

# Configuring IP Routing Protocols

---

This chapter describes how to configure the various IP routing protocols. For a complete description of the commands listed in this chapter, refer to the *Communication Server Command Reference* publication. For information on configuring the IP protocol, refer to the IP configuration 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

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 communication servers as *gateways*, so the word *gateway* appears often as part of routing protocol names. This is unfortunate because a communication server is usually defined as a layer 3 internetworking device, whereas a protocol translation gateway is usually 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 have a list of networks specified before routing activities can begin. A routing process listens to updates from other communication servers on these networks and broadcasts its own routing information on those same networks. The interior routing protocols supported are as follows:

- IGRP
- OSPF
- RIP

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

The supported exterior routing protocols are as follows:

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

## Router Discovery Protocols

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

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

## Multiple Routing Protocols

We support the operation of multiple routing protocols in a single communication server in order to connect networks that use different routing protocols. It is possible, for example, to run RIP on one subnetted network, IGRP on another subnetted network, and to exchange routing information between them in a controlled fashion. The routing protocols available 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 to enable to filter the exchange of routing information.

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

- Any number of IGRP routing processes up to 30
- Any number of OSPF routing processes up to 30
- One RIP routing process
- One BGP routing process
- Any number of EGP routing processes up to 30

## 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 following tasks to configure IP routing protocols:

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

Each task is described in a following section. See the configuration examples at the end of this chapter, in the section “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
- 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 routing protocol that we support. 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

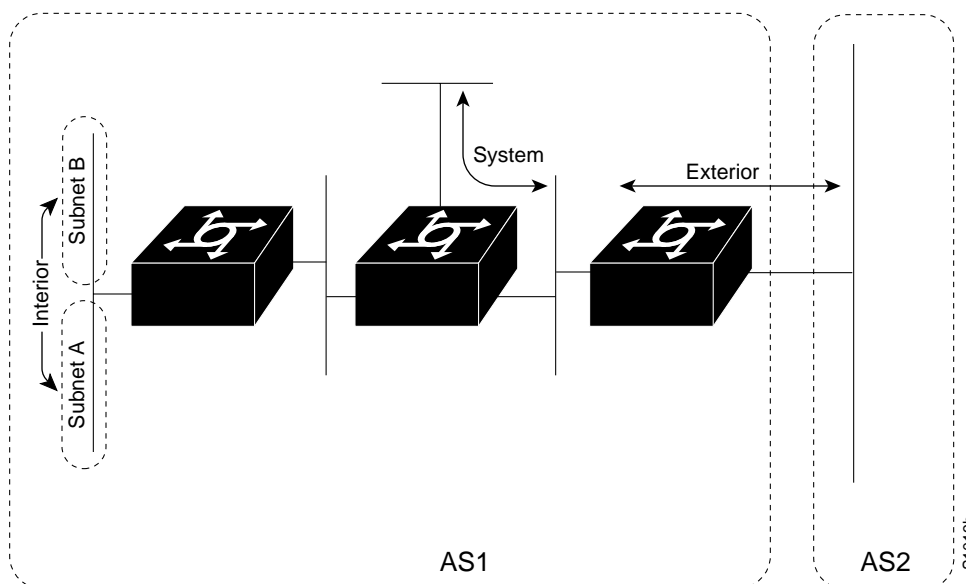
The Interior Gateway Routing Protocol (IGRP) is a dynamic distance-vector routing protocol designed by Cisco Systems in the mid-1980s for routing in an autonomous system containing large, arbitrarily complex networks with diverse bandwidth and delay characteristics.

## Cisco's IGRP Implementation

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

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

Figure 1-1 Interior, System, and Exterior Routes



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

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

## IGRP Updates

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

IGRP uses *flash update* and *poison reverse updates* to speed up the convergence of the routing algorithm. Flash update is the sending of an update sooner than the standard periodic update interval of notifying other communication servers of a metric change. Poison reverse updates are intended to defeat larger routing loops caused by increases in routing metrics. The poison reverse updates are sent to remove the 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, 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

Each task is described in a following section.

## Create the IGRP Routing Process

The tasks you must perform to create the IGRP routing process are summarized in the following table. Perform them in the order listed.

Task	Command
Step 1 Enter global configuration mode.	See Table 1-2.
Step 2 Enable an IGRP routing process.	<b>router igrp</b> <i>autonomous-system</i>
Step 3 Associate networks with an IGRP routing process.	<b>network</b> <i>network-number</i>

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

## 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 respond to IGRP routes and another part will respond to RIP routes. This will result in routing loops. Even though these loops do not exist for very long, the TTL will quickly reach zero and ICMP will send a "TTL exceeded" message. This message will cause most applications to stop attempting network connections.

## Allow Point-to-Point Updates for IGRP

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

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

Task	Command
Define a neighboring communication server with which to exchange point-to-point routing information.	<b>neighbor</b> <i>ip-address</i>

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 communication server in the path is *closer* to the destination (has a lower metric value) than the current communication server and if the metric for the entire alternate path is *within* the variance. Only paths that are feasible can be used for load balancing and included in the routing table. These conditions limit the number of cases in which load balancing can occur, but ensure that the dynamics of the network will remain stable.

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

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

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

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

Task	Command
Define the variance associated with a particular path.	<b>variance</b> <i>multiplier</i>

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

---

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

---

## Control Traffic Distribution

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

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

Task	Command
Distribute traffic proportionately to the ratios of metrics, or by the minimum-cost route.	<b>traffic-share { <i>balanced</i>   <i>min</i> }</b>

## 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 of 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 that all metric adjustments are made 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 this task with guidance from an experienced system designer.

Task	Command
Adjust the IGRP metric.	<b>metric weights <i>TOS K1 K2 K3 K4 K5</i></b>

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 (Ethernet, and serial lines running from 9600 bps to T1 rates), the route with the lowest metric reflects the most desirable path to a destination.

## Disable Holddown

When a communication server learns a network is at a greater distance than was previously known, or is now 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 other communication server other than the one that originally advertised the network's new metric will be ignored. This mechanism is often used to help avoid routing loops in the network, but has the effect of increasing the topology convergence time. Since release 8.2(5) of our software, IGRP has had other mechanisms available to avoid these routing loops, so the holddown period can be disabled.

To disable holddowns with IGRP, perform the following task in router configuration mode. All communication servers running IGRP on your network must be consistent in their use of this command:

Task	Command
Disable the IGRP holddown period.	<b>no metric holddown</b>

## Enforce a Maximum Network Diameter

The communication server enforces a maximum diameter to the IGRP network. Routes whose hopcounts 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.	<b>metric maximum-hops</b> <i>hops</i>

## 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 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, and RIP. OSPF routes also can be exported into IGRP, and RIP. At the interdomain level, OSPF can import routes learned via EGP and BGP. OSPF routes can be exported into EGP and BGP.
- Authentication—Authentication among neighboring communication servers within an area is supported.
- Routing interface parameters—Configurable parameters supported include interface output cost, retransmission interval, interface transmit delay, communication server priority, communication server “dead” and Hello interval, and authentication key.
- Virtual links—Virtual links are supported.

