, even if other interfaces have larger IP addresses. Since 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 router 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 *Router Products Command Reference* publication.

Configure OSPF on Simplex Ethernet Interfaces

Because simplex interfaces between two routers on an Ethernet represent only one network segment, for OSPF you have to configure the transmitting interface to be a passive interface. This prevents OSPF from sending hello packets for the transmitting interface. Both routers are able to see each other via the hello packet generated for the receiving interface.

To configure OSPF on simplex Ethernet interfaces, perform the following task in router configuration mode:

Task	Command
Suppress the sending of hello packets through the specified interface.	**passive-interface** *interface*

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 router sends routing information updates every 30 seconds; this process is termed *advertising.* If a router does not receive an update from another router for 180 seconds or more, it marks the routes served by the nonupdating router as being unusable. If there is still no update after 240 seconds, the router removes all routing table entries for the nonupdating router.

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 routers 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 router has a default network path, RIP advertises a route that links the router 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 routers will advertise the default network if a default was learned by RIP, or if the router 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 router to permit this exchange of routing information.

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

Task	Command
Define a neighboring router 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 section in this chapter titled “Filter Routing Information.”

Configure IS-IS

IS-IS, which stands for Intermediate System-to-Intermediate System, is an International Organization for Standardization (ISO) dynamic routing specification. IS-IS is described in ISO 10589. Cisco’s implementation of IS-IS allows you to configure IS-IS as an IP routing protocol on your router.

IS-IS Configuration Task List

To configure IS-IS, complete the tasks in the following sections. Only enabling IS-IS is required; the remainder of the tasks are optional although you might be required to perform them depending upon your specific application.

- Enable IS-IS
- Configure IS-IS Interface Parameters
- Configure Miscellaneous IS-IS Parameters

In addition, you can filter routing information (see the task “Filter Routing Information” later in this chapter for information on how to do this), and specify route redistribution (see the task “Redistribute Routing Information” later in this chapter for information on how to do this).

Enable IS-IS

As with other routing protocols, enabling IS-IS requires that you create an IS-IS routing process and assign it to specific networks. You can specify *only one* IS-IS process per router. Only one IS-IS process is allowed whether you run it in integrated mode, ISO CLNS only, or IP only.

Network Entity Titles (NETs) define the area addresses for the IS-IS area. Multiple NETs per router are allowed, up to a maximum of three. Refer to the "Configuring ISO CLNS" chapter for a more detailed discussion of NETs.

Perform the following tasks to enable IS-IS on the router:

Task		Command
Step 1	Enter global configuration mode.	See Table 2-1.
Step 2	Enable IS-IS routing and specify an IS-IS process for IP, which places you in router configuration mode.	**router isis** [*tag*]
Step 3	Configure NETs for the routing process; you can specify a name for a NET as well as an address.	**net** *network-entity-title*
Step 4	Enter interface configuration mode.	See Table 2-1.
Step 5	Specify the interfaces that should be actively routing IS-IS.	**ip router isis** [*tag*]

See the "IP Routing Protocol Configuration Examples" section at the end of this chapter for an example of configuring IS-IS as an IP routing protocol.

Configure IS-IS Interface Parameters

Our IS-IS implementation allows you to alter certain interface-specific IS-IS parameters. You can do the following:

- Configure IS-IS link state metrics
- Set the advertised hello interval
- Set the advertised CSNP interval
- Set the retransmission interval
- Specify designated router election
- Specify the interface circuit type
- Assign a password for an interface

You are not required to alter any of these parameters, but some interface parameters must be consistent across all routers in an attached network. Therefore, be sure that if you do configure any of these parameters, the configurations for all routers on the network have compatible values.

Configure IS-IS Link-State Metrics

You can configure a cost for a specified interface. The only metric that is supported by the router and that you can configure is the *default-metric*, which you can configure for Level 1 and/or Level 2 routing. The other metrics currently are not supported.

To configure the metric for the specified interface, perform the following task in interface configuration mode:

Task	Command
Configure the metric (or cost) for the specified interface.	**isis metric** *default-metric* [*delay-metric* [*expense-metric* [*error-metric*]]] {**level-1** | **level-2**}

Set the Advertised Hello Interval

You can specify the length of time, in seconds, between hello packets that the router sends on the interface.

To specify the length of time between hello packets for the specified interface, perform the following task in interface configuration mode:

Task	Command
Specify the length of time, in seconds, between hello packets the router sends on the specified interface.	**isis hello-interval** *seconds* {**level-1** | **level-2**}

The hello interval can be configured independently for Level 1 and Level 2, except on serial point-to-point interfaces. (Because there is only a single type of hello packet sent on serial links, it is independent of Level 1 or Level 2.) Specify an optional level for X.25, SMDS, and Frame Relay multiaccess networks.

Set the Advertised CSNP Interval

Complete Sequence Number PDUs (CSNPs) are sent by the designated router to maintain database synchronization. You can configure the IS-IS CSNP interval for the interface.

To configure the CSNP interval for the specified interface, perform the following task in interface configuration mode:

Task	Command
Configure the IS-IS CSNP interval for the specified interface.	**isis csnp-interval** *seconds* {**level-1** | **level-2**}

This feature does not apply to serial point-to-point interfaces. It applies to WAN connections if the WAN is viewed as a multiaccess meshed network.

Set the Retransmission Interval

You can configure the number of seconds between retransmission of IS-IS link state PDUs (LSPs) for point-to-point links.

To set the retransmission level, perform the following task in interface configuration mode:

Task	Command
Configure the number of seconds between retransmission of IS-IS LSPs for point-to-point links.	**isis retransmit-interval** *seconds*

The value you specify should be an integer greater than the expected round-trip delay between any two routers on the attached network. The setting of this parameter should be conservative, or needless retransmission will result. The value should be larger for serial lines and virtual links.

Specify Designated Router Election

You can configure the priority to use for designated router election. Priorities can be configured for Level 1 and Level 2 individually.