---

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

---



## OSPF Configuration Task List

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

Configuring an OSPF routing process involves the general tasks listed below. 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
- Specify route redistribution; see the task “Redistribute Routing Information” later in this chapter for information on how to configure route redistribution.

For all other tasks, see the following sections.

### 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 these tasks as shown in the following table.

Task	Command
Step 1 Enter global configuration mode.	See Table 2-2.
Step 2 Enable OSPF routing.	<b>router ospf</b> <i>ospf-process-id</i>
Step 3 Define an interface on which OSPF runs and define the area ID for that interface.	<b>network</b> <i>address wildcard-mask</i> <b>area</b> <i>area-id</i>

### Configure OSPF Interface Parameters

Our OSPF implementation allows you to alter certain interface-specific OSPF parameters, as needed. You can:

- Specify the OSPF path cost
- Set the link state retransmission interval
- Set the transmission time for link state updates
- Set communication server priority

- Set the advertised Hello interval
- Set the communication server dead interval
- Specify the OSPF authentication key

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

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

## Configure OSPF over Different Physical Networks

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

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

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

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

## Configure OSPF Network Type

You have the choice of configuring your OSPF network type to either broadcast or nonbroadcast multiaccess, regardless of the default media type. Using this feature, you can configure broadcast networks as nonbroadcast multiaccess networks when, for example, you have communication servers in your network that do not support multicast addressing. You also can configure nonbroadcast multiaccess networks, such as X.25, Frame Relay, and SMDS, as broadcast networks. This feature saves you from having to configure neighbors, as described in 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.	<b>ip ospf network</b> [broadcast   non-broadcast]

## Configure OSPF for Nonbroadcast Networks

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

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

To configure communication servers that interconnect to nonbroadcast networks, perform the following task in router configuration mode. You can specify the following neighbor parameters, as required:

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

Task	Command
Configure communication servers interconnecting to nonbroadcast networks.	<b>neighbor</b> <i>ip-address</i> [priority <i>number</i> ] [poll-interval <i>number-of-seconds</i> ]

## 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 communication server into the stub area for destinations outside the autonomous system (AS).

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

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

## 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 communication server. In OSPF, an area border communication server 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 communication server to advertise a summary route which 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.	<b>area area-id range address mask</b>

## 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 endpoints of a virtual link are area border communication servers. The virtual link must be configured in both communication servers. The configuration information in each communication server consists of the other virtual endpoint (the other area border communication server), and the nonbackbone area the two communication servers have in common (called the *transit area*). Note that virtual links cannot be configured through stub areas.

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

Task	Command
Establish a virtual link.	<b>area area-id virtual-link communication server-id [hello-interval number-of-seconds] [retransmit-interval number-of-seconds] [transmit-delay number-of-seconds] [dead-interval number-of-seconds] [authentication-key 8-bytes-of-password]</b>

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

## Generate a Default Route

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

To force the AS boundary communication server to generate a default route, perform the following task in router configuration mode:

Task	Command
Force the AS boundary communication server to generate a default route into the OSPF routing domain.	<b>default-information originate</b> [always] [metric <i>metric-value</i> ] [metric-type <i>type-value</i> ] [route-map <i>map-name</i> ]

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

## Configure Lookup of DNS Names

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

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

Task	Command
Configure DNS name lookup.	<b>ip ospf-name-lookup</b>

## Force the Router ID Choice with a Loopback Interface

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

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

To configure an IP address on a loopback interface, perform the following tasks:

Task	Command
<b>Step 1</b> Enter global configuration mode.	See Table 2-2.
<b>Step 2</b> Create a loopback interface.	<b>interface loopback 0</b>
<b>Step 3</b> Assign an IP address to this interface.	<b>ip address</b> <i>IP-address network-mask</i>

## 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.	<b>passive-interface</b> <i>interface</i>

## 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.

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

The measure, or metric, that RIP uses to rate the value of different routes is the *hop count*. The hop count is the number of communication servers that can be traversed in a route. A directly connected network has a metric of zero; an unreachable network has a metric of 16. This small range of metrics makes RIP unsuitable as a routing protocol for large networks. If the communication server has a default network path, RIP advertises a route that links the communication server to the pseudo-network 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 communication servers will advertise the default network if a default was learned by RIP, or if the communication server has a gateway of last resort and RIP is configured with a default metric.

RIP sends updates to the interfaces in the specified network(s). 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 on in this chapter. RIP is documented in RFC 1058.

To configure RIP, perform the following tasks in the order shown:

Task	Command
Step 1 Enter global configuration mode.	See Table 2-2.
Step 2 Enable a RIP routing process.	<b>router rip</b>
Step 3 Associate a network with a RIP routing process.	<b>network</b> <i>network-number</i>

## Allow Point-to-Point Updates for RIP

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

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

Task	Command
Define a neighboring communication server with which to exchange point-to-point routing information.	<b>neighbor</b> <i>ip-address</i>

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 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 (ASs).

### Cisco’s BGP Implementation

In BGP, each route consists of a network number, a list of ASs 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. This section describes some details of that implementation.

You can configure the value for the INTER\_AS metric attribute. The same metric value will be sent with all BGP updates originating from the communication server. The default is to not include an INTER\_AS METRIC in BGP updates.

A third-party next hop communication server address is used in the NEXT\_HOP attribute, regardless of the AS of that third-party communication server. The communication server automatically calculates the value for this attribute.

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

## How BGP Selects Paths

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

- If IGP synchronization is enabled, internal paths which are not synchronized via an IGP are never considered.
- BGP administrative weights are considered. Larger weights are preferred.
- If the routes have the same administrative weight, prefer the route that this communication server originated.
- If no route was originated, prefer the shorter AS path.
- If the AS paths are of the same length, prefer external paths over internal paths.
- If all paths are external, prefer the lowest origin code (IGP <EGP <INCOMPLETE).
- If origin codes are the same, prefer the path with the lowest INTER\_AS 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 peer's BGP communication server ID with the largest IP address value.

## BGP Configuration Task List

Perform the following mandatory tasks to configure BGP on your communication server:

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

The following BGP tasks are optional:

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

Each task is described in a following section.



## Enable BGP Routing

To enable BGP routing, establish a BGP routing process in the communication server and specify those networks within the communication server's autonomous system (AS) to be advertised. The following table shows how to do this. There is a limit of 200 networks that can be advertised from one AS.