To specify the designated router election, perform the following task in interface configuration mode:

Task	Command
Configure the priority to use for designated router election.	**isis priority** *value* {**level-1** | **level-2**}

Specify the Interface Circuit Type

You can specify adjacency levels on a specified interface. This parameter is also referred to as the interface circuit type.

To specify the interface circuit type, perform the following task in interface configuration mode:

Task	Command
Configure the type of adjacency desired for neighbors on the specified interface (the interface circuit type).	**isis circuit-type** {**level-1** | **level-1-2** | **level-2-only**}

Assign a Password for an Interface

You can assign different passwords for different routing levels. Specifying Level 1 or Level 2 configures the password for only Level 1 or Level 2 routing, respectively. If you do not specify a level, the default is Level 1. By default, authentication is disabled.

To configure a password for the specified level, perform the following task in interface configuration mode:

Task	Command
Configure the authentication password for a specified interface.	**isis password** *password* {**level-1** | **level-2**}

Configure Miscellaneous IS-IS Parameters

You can configure the following miscellaneous, optional IS-IS parameters:

- Generate a default route
- Specify router level support
- Configure IS-IS authentication passwords
- Summarize address ranges

Generate a Default Route

You can force a default route into an IS-IS routing domain. Whenever you specifically configure redistribution of routes into an IS-IS routing domain, the router does not, by default, generate a *default route* into the IS-IS routing domain. The following feature allows you to force the boundary router do this.

To generate a default route, perform the following task in router configuration mode:

Task	Command
Force a default route into the IS-IS routing domain.	**default-information originate** [**metric** *metric-value*] [**metric-type** *type-value*] {**level-1** | **level-1-2** | **level-2**} [**route-map** *map-name*]

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

Specify Router-Level Support

You can configure the router to act as a Level 1 (intra-area) router, as both a Level 1 router and a Level 2 (interarea) router, or as an interarea router only.

To specify router level support, perform the following task in router configuration mode:

Task	Command
Configure the level at which the router should operate.	**is-type** {**level-1** | **level-1-2** | **level-2-only**}

Configure IS-IS Authentication Passwords

You can assign passwords to areas and domains.

The area authentication password is inserted in Level 1 (station router level) LSPs, CSNPs, and Partial Sequence Number PDUs (PSNPs). The routing domain authentication password is inserted in Level 2 (the area router level) LSP, CSNP, and PSNPs.

To configure either area or domain authentication passwords, perform the following tasks in router configuration mode:

Task	Command
Configure the area authentication password.	**area-password** *password*
Configure the routing domain authentication password.	**domain-password** *password*

Summarize Address Ranges

You can create aggregate addresses that are represented in the routing table by a summary address. This process is called route summarization. One summary address can include multiple groups of addresses for a given level. Routes learned from other routing protocols also can be summarized. The metric used to advertise the summary is the smallest metric of all the more specific routes.

To create a summary of addresses for a given level, perform the following task in router configuration mode:

Task	Command
Create a summary of addresses for a given level.	**summary-address** *address mask* {**level-1** | **level-1-2** | **level-2**}

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 *AS path*), and a list of other *path attributes*. The BGP implementation supports all path attributes defined in RFC 1163 and 1267. 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 router address is used in the NEXT_HOP attribute, regardless of the AS of that third-party router. The router automatically calculates the value for this attribute.

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

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 and ISIS-IP.

See the "Using Route Maps with BGP" section 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 routers. 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 routers 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 router 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, 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 router and specify those networks within the router'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.
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 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 AS; external neighbors are in different ASs. 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 routers 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 router 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 *Router Products 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.
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.
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 router 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 router. A weight can be a number from 0 to 65535. Paths that the router 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.
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.
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 routers in your network had learned about the route through your IGP, your autonomous system could receive traffic that some routers 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 routers 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 routers in your autonomous system and there must be a full IBGP connectivity mesh between these routers. 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

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 router 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 router. 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 networks.	**network** *address* **weight** *weight*

Indicate Backdoor Routes

You can indicate which networks are reachable using a *backdoor* route that the border router 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 router 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 router that traffic should avoid, you can configure that router with a higher MULTI_EXIT_DISC METRIC. Doing this sets the MULTI_EXIT_DISC METRIC on all paths that the router advertises. Perform the following task in router configuration mode:

Task	Command
Set an 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 routers in the Defense Data Network (DDN) that the U.S. Department of Defense designates as *core routers*. EGP also was used extensively when attaching to the National Science Foundation Network (NSFnet) and other large backbone networks.

An exterior router uses EGP to advertise its knowledge of routes to networks within its autonomous system. It sends these advertisements to the core routers, which then readvertise their collected routing information to the exterior router. A neighbor or peer router is any router with which the router 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 routers poll their neighbors periodically to see if they are "alive."
- EGP routers send update messages containing information about the reachability of networks within their autonomous systems.

EGP Configuration Task List

To enable EGP routing on your router, 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 Routers
- 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.
Step 2	Specify the autonomous system that the router 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 router using EGP cannot dynamically determine its neighbor or peer routers. You must therefore provide a list of neighbor routers.

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 router sends hello messages to its peer. The poll time is 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 router (the third party) on the shared network is the appropriate router for some set of destinations.

To specify third-party routers 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 at the end of this chapter for an example of configuring third-party EGP support.

Configure Backup Routers

You might want to provide backup in the event of site failure by having a second router belonging to a different autonomous system act as a backup to the EGP router for your autonomous system. To differentiate between the primary and secondary EGP routers, the two routers 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 at the end of this chapter for an example of configuring backup routers.

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 router.

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.
Step 2	Allow a specific router 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 router informs the router about a specific network, and that router is the *best* path, the router will *not* inform the originating external router about that path). Our routers can also perform per-gateway split horizon on third-party updates.