Task	Command
Step 1 Enter global configuration mode.	See Table 2-2.
Step 2 Enable a BGP routing process.	<b>router bgp</b> <i>autonomous-system</i>
Step 3 Flag a network as local to this AS.	<b>network</b> <i>network-number</i>

**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** subcommand to determine where to send updates.

## Configure BGP Neighbors

Like other exterior gateway protocols (EGPs), BGP must completely understand the relationships it is to have 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 AS. To configure BGP neighbors, perform the following task in router configuration mode:

Task	Command
Specify a BGP neighbor.	<b>neighbor</b> <i>ip-address</i> <b>remote-as</b> <i>number</i>

You also can configure neighbor templates which use a word argument rather than an IP address to configure BGP neighbors. This is an advanced feature requiring a well-thought 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 peerings by configuring a neighbor template.	<b>neighbor</b> <i>template-name</i> <b>neighbor-list</b> <i>access-list-number</i>
Treat neighbors that have been accepted by a template, as if they were configured by hand.	<b>neighbor</b> <i>template-name</i> <b>configure-neighbors</b>

## 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.	<b>clear ip bgp</b> <i>address</i>
Reset all BGP connections.	<b>clear ip bgp</b> *

## Configure BGP Route Filtering by Neighbor

If you want to restrict the routing information that the communication server learns or advertises, you can filter BGP routing updates to and from particular neighbors. To do this, define an access list and apply it to updates to and from particular neighbors. 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.	<b>neighbor</b> <i>ip-address</i> <b>distribute-list</b> <i>access-list-number</i> { <b>in</b>   <b>out</b> }

This command is used to filter by network numbers, not AS paths.

## Configure BGP Path Filtering

In addition to filtering routing updates to and from particular neighbors, you can specify an access list filter on both incoming and outbound 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 for more information on forming regular expressions.

The following two tables show the tasks to configure BGP path filtering.

Perform the following task in global configuration mode:

Task	Command
Define a BGP-related access list.	<b>ip as-path access-list</b> <i>access-list-number</i> { <b>permit</b>   <b>deny</b> } <i>as-regular-expression</i>

Perform the following task in router configuration mode:

Task	Command
Establish a BGP filter.	<b>neighbor</b> <i>ip-address</i> <b>filter-list</b> <i>access-list-number</i> { <b>in</b>   <b>out</b> }

## Disable Next-hop Processing on BGP Updates

You can configure the communication server to disable next-hop processing on BGP updates. This is useful in non-meshed networks such as Frame Relay or X.25 where BGP neighbors may 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.	<b>neighbor ip-address next-hop-self</b>

## 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. A weight can be a number from -65536 to 65535. Paths that the communication server originates have weight 32768 by default, other paths have weight zero. If you have particular neighbors that you wish 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.	<b>neighbor ip-address weight weight</b>

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

Task	Command
Enter global configuration mode.	See Table 2-2.
Define a BGP-related access list.	<b>ip as-path access-list access-list-number { permit   deny } as-regular-expression</b>
Enter router configuration mode.	See Table 2-2.
Establish a BGP filter. This applies to routes learned from the neighbor, not routes sent to the neighbor.	<b>neighbor ip-address filter-list access-list-number weight weight</b>

## Configure BGP Interactions with IGP

If your autonomous system (AS) will be passing traffic through it from another AS to a third AS, then it is very important that your AS be consistent about the routes that it advertises. For example, if your BGP were to advertise a route before all communication servers in your network had learned about the route through your IGP, your AS could receive traffic which some communication servers cannot yet route. To prevent this from happening, BGP must wait until the IGP has propagated routing information across your AS. 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 AS through your AS, or if all communication servers in your AS 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. To disable synchronization, perform the following task in router configuration mode:

Task	Command
Disable synchronization between BGP and an IGP.	<b>no synchronization</b>

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 AS.

In most circumstances, you also will not want to redistribute your IGP into BGP. Just list the networks in your AS with **network** router configuration commands and your networks will be advertised. Networks which 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 Miscellaneous BGP Parameters

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

### Configure Neighbor Options

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

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

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

Task	Command
Specify an access list of BGP neighbors.	<b>neighbor any</b> [ <i>access-list-number</i> ]
Specify the BGP version to use when communicating with a neighbor.	<b>neighbor ip-address version value</b>
Allow BGP sessions even when the outbound interface is down.	<b>neighbor ip-address update-source interface</b>
Allow BGP sessions even when the neighbor is not on a directly connected segment.	<b>neighbor ip-address ebgp-multihop</b>

This is especially useful when there is more than one internal path between routers and you want to be sure that the IBGP peering stays up all the time if some links in your backbone fail.

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

## Indicate Backdoor Routes

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

Task	Command
Indicate reachable networks through backdoor routes.	<b>network address backdoor</b>

## 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 which are part of this AS 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.	<b>distance bgp external-distance internal-distance local-distance</b>

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

## Adjust BGP Timers

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

Task	Command
Adjust BGP timers.	<b>timers bgp</b> <i>keepalive holdtime</i>

## Configure the INTER\_AS METRIC

BGP uses the INTER\_AS METRIC as a hint to external neighbors about preferred paths. If you have a communication server that traffic should avoid, you can configure that communication server with a higher INTER\_AS METRIC. Doing this sets the INTER\_AS METRIC on all paths that the communication server advertises. Perform the following task in router configuration mode:

Task	Command
Set an INTER_AS METRIC.	<b>default-metric</b> <i>metric</i>

INTER\_AS METRIC is only sent to external peers, not to internal peers.

## Configure EGP

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

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

## Cisco's EGP Implementation

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

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

## EGP Configuration Task List

The following tasks enable EGP routing on your communication server. The first two tasks are mandatory; the others 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)

Each task is described in a following section.

### Enable EGP Routing

To enable EGP routing, you must specify an AS 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
<b>Step 1</b> Enter global configuration mode.	See Table 2-2.
<b>Step 2</b> Specify the autonomous system that the communication server resides in for EGP.	<b>autonomous-system</b> <i>local-AS</i>
<b>Step 3</b> Enable an EGP routing process.	<b>router egp</b> <i>remote-AS</i>
<b>Step 4</b> Specify a network to be advertised to the EGP peers of an EGP routing process.	<b>network</b> <i>network-number</i>

---

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

---

## Configure EGP Neighbor Relationships

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

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

Task	Command
Specify an EGP neighbor.	<b>neighbor</b> <i>ip-address</i>

## Adjust EGP Timers

The EGP timers consist of a hello timer and a poll time interval timer. The hello timer determines the frequency in seconds with which the communication server sends hello messages to its peer. The poll time 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.	<b>timers egp</b> <i>hello polltime</i>

## Configure Third-Party EGP Support

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

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

Task	Command
Specify a third-party through which certain destinations can be achieved.	<b>neighbor</b> <i>ip-address</i> <b>third-party</b> <i>third-party-ip-address</i> [ <b>internal</b>   <b>external</b> ]

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 communication server belonging to a different AS act as a backup to the EGP communication server for your AS. To differentiate between the primary and secondary EGP communication servers, the two communication servers will advertise network routes with differing EGP distances or metrics. A network with a low metric is generally favored over a network with a high metric.

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

See the “IP Routing Protocol Configuration Examples” section at the end of this chapter for an example of configuring backup communication servers.



## Configure Default Routes

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

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

Task	Command
Configure EGP to generate a default route.	<b>default-information originate</b>

## Define a Central Routing Information Manager (Core Gateway)

Normally, an EGP process expects to communicate with neighbors from a single AS. Because all neighbors are in the same AS, 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 ASs, and all have inconsistent information. In this case, the EGP process distributes routes from one neighbor to all others (but not back to the originator). This allows the EGP process to be a central clearinghouse for information with a single, central manager of routing information (sometimes called a *core gateway*). To this end, one core gateway process can be configured for each communication server.

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

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

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

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

---

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

---

## Configure GDP

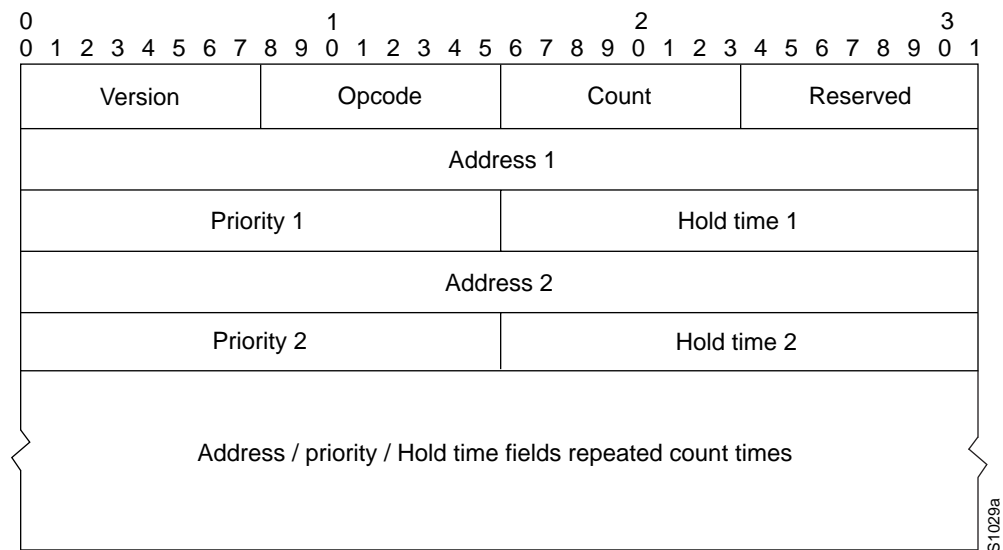
The Gateway Discovery Protocol (GDP), designed by Cisco to address customer needs, allows hosts to dynamically detect the arrival of new communication servers, as well as determine when a communication server goes down. You must have host software to take advantage of this protocol.

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

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

Figure 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 communication server on the local network segment. There are no other restrictions on this address. If a host encounters an address that it believes is not on its local network segment, then the host should ignore that address.
- **Priority**—16-bit fields that indicate the relative quality of the associated address. The numerically larger the value in the priority field, the better the address should be considered.
- **Hold Time**—16-bit fields. On broadcast media, the number of seconds the associated address should be used as a communication server without hearing further report messages regarding that address. On nonbroadcast media such as X.25, this is the number of seconds the requester should wait before sending another query message.

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

To enable GDP routing, perform the following tasks in interface configuration mode:

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

## Configure IRDP

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

The only required task for configuring IRDP routing on a specified interface is listed in the following table. Perform the following task in interface configuration mode:

Task	Command
Enable IRDP processing on an interface.	<b>ip irdp</b>

## Configure Miscellaneous IRDP Parameters

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

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

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

## Configure Routing Protocol-Independent Features

Previous sections addressed configurations of specific routing protocols. This section describes the following protocol-independent tasks:

- Use Variable-Length Subnet Masks
- Configure Static Routes
- Specify Default Routes
- Redistribute Routing Information
- Filter Routing Information
- Adjust Timers
- Enable Split Horizon

Each task is described in a following section.

### Use Variable-Length Subnet Masks

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

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

---

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

---

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

## Configure Static Routes

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

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

Task	Command
Establish a static route.	<b>ip route</b> <i>network</i> [ <i>mask</i> ] { <i>address</i>   <i>interface</i> } [ <i>distance</i> ]

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

The communication server remembers static routes until you remove them (using the **no** form of the **ip route** global configuration command). However, you can override static routes with dynamic routing information through prudent assignment of administrative distance values. Each dynamic routing protocol has a default administrative distance, as listed in Table 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 using RIP, IGRP, and other dynamic routing protocols, regardless of whether **redistribute static** commands were specified for those routing protocols. These static routes will be advertised because static routes that point to an interface are considered to be connected in the routing table and hence lose their static nature. However, if you define a static route to an interface that is not in one of the networks defined in a **network** command, no dynamic routing protocols will advertise the route unless a **redistribute static** command is specified for these protocols.

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

**Table 1-1** Default Administrative Distances

Route Source	Default Distance
Connected interface	0
Static route	1
External BGP	20
IGRP	100
OSPF	110

Route Source	Default Distance
RIP	120
EGP	140
Internal BGP	200
Unknown	255

## Specify Default Routes

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

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

## Specify a Default Network

If a communication server has a directly connected interface onto the specified default network, the dynamic routing protocols running on that communication server will generate or source a default route. 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 communication server 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 communication server.

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

Task	Command
Specify a default network.	<b>ip default-network</b> <i>network-number</i>

## 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 may be several networks that can be candidates for the system default. The communication server uses both administrative distance and metric information to determine the default route (gateway of last resort). The selected default route appears in the gateway of last resort display of the **show ip route EXEC** command.

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

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

## Redistribute Routing Information

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

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

The following two tables list tasks associated with route redistribution

Perform the following task in global configuration mode:

Task	Command
Define any route maps needed to control redistribution.	<b>route-map</b> <i>map-tag</i> { <b>permit</b>   <b>deny</b> } <i>sequence-number</i>

Perform the following tasks in router configuration mode:

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

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. These 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 communication server 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 communication servers from learning a particular communication server's interpretation of one or more routes.
- Suppress a routing protocol from both sending and receiving updates on a particular interface. This is usually done when a wildcard command has been used to configure the routing protocol for more communication server interfaces than is desirable.
- Suppress networks listed in updates from being accepted and acted upon by a routing process. This is done to keep a communication server 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.

Use the following tables to help you perform these tasks.

## Prevent Routing Updates through an Interface

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

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



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

Task	Command
Suppress the sending of routing updates through the specified communication server interface.	<b>passive-interface</b> <i>interface</i>

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 communication servers from learning one or more routes, you can suppress routes from being advertised in routing updates. The interface-name argument of this feature does not apply to 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.	<b>distribute-list</b> <i>access-list-number</i> <b>out</b> [ <i>interface-name</i>   <i>routing-process</i> ]

### 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.

Perform this task in router configuration mode:

Task	Command
Suppress routes listed in updates from being processed.	<b>distribute-list</b> <i>access-list-number</i> <b>in</b> [ <i>interface-name</i> ]

### 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.	<b>offset-list</b> { <b>in</b>   <b>out</b> } <i>offset</i> [ <i>access-list-number</i> ]

### Filter Sources of Routing Information

An *administrative distance* is a rating of the trustworthiness of a routing information source, such as an individual communication server or a group of communication servers. In a large network, some routing protocols and some communication servers can be more reliable than others as sources of routing information. Also, when multiple routing processes are running in the same communication server for IP, it is possible for the same route to be advertised by more than one routing process. By

specifying administrative distance values, you enable the communication server to intelligently discriminate between sources of routing information. The communication server will always pick the route whose routing protocol has the lowest administrative distance.