Configure GDP

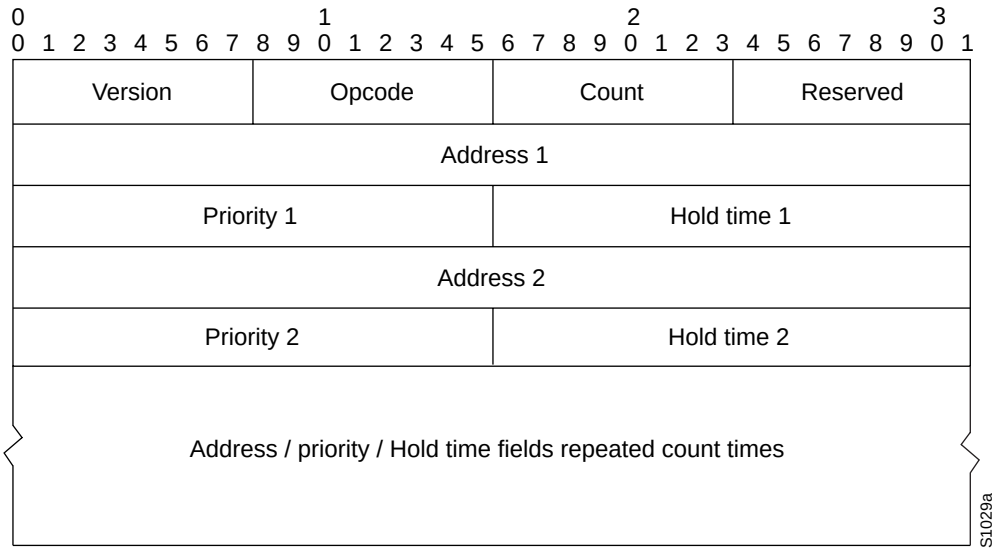
The Gateway Discovery Protocol (GDP), designed by Cisco to address customer needs, allows hosts to dynamically detect the arrival of new routers, as well as determine when a router goes down. You must have host software to take advantage of this protocol.

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 router is present and functioning. By listening for these report packets, a host can detect a vanishing or appearing router. If a host issues a query packet to the broadcast address, the routers each respond with a report sent to the host's IP address. On nonbroadcast media, routers 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 1-2 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 1-2 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 router 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 router 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 router appears or disappears. A more complex possibility is to update a host routing table based on the coming and going of routers. 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 router 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 routers. When operating as a client, router discovery packets are generated, and when operating as a host, router discovery packets are received.

The only required task for configuring IRDP routing on a specified interface is to enable IRDP processing ona ninterface. 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 router's IRDP preference level.	**ip irdp preference** *number*
Specify an IRDP address and preference to proxy-advertise.	**ip irdp address** *address* [*number*]

A router 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 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
- Enable or Disable Split Horizon

Use Variable-Length Subnet Masks

OSPF, static routes, and IS-IS 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 router 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 router 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 1-1. 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 router can no longer find a valid next hop for the address specified as the forwarding router's address in a static route, the static route is removed from the IP routing table.

Table 1-1 Default Administrative Distances

Route Source	Default Distance
Connected interface	0
Static route	1
External BGP	20
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
Internal BGP	200
Unknown	255

Specify Default Routes

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

Most dynamic interior routing protocols include a mechanism for causing a smart router to generate dynamic default information that is then passed along to other routers.

Specify a Default Network

If a router has a directly connected interface onto the specified default network, the dynamic routing protocols running on that router will generate or source a default route. In the case of RIP, it will advertise the pseudonetwork 0.0.0.0. In the case of IGRP, the network itself is advertised and flagged as an exterior route.

A router that is generating the default for a network also may 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 router.

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 router 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 router, 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 router.

If the router 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 router.

Redistribute Routing Information

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

You also can 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*

Task	Command
Match a next-hop router 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** *name unit...name unit*
Match the address specified by the specified advertised access lists.	**match ip route-source** *access-list-number...access-list-number*
Match the specifed route type.	**match route-type** {**local** | **internal** | **external** [**type-1** | **type-2**] | **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** *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**]

Task	Command
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:

- Suppress the sending of routing updates on a particular router interface. This is done to prevent other systems on an interface from learning about routes dynamically.
- Suppress networks from being advertised in routing updates. This is done to prevent other routers from learning a particular router's interpretation of one or more routes.
- Suppress networks listed in updates from being accepted and acted upon by a routing process. This is done to keep a router from using certain routes.
- Filter on the source of routing information. This is done 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. This is done 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.

Suppress Routing Updates through an Interface

To prevent other routers on a local network from learning about routes dynamically, you can keep routing update messages from being sent through a router interface. This feature applies to all IP-based routing protocols except BGP and EGP.

OSPF and IS-IS behaviors are 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 router interface. In IS-IS, the specified IP addresses are advertised without actually running IS-IS on those interfaces.

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

Task	Command
Suppress the sending of routing updates through the specified router interface.	**passive-interface** *interface*

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

Suppress Routes from Being Advertised in Routing Updates

To prevent other routers from learning one or more routes, you can suppress routes from being advertised in routing updates. You cannot specify an interface name in OSPF. When used for OSPF, this feature applies only to external routes.

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

Task	Command
Permit or deny routes from being advertised in routing updates depending upon the action listed in the access list.	**distribute-list** *access-list-number* **out** [*interface-name* | *routing-process*]

Suppress Routes Listed in Updates from Being Processed

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

Perform this task in router configuration mode:

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

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 router or a group of routers. In a large network, some routing protocols and some routers can be more reliable than others as sources of routing information. Also, when multiple routing processes are running in the same router 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 router to intelligently discriminate between sources of routing information. The router will always pick the route whose routing protocol has the lowest administrative distance.

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**]

For example, consider a router 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 router 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 router uses the RIP-derived information until the IGRP-derived information reappears.

Note You also can use administrative distance to rate the routing information from routers 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.

Assigning administrative distances is a problem unique to each network and is done in response to the greatest perceived threats to the network. Even when general guidelines exist, the network manager must ultimately determine a reasonable matrix of administrative distances for the network as a whole. Table 1-2 shows the default administrative distance for various sources of routing information.

Table 1-2 Default Administrative Distances

Route Source	Default Distance
Connected interface	0
Static route	1
External BGP	20
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
Internal BGP	200
Unknown	255

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 routers. 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 router 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 *Router Products Command Reference* publication.

You can also configure the *keepalive* interval, the frequency at which the router 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 one-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.

Enable or Disable Split Horizon

Normally, routers 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 router out any interface from which that information originated. This behavior usually optimizes communications among multiple routers, 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 routers 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 router 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 router 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 router's packets are taking through the network.

To display various router 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*]

Task	Command
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 IRDP values.	**show ip irdp**
Display general information about OSPF routing processes in a particular router.	**show ip ospf** [*process-id*]
Display lists of information related to the OSPF database for a specific router.	**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** [*interface-name*]
Display OSPF-neighbor information on a per-interface basis.	**show ip ospf neighbor** [*interface-name*] [*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 the IS-IS link state database.	**show isis database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]
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
- Configuring IS-IS as an IP Routing Protocol Example
- Static Routing Redistribution Example
- IGRP Redistribution Example
- RIP and IGRP Redistribution Example
- OSPF Routing and Route Redistribution Examples
- BGP Route Advertisement and Redistribution Examples
- Default Metric Values Redistribution Example
- Route-Map Examples
- IGRP Feasible Successor Relationship Example
- BGP Synchronization Example
- BGP Basic Neighbor Specification Examples
- BGP Aggregate Route Examples
- Third-Party EGP Support Example
- Backup EGP Router Example
- EGP Core Gateway Example
- Autonomous System within EGP Example
- Passive Interface Examples
- Administrative Distance Examples
- Split Horizon Examples

Variable-Length Subnet Masks Example

OSPF, static routes, and IS-IS 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.

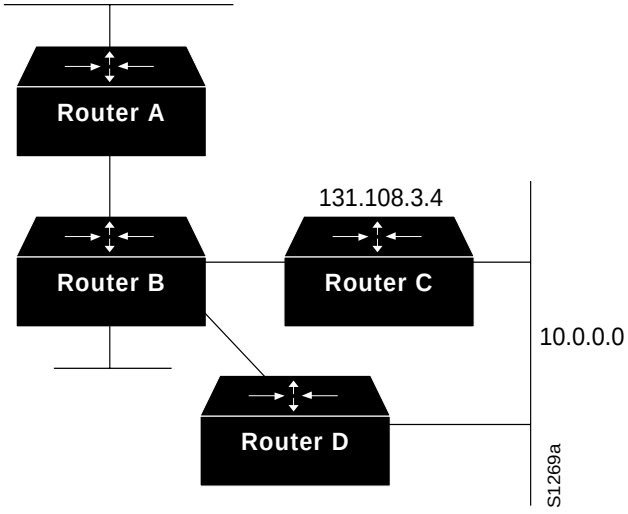
```
interface ethernet 0
ip address 131.107.1.1 255.255.255.0
! 8 bits of host address space reserved for ethernets

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

! Router is configured for OSPF and assigned AS 107
router ospf 107
! Specifies network directly connected to the router
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 1-3 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
```

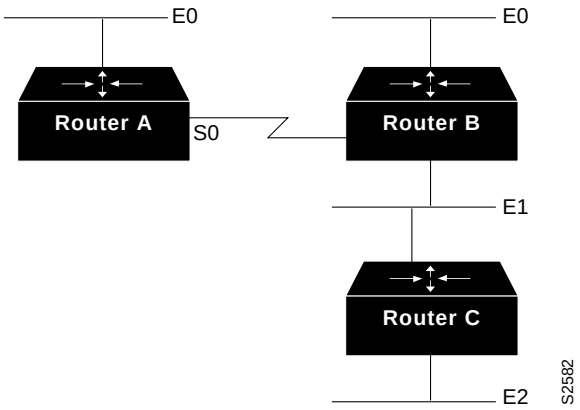
Figure 1-3 Overriding Static Routes



Configuring IS-IS as an IP Routing Protocol Example

The following example shows how you would configure three routers to run IS-IS as an IP routing protocol. Figure 1-4 illustrates the example configuration.

Figure 1-4 Illustration of IS-IS Routing



Configuration for Router A

```
router isis
net 49.0001.0000.0000.000a.00
interface e 0
ip router isis
interface s 0
ip router isis
```

Configuration for Router B

```
router isis
net 49.0001.0000.0000.000b.00
interface e 0
ip router isis
interface e 1
ip router isis
interface s 0
ip router isis
```

Configuration for Router C

```
router isis
net 49.0001.0000.0000.000c.00
interface e 1
ip router isis
interface e 2
ip router isis
```

Static Routing Redistribution Example

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 router 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
```

IGRP Redistribution Example

Each IGRP routing process can provide routing information to only one autonomous system; the router 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 router 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
```

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 routers on the regional network. The commands for the interconnecting router 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 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.

OSPF Routing and Route Redistribution Examples

OSPF typically requires coordination among many internal routers, area border routers, and autonomous system boundary routers. At a minimum, OSPF-based routers 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 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 Ethernet0
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 router sequentially evaluates the *address*/*wildcard-mask* pair for each interface. See the "IP Routing Protocols Commands" chapter of the Router Products Command Reference 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 interface Ethernet 0. Interface Ethernet 0 is attached to Area 10.9.50.0 only.