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

Task	Command
Filter on routing information sources.	<b>distance weight</b> [ <i>address-mask</i> [ <i>access-list-number</i> ]] [ <b>ip</b> ]

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

---

**Note** You also can use administrative distance to rate the routing information from communication servers running the same routing protocol. This application is generally discouraged if you are unfamiliar with this particular use of administrative distance, since 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
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 (in milliseconds) for which routing updates will be postponed

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

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

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

Perform the following task in router configuration mode:

Task	Command
Adjust routing protocol timers.	<b>timers basic update invalid holddown flush</b> [ <i>sleep</i> time]

Perform the following the following task interface configuration mode:

Task	Command
Adjust the frequency with which the communication server sends messages to itself (Ethernet and Token Ring) or to the other end (hdlc-serial and ppp-serial links) to ensure that a network interface is alive for a specified interface.	<b>keepalive</b> [ <i>seconds</i> ]

You also can configure the *keepalive* interval, the frequency at which the communication server sends messages to itself (Ethernet and Token Ring) or to the other end (hdlc-serial, ppp-serial) to ensure that a network interface is alive. The interval in some previous software versions was 10 seconds; it is now adjustable in 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 Split Horizon

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

To disable split horizon, perform the following task in interface configuration mode:

Task	Command
Disable split horizon.	<b>no ip split-horizon</b>

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.

---

**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.

---

## Monitor and Maintain the IP Network

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

### Clear Caches, Tables, and Databases

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

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

Task	Command
Reset a particular BGP connection.	<b>clear ip bgp <i>address</i></b>
Reset all BGP connections.	<b>clear ip bgp *</b>
Clear one or more routes from the IP routing table.	<b>clear ip route {<i>network</i> [<i>mask</i>]   *}</b>

### Display System and Network Statistics

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

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

Task	Command
Display the contents of the BGP routing table.	<b>show ip bgp</b> [ <i>network</i> ]
Display detailed information on the TCP and BGP connections to individual neighbors.	<b>show ip bgp neighbors</b> [ <i>address</i> ]
Display routes learned from a particular BGP neighbor.	<b>show ip bgp neighbors</b> <i>address</i> [ <b>routes</b> ]
Display all BGP paths in the database.	<b>show ip bgp paths</b>
Display the status of all BGP connections.	<b>show ip bgp summary</b>
Display statistics on EGP connections and neighbors.	<b>show ip egp</b>
Display IRDP values.	<b>show ip irdp</b>
Display general information about OSPF routing processes in a particular communication server.	<b>show ip ospf</b> [ <i>ospf-process-id</i> ]
Display lists of information related to the OSPF database for a specific communication server.	<b>show ip ospf</b> [ <i>ospf-process-id area-id</i> ] <b>database</b> <b>show ip ospf</b> [ <i>ospf-process-id area-id</i> ] <b>database</b> <b>[communication server]</b> [ <i>link-state-id</i> ] <b>show ip ospf</b> [ <i>ospf-process-id area-id</i> ] <b>database</b> <b>[network]</b> [ <i>link-state-id</i> ] <b>show ip ospf</b> [ <i>ospf-process-id area-id</i> ] <b>database</b> <b>[summary]</b> [ <i>link-state-id</i> ] <b>show ip ospf</b> [ <i>ospf-process-id area-id</i> ] <b>database</b> <b>[asb-summary]</b> [ <i>link-state-id</i> ] <b>show ip ospf</b> [ <i>ospf-process-id</i> ] <b>database</b> <b>[external]</b> [ <i>link-state-id</i> ]
Display OSPF-related interface information.	<b>show ip ospf interface</b> [ <i>interface-name</i> ]
Display OSPF-neighbor information on a per-interface basis.	<b>show ip ospf neighbor</b> [ <i>interface-name</i> ] [ <i>neighbor-id</i> ]
Display OSPF-related virtual links information.	<b>show ip ospf virtual-links</b>
Display the parameters and current state of the active routing protocol process.	<b>show ip protocols</b>
Display the current state of the routing table.	<b>show ip route</b> [ <i>address [mask]</i> ]   [ <i>protocol [process-id]</i> ]
Display the current state of the routing table in summary form.	<b>show ip route summary</b>
Display all route-maps configured or only the one specified.	<b>show route-map</b> [ <i>map-name</i> ]

## IP Routing Protocol Configuration Examples

This section contains configuration examples of various IP routing protocol features. The topics addressed are as follows:

- Variable-Length Subnet Masks Example
- Example of Overriding Static Routes with Dynamic Protocols
- 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
- Third-Party EGP Support Example
- Backup EGP Router Example
- EGP Core Gateway Example
- Example of Autonomous System within EGP
- Passive Interface Examples
- Administrative Distance Examples

### Variable-Length Subnet Masks Example

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

In the following example, a 14-bit subnet mask is used, leaving two bits of address space reserved for serial line host addresses. This 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 ethernet

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

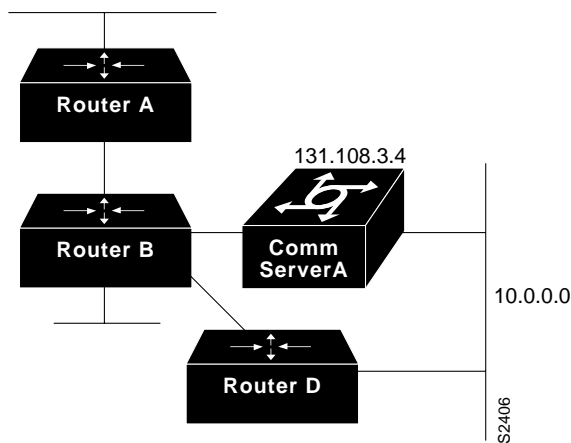
! Router is configured for OSPF and assigned AS 107
router ospf 107
! Specifies network directly connected to the communication server
network 131.107.0.0 0.0.255.255 area 0.0.0.0
```

## Example of Overriding Static Routes with Dynamic Protocols

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 less than 110 may 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



## 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 communication server to minimize the likelihood of creating a routing loop.

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

## IGRP Redistribution Example

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

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

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

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

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

## RIP and IGRP Redistribution Example

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

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

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

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

## OSPF Routing and Route Redistribution Examples

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

Three examples follow:

- The first is a simple configuration illustrating basic OSPF commands.



- The second example illustrates a configuration for internal, area border, and AS boundary communication servers within a single, arbitrarily assigned, OSPF AS.
- 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 communication server sequentially evaluates the *address/wildcard-mask* pair for each interface as follows:

- Step 1** The *wildcard-mask* is logically ORed with the interface IP address.
- Step 2** The *wildcard-mask* is logically ORed with *address* in the **network** command.
- Step 3** The communication server compares the two resulting values.
- Step 4** If they match, OSPF is enabled on the associated interface and this interface is attached to the OSPF area specified.

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.

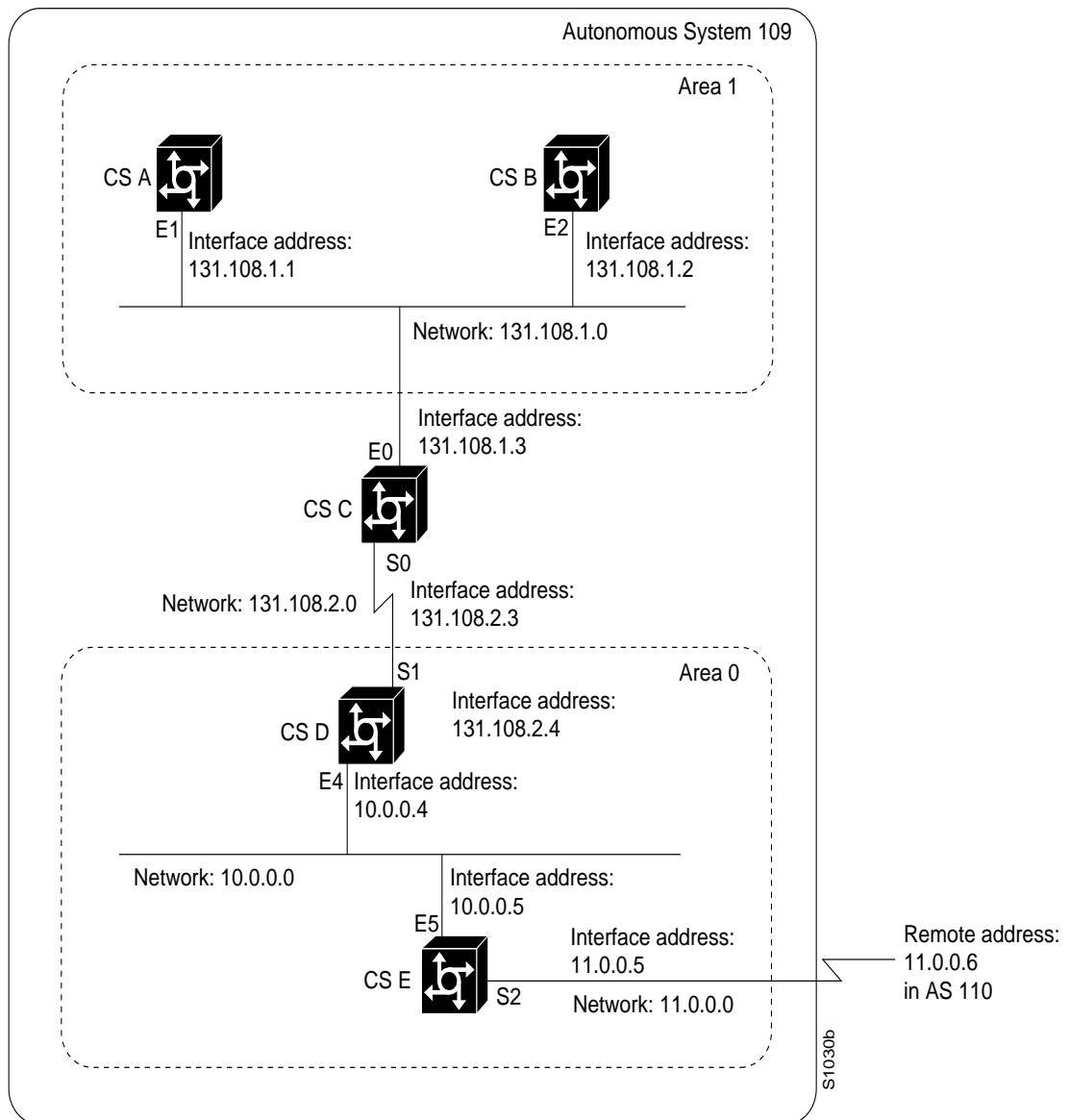
The second **network** command is evaluated next. For Area 2, the same process is then applied to all interfaces (except interface Ethernet 0). Assume that a match is determined for interface Ethernet 1. OSPF is then enabled for that interface and Ethernet 1 is attached to Area 2.

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 AS Boundary Routers

The following example outlines a configuration for several communication servers within a single OSPF autonomous system. Figure 1-4 provides a general network map that illustrates this example configuration.

Figure 1-4 Sample OSPF Autonomous System Network Map



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

- Router A and Router B are both internal communication servers within Area 1.
- Router C is an OSPF area border communication server; note that for Router C, Area 1 is assigned to subnet 1 and Area 0 is assigned to subnet 2.
- Router D is an internal communication server 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 AS boundary communication server; 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 AS in the configuration of all communication servers in the AS. You need only define the *directly* connected areas. In the example that follows, routes in Area 0 are learned by the communication servers in Area 1 (Router A and Router B) when the area border communication server (Router C) injects summary link state advertisements (LSAs) into Area 1.

---

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

```
! Router A - internal communication server
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

! Router B - internal communication server
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
!
! Router C - area border communication server
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
!
! Router D - internal communication server
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
!
! Router E - AS boundary communication server
```

```

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

```

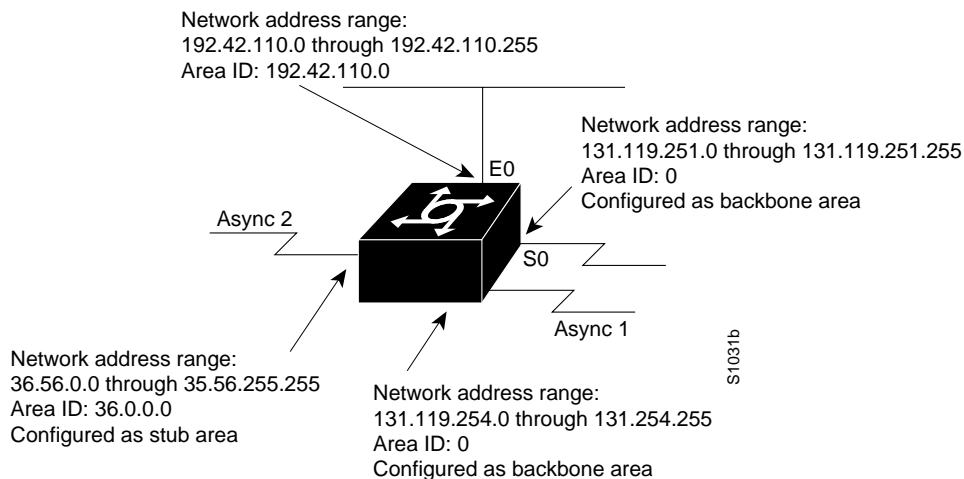
#### Example 4: Complex OSPF Configuration

The following example configuration accomplishes several tasks in setting up an area border communication server. 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 1-5 illustrates the network address ranges and area assignments for the interfaces.

Figure 1-5 Interface and Area Specifications for OSPF Example Configuration



The basic configuration tasks in this example are as follows:

- Configure address ranges for Ethernet 0 through Ethernet 3 interfaces.
- Enable OSPF on each interface.
- Set up an OSPF authentication password for each area and network.
- Assign link state metrics and other OSPF interface configuration options.

- Create a *stub area* with area id 36.0.0.0. (Note that the **authentication** and **stub** options of the **area** router configuration command are specified with separate **area** command entries, but can be merged into a single **area** command.)
- Specify the backbone area (Area 0).

Configuration tasks associated with redistribution are as follows:

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

The following is an example OSPF configuration listing:

```
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 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 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 communication servers that run IGRP within their respective ASs 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 communication servers (R1 and R2) that are intended to advertise BGP routes to each other and to redistribute BGP into IGRP.

```

! Router R1 Configuration:
! Assumes AS 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
!
! Router R2 Configuration:
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 communication server 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 communication server 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.