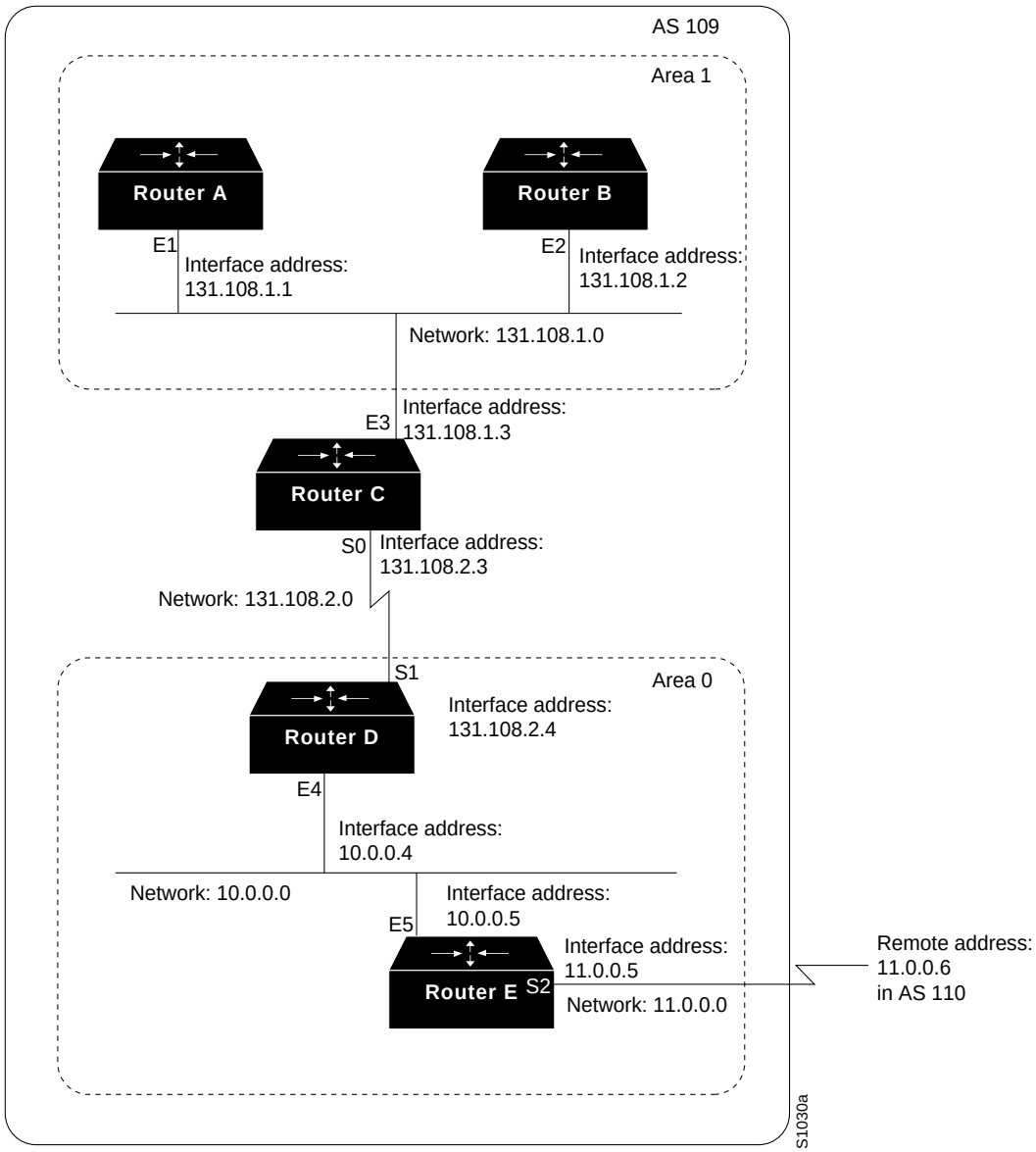
The second **network** command is evaluated next. For Area 2, the same process is then applied to all interfaces (except interface Ethernet 0). Assume that a match is determined for interface Ethernet 1. OSPF is then enabled for that interface and Ethernet 1 is attached to Area 2.

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 routers within a single OSPF autonomous system. Figure 1-5 provides a general network map that illustrates this example configuration.

Figure 1-5 Sample OSPF Autonomous System Network Map



In this configuration, five routers are configured in OSPF AS 109:

- Router A and Router B are both internal routers within Area 1.
- Router C is an OSPF area border router; note that for Router C, Area 1 is assigned to E3 and Area 0 is assigned to S0.
- Router D is 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).
- Router E is 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 routers 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 routers in Area 1 (Router A and Router B) when the area border router (Router 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 Router 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 Router 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 Router C - Area Border Router

```
interface Ethernet 3
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 Router 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 Router 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 an area border router. These tasks can be split into two general categories:

- Basic OSPF configuration
- Route redistribution

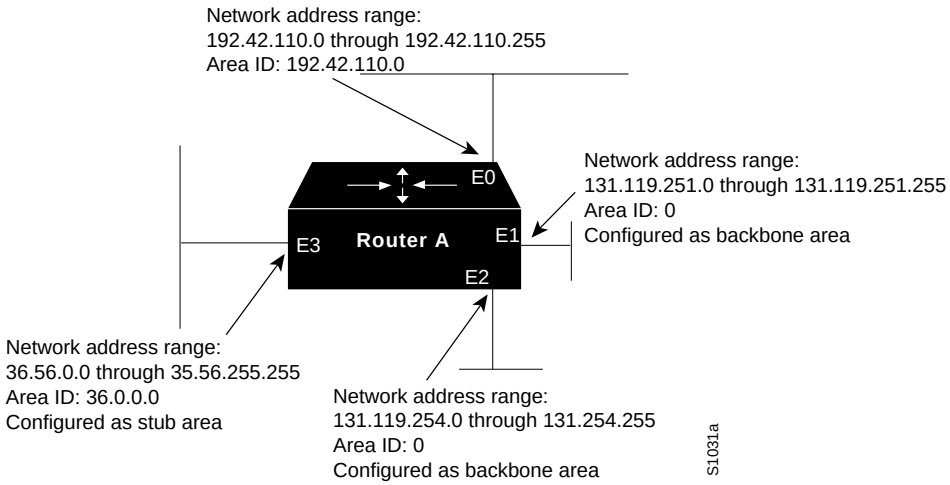
The specific tasks outlined in this configuration are detailed briefly in the following descriptions. Figure 1-6 illustrates the network address ranges and area assignments for the interfaces.

Figure 1-6 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 Ethernet0
ip address 192.42.110.201 255.255.255.0
ip ospf authentication-key abcdefgh
ip ospf cost 10
!
interface Ethernet1
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 Ethernet2
ip address 131.119.254.201 255.255.255.0
ip ospf authentication-key abcdefgh
ip ospf cost 10
!
interface Ethernet3
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 between 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 192.5.10.15 remote-as 1
! Same AS; therefore internal neighbor
neighbor 192.5.10.24 remote-as 3
! A second External neighbor
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
```

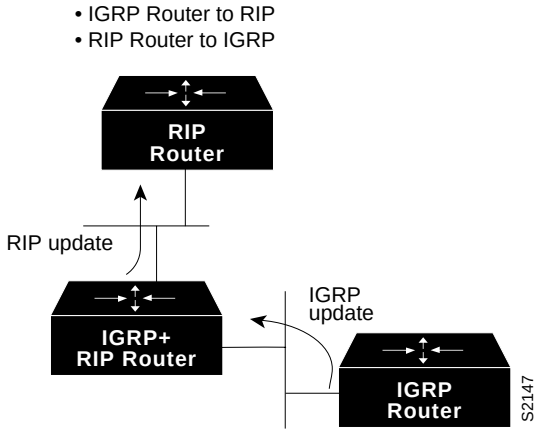
Default Metric Values Redistribution Example

The following example shows a router 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
```

Figure 1-7 shows this type of redistribution.

Figure 1-7 Assigning Metrics for Redistribution



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 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 routers on interface serial 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
```

The following example redistributes two types of routes into the integrated IS-IS routing table (supporting both IP and CLNS). The first are OSPF external IP routes with tag 5; these are inserted into Level 2 IS-IS LSPs with a metric of 5. The second are ISO-IGRP derived CLNS prefix routes that match CLNS access list 2000. These will be redistributed into IS-IS as Level 2 LSPs with a metric of 30.

```
router isis
redistribute ospf 109 route-map 2
redistribute iso-igrp nsfnet route-map 3
!
route-map 2 permit
match route-type external
match tag 5
set metric 5
set level level-2
!
route-map 3 permit
match address 2000
set metric 30
```

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
```

Given the following configuration, a RIP learned route for network 160.89.0.0 and an ISO-IGRP learned route with prefix 49.0001.0002 will be redistributed into an IS-IS Level 2 LSP with a metric of 5:

```
router isis
redistribute rip route-map 1
redistribute iso-igrp remote route-map 1
!
route-map 1 permit
match ip address 1
match clns address 2
set metric 5
set level level-2
!
access-list 1 permit 160.89.0.0 0.0.255.255
clns filter-set 2 permit 49.0001.0002...
```

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 to not 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 router will not accept the route. However, you can configure the router 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
```

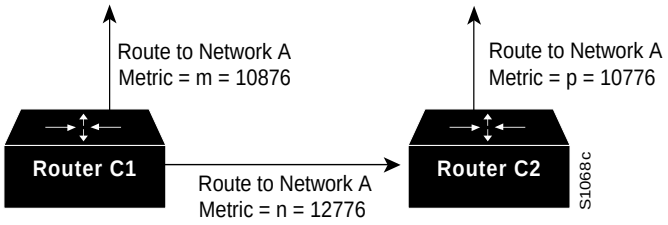
The following example shows how you can use route maps in a reverse operation to set the route tag (as defined by the BGP/OSPF interaction document, RFC 1403) when exporting routes from BGP into the main IP routing table:

```
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 .*
```

IGRP Feasible Successor Relationship Example

In Figure 1-8, 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 1-8 Assigning Metrics for IGRP Path Feasibility



The feasibility test would work as follows:

Assume that Router C1 already has a route to Network A with metric m and has just received an update about Network A from C2. The best metric at C2 is p. The metric that C1 would use through C2 is n.

If both of the following two conditions are met, the route to network A through C2 will be included in C1's routing table:

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

The configuration for Router C1 would be as follows:

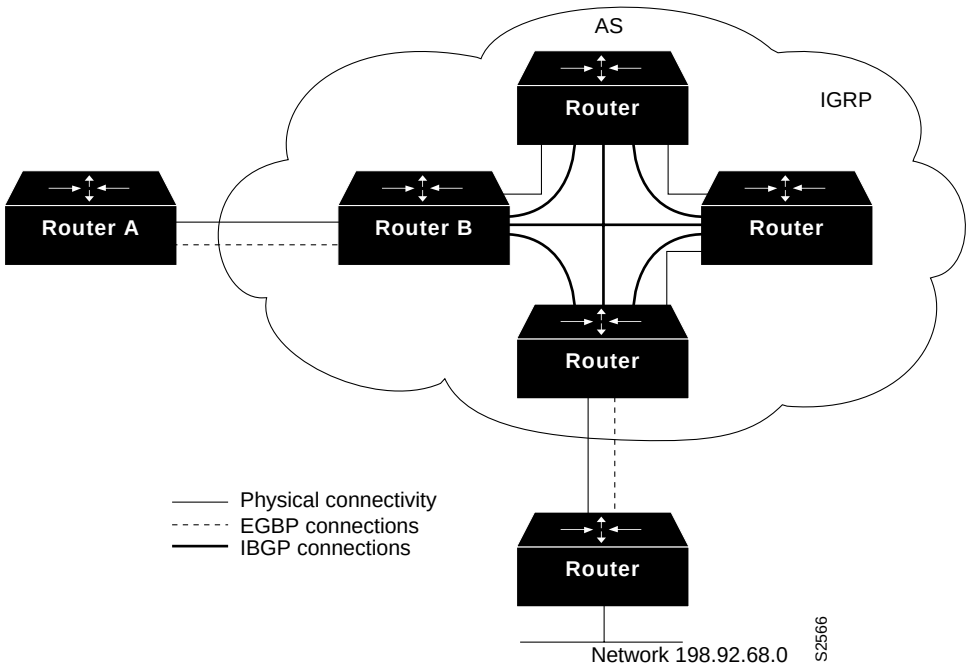
```
router igrp 109
variance 10
```

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 1-9, with synchronization on, Router B will not advertise network 10.0.0.0 to Router A until an IGRP route for network 10.0.0.0 exists. If you specify the **no synchronization** router configuration command, Router B will advertise network 10.0.0.0 as soon as possible. However, since routing information still must be sent to interior peers, you must configure a full internal BGP mesh.