```

! Configuration for communication server R1:
router bgp 1
network 131.108.0.0
neighbor 192.5.10.1 remote-as 2
neighbor 192.5.10.15 remote-as 1
! 131.108.2.1 an internal peer
neighbor 192.5.10.24 remote-as 3
! 131.108.3.1 is an external peer
redistribute 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 communication server in autonomous system 109 using both the RIP and the IGRP routing protocols. The example advertises IGRP-derived routes using the RIP protocol and assigns the IGRP-derived routes a RIP metric of 10.

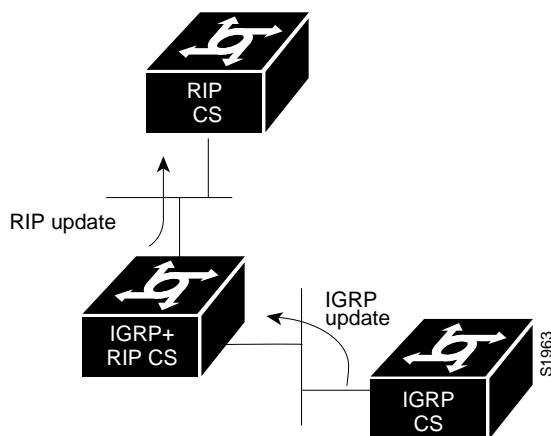
```

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

Figure 1-6 shows this type of redistribution.

**Figure 1-6 Assigning Metrics for Redistribution**

- IGRP Communication Server to RIP
- RIP Communication Server to IGRP





## Route Map Examples

The examples in this section illustrate the use of redistribution in various cases, with and without route maps.

The following redistributes all OSPF routes into IGRP:

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

The following 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 type1 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 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 redistributes OSPF intra-area and interarea routes with next hop communication servers 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
```

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 exists 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 ip destination 2
 set metric 1

access-list 1 permit 140.222.0.0 0.0.255.255
access-list 2 permit 0.0.0.0 0.0.0.0
```

The following configuration illustrates how a route map is referenced by the **default-information** router configuration command. This behavior 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 a 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

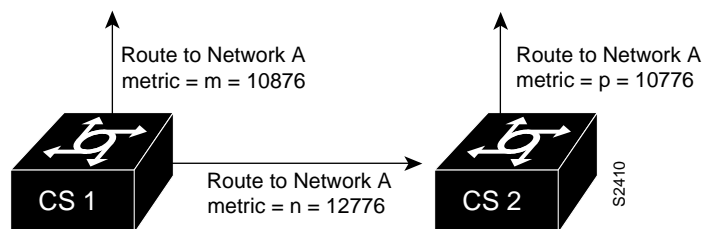
access-list 1 140.222.0.0 0.0.255.255

router ospf 109
 default-information originate route-map ospf-default
```

## IGRP Feasible Successor Relationship Example

In Figure 1-7, 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 1-7 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 the following two conditions are met, the route to network A through C2 will be included in C1's routing table:

- 1 If  $m$  is greater than  $p$ .
- 2 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:

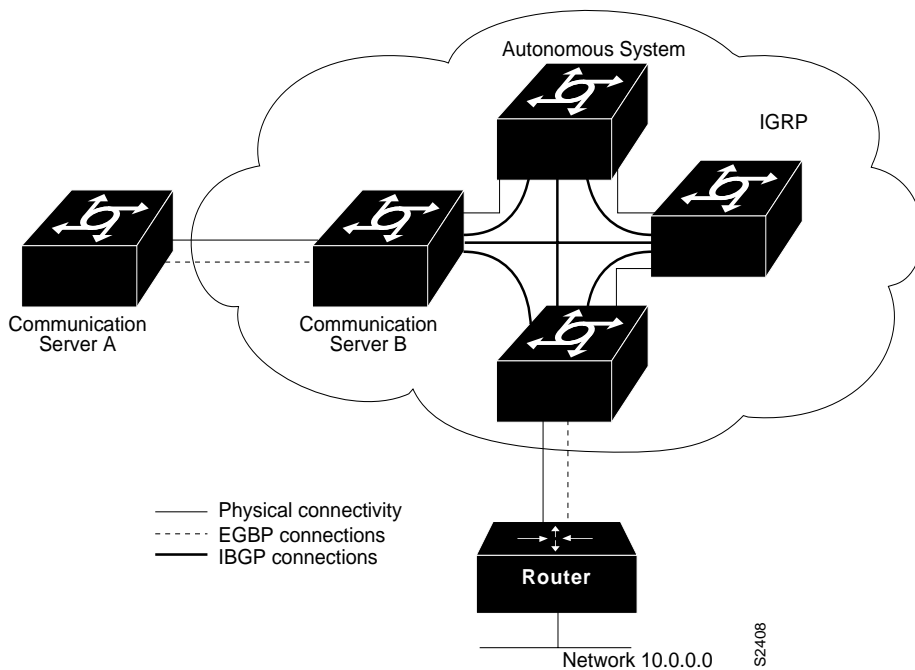
```
router igrp 109
variance 10
```

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 1-8, 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 1-8 Illustration of Synchronization



## BGP Basic Neighbor Specification Examples

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

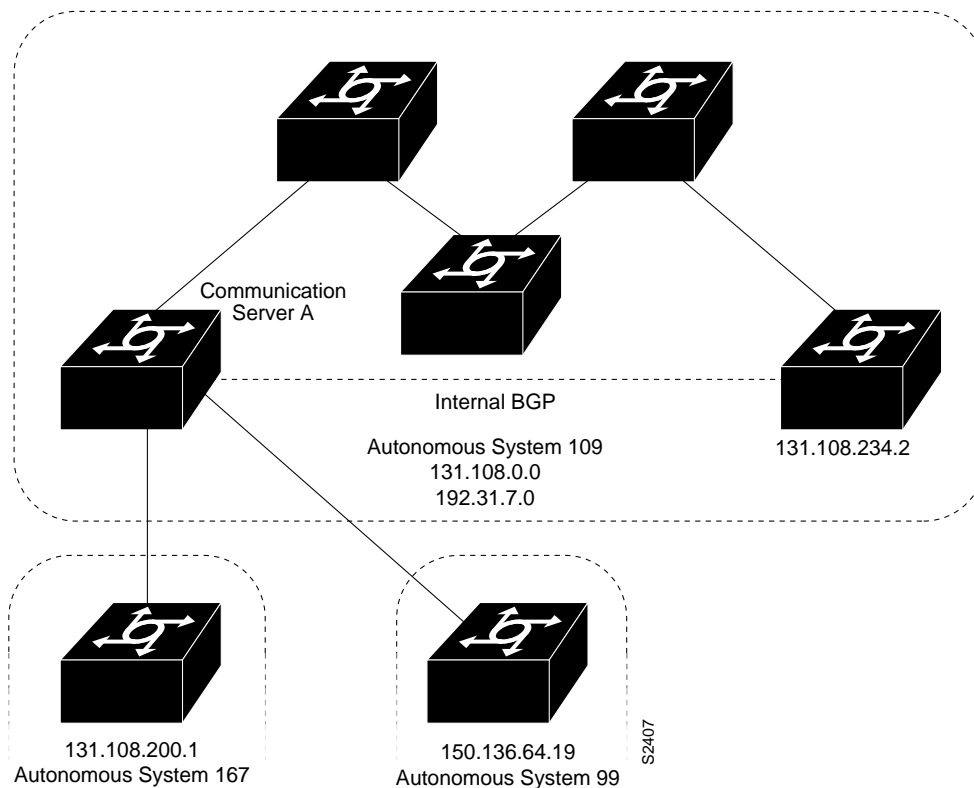
```
neighbor 131.108.1.2 remote-as 109
```