Figure 1-9 Illustration of Synchronization



BGP Basic Neighbor Specification Examples

The following example specifies that the router at the address 131.108.1.2 is a neighbor in AS number 109.

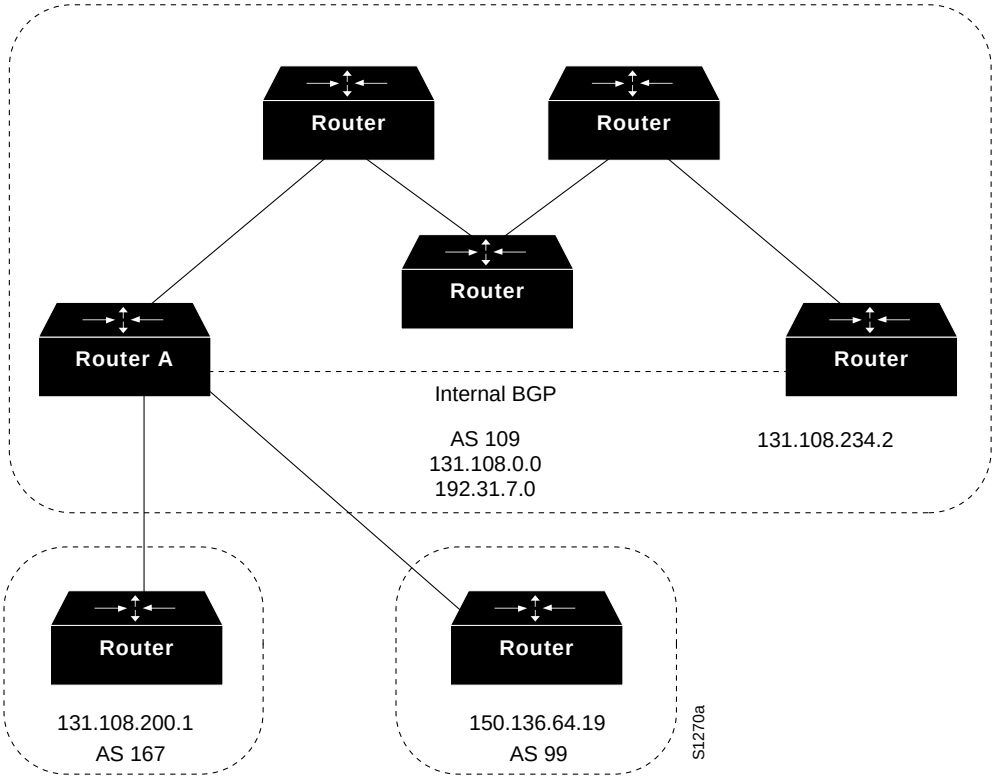
```
neighbor 131.108.1.2 remote-as 109
```

In the following example, a BGP router is assigned to autonomous system 109, and two networks are listed as originating in the AS. Then the addresses of three remote routers (and their autonomous systems) are listed. The router being configured will share information about networks 131.108.0.0 and 192.31.7.0 with the neighbor routers. The first router 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 AS 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 1-10, Router A is being configured. The internal BGP neighbor is not directly linked to Router A. External neighbors (in AS 167 and AS99) must be linked directly to Router A.

Figure 1-10 Assigning Internal and External BGP Neighbors



Using Access Lists to Specify Neighbors

In the following example, the router is configured to allow connections from any router that has an IP address in access list 1; that is, any router with a 192.31.7.x address. Neighbors not explicitly specified as neighbors can connect to the router, but the router 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 router 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 AS 110. The configuration specifies that two routers, 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 routers. 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 Router Example

The following example configuration illustrates a router that is in autonomous system 110 communicating with an EGP neighbor in autnomous 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 1-11 illustrates an environment with three routers (designated C1, C2, and C3) attached to a common X.25 network. The routers are intended to route information using EGP. With the following configuration (on the router designated Core), C1, C2, and C3 cannot route traffic directly to each other via the X.25 network:

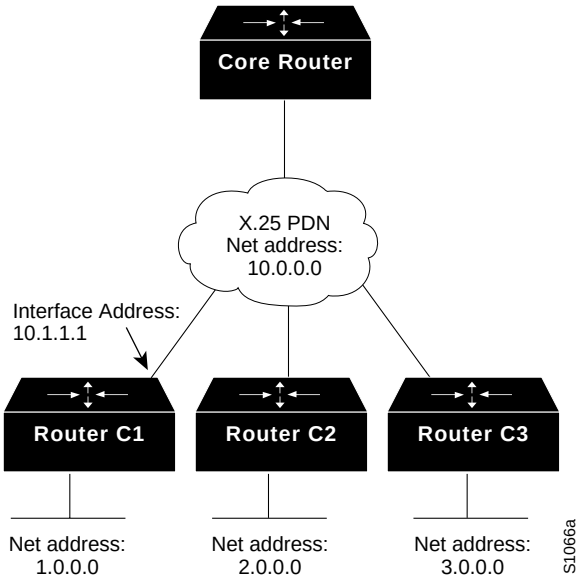
```
access-list 1 permit 10.0.0.0 0.255.255.255
! global access list assignment
router egp 0
neighbor any 1
```

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

Third-party advertisements allow traffic to bypass the Core router and go directly to the router 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 1-11 Core EGP Third-Party Update Configuration Example

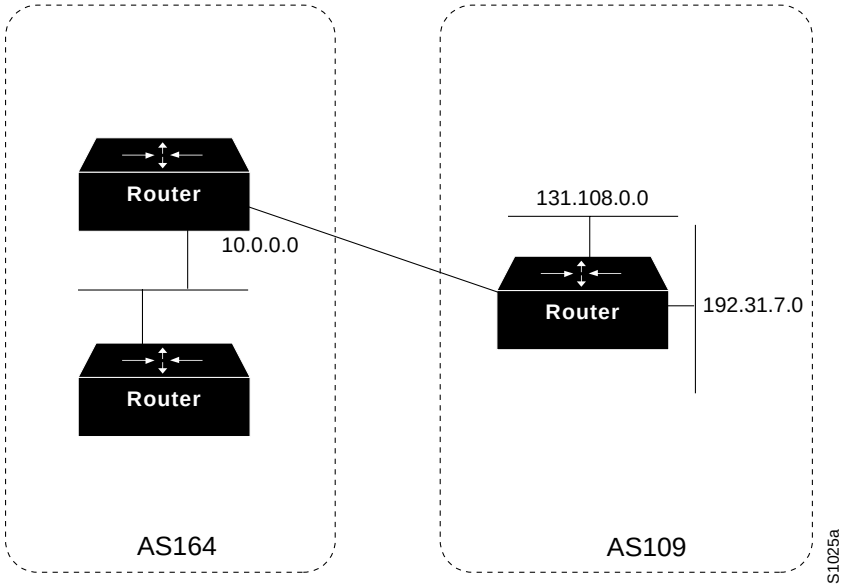


Autonomous System within EGP Example

The following example illustrates a typical configuration for an EGP router process. The router is in autonomous system 109 and is peering with routers in autonomous system 164, as shown in Figure 1-12. It will advertise the networks 131.108.0.0 and 192.31.7.0 to the router in autonomous system 164, 10.2.0.2. The information sent and received from peer routers can be filtered in various ways, including blocking information from certain routers 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 1-12 Router in AS 164 Peers with Router in AS 109

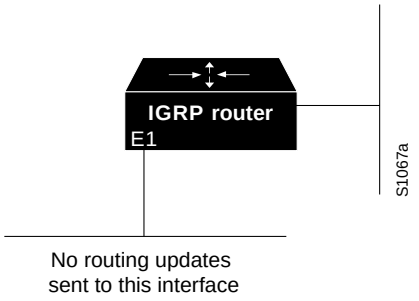


Passive Interface Examples

The following example sends IGRP updates to all interfaces on network 131.108.0.0 except interface Ethernet 1. Figure 1-13 shows this configuration.

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

Figure 1-13 Filtering IGRP Updates



As in the first example, IGRP updates are sent to all interfaces on network 131.108.0.0 except interface Ethernet 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 router will not be able to discover any neighbors, and none of the OSPF neighbors will be able to see the router 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 AS number 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 router to ignore all routing updates from routers for which an explicit distance has not been set. The second **distance** command sets the administrative distance to 90 for all routers on the Class C network 192.31.7.0. The third **distance** command sets the administrative distance to 120 for the router 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 router with the address 192.31.7.18 an administrative distance of 100, and all other routers 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 routers on subnet 192.31.7.0 are assigned an administrative distance of 200, including the router 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
```

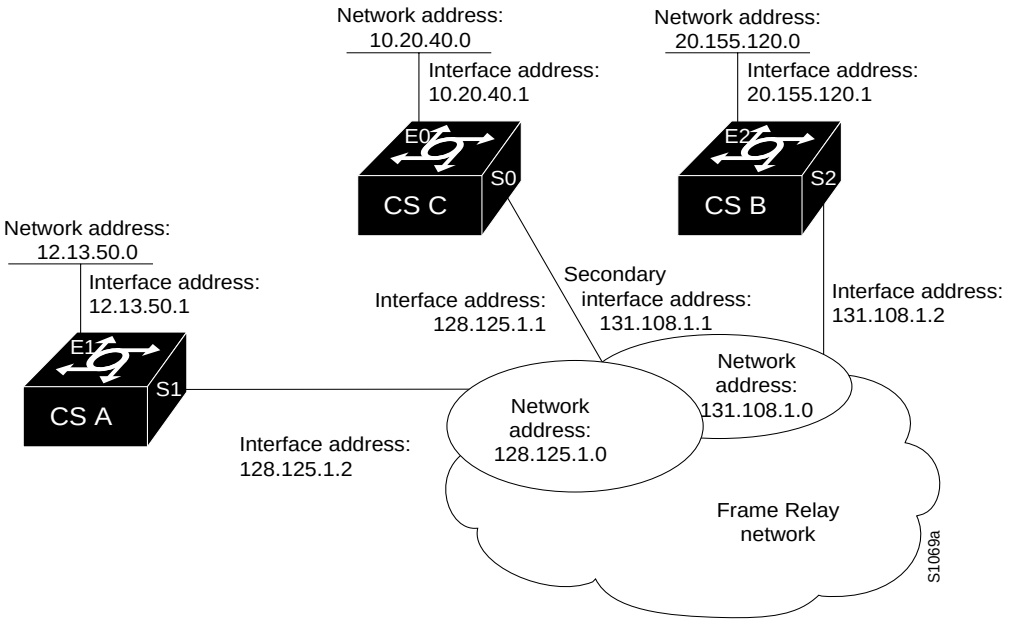
Example 2

In the next example, Figure 1-14 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 Router C (connected to Frame Relay network). In this example, the serial interface on Router C accommodates one of the subnets via the assignment of a secondary IP address.

The Ethernet interfaces for Router A, Router B, and Router 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 router that follow Figure 1-14 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 Router C, interface S0. If split horizon were enabled on serial interface S0, it would not advertise a route back into the Frame Relay network for either of these networks.

Figure 1-14 Disabled Split Horizon Example for Frame Relay Network



Configuration for Router A

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

Configuration for Router B

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

Configuration for Router C

```
interface ethernet 0
ip address 10.20.40.1
!
interface serial 0
ip address 128.124.1.1
ip address 131.108.1.1 secondary
encapsulation frame-relay
```