In the following example, a BGP communication server is assigned to AS 109, and two networks are listed as originating in the AS. Then the addresses of three remote communication servers (and their ASs) are listed. The communication server being configured will share information about networks 131.108.0.0 and 192.31.7.0 with the neighbor communication servers. The first communication server listed is in the same Class B network, but in a different AS; 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 1-9, 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 1-9 Assigning Internal and External BGP Neighbors



### Using Access Lists to Specify Neighbors

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

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

### Third-Party EGP Support Example

In this example configuration, the communication server is in AS 110 communicating with an EGP neighbor in AS 109 with address 131.108.6.5. Network 131.108.0.0 is advertised as originating within AS 110. The configuration specifies that two communication servers, 131.108.6.99 and 131.108.6.100, should be advertised as third-party sources of routing information for those networks that are accessible through those communication servers. The global configuration commands also specify that those networks should be flagged as internal to AS 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 communication server that is in AS 110 communicating with an EGP neighbor in AS 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 1-10 illustrates an environment with three communication servers (designated C1, C2, and C3) attached to a common X.25 network. The communication servers are intended to route information using EGP. With the following configuration (on the communication server designated Core), C1, C2, and C3 cannot route traffic directly to each other via the X.25 network:

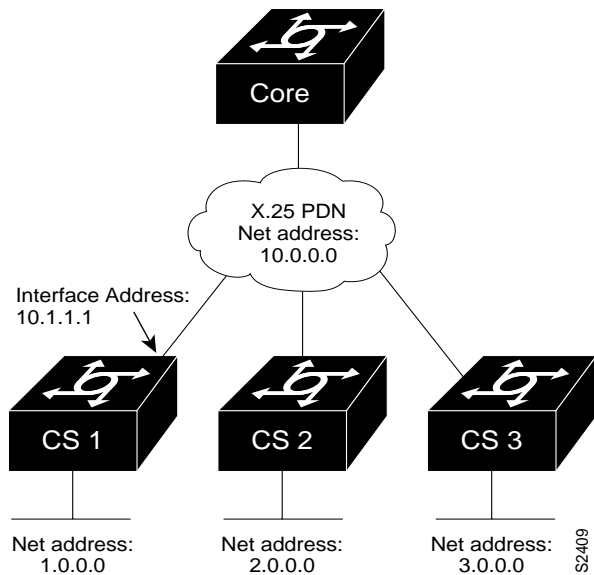
```
access-list 1 permit 10.0.0.0 0.255.255.255
! global access list assignment
router egp 0
neighbor any 1
```

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

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

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

Figure 1-10 Core EGP Third-Party Update Configuration Example



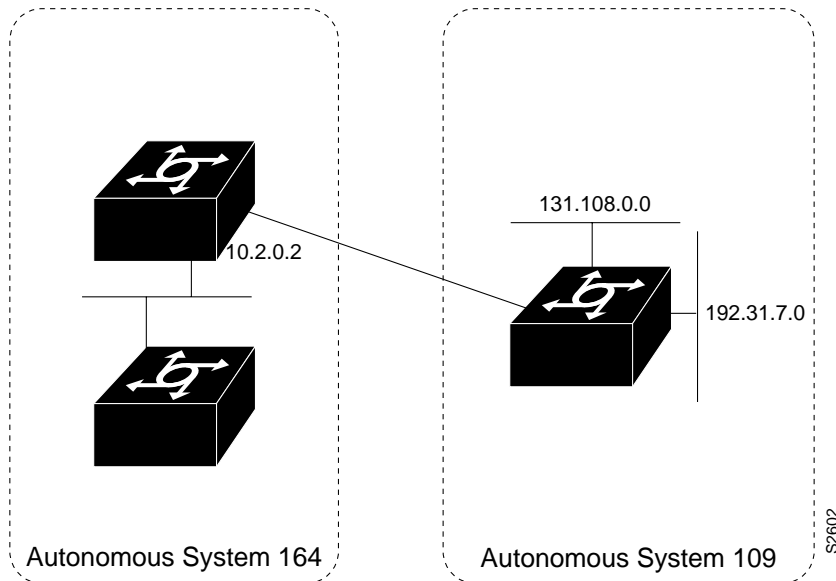
## Example of Autonomous System within EGP

The following example illustrates a typical configuration for an EGP communication server process. The communication server is in autonomous system 109 and is peering with communication servers in AS 164, as shown in Figure 1-11. It will advertise the networks 131.108.0.0 and 192.31.7.0 to the

communication server in AS 164, 10.2.0.2. The information sent and received from peer communication servers can be filtered in various ways, including blocking information from certain communication servers and suppressing the advertisement of specific routes.

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

Figure 1-11 Router in AS 164 Peers with Router in AS 109

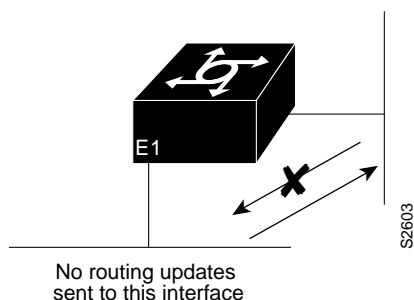


## Passive Interface Examples

The following example sends IGRP updates to all interfaces on network 131.108.0.0 except interface Ethernet 1. Figure 1-12 shows this configuration.

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

Figure 1-12 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 communication server will not be able to discover any neighbors, and none of the OSPF neighbors will be able to see the communication server on that network. In effect, this interface will appear as a stub network to the OSPF domain. This is useful if you want to import routes associated with a connected network into the OSPF domain without any OSPF activity on that interface.

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

```
interface Ethernet 0
ip address 131.108.1.1 255.255.255.0
interface Ethernet 1
ip address 131.108.2.1 255.255.255.0
interface Ethernet 2
ip address 131.108.3.1 255.255.255.0

router ospf 109
network 131.108.0.0 0.0.255.255 area 0
```

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

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



## Administrative Distance Examples

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

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

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

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

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

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

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

