Configuring IP Routing Protocols

This chapter describes how to configure the various Internet Protocol (IP) routing protocols. For a complete description of the commands listed in this chapter, refer to the “IP Routing Protocols Commands” chapter of the Network Protocols Command Reference, Part 1. For information on configuring the IP protocol, refer to the “Configuring IP” chapter of this manual.

IP Routing Protocols Task List

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

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

• Determine a Routing Process
• Configure IGRP
• Configure Enhanced IGRP
• Configure OSPF
• Configure Stub Routing
• Configure RIP
• Configure IS-IS
• Configure BGP
• Configure EGP
• Configure GDP (which, in future Cisco IOS software releases, will not be supported)
• Configure IRDP
• Configure Resource Reservation Protocol (RSVP)
• Configure IP Multicast Routing
• Configure Routing Protocol-Independent Features
• Monitor and Maintain the IP Network

See the end of this chapter for IP routing protocol configuration examples.
Determine a Routing Process

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

• Internetwork size and complexity
• Support for variable-length subnet masks (VLSM). Enhanced IGRP, IS-IS, static routes, and OSPF support VLSM
• Internetwork traffic levels
• Security needs
• Reliability needs
• Internetwork delay characteristics
• Organizational policies
• Organizational acceptance of change

The following sections describe the configuration tasks associated with each supported routing protocol. This publication does not provide in-depth information on how to choose routing protocols; you must choose routing protocols that best suit your needs.

Configure IGRP

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

Cisco’s IGRP Implementation

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

IGRP also advertises three types of routes: interior, system, and exterior, as shown in Figure 18. Interior routes are routes between subnets in the network attached to a router interface. If the network attached to a router is not subnetted, IGRP does not advertise interior routes.
System routes are routes to networks within an autonomous system. The Cisco IOS software derives system routes from directly connected network interfaces and system route information provided by other IGRP-speaking routers or access 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 Cisco IOS software chooses a gateway of last resort from the list of exterior routes that IGRP provides. The software uses the gateway (router) of last resort if it does not have a better route for a packet and the destination is not a connected network. If the autonomous system has more than one connection to an external network, different routers can choose different exterior routers as the gateway of last resort.

IGRP Updates

By default, a router running IGRP sends an update broadcast every 90 seconds. It declares a route inaccessible if it does not receive an update from the first router in the route within 3 update periods (270 seconds). After 7 update periods (630 seconds), the Cisco IOS software removes the route from the routing table.

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

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

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

Create the IGRP Routing Process

To create the IGRP routing process, perform the following required tasks starting in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enable an IGRP routing process, which places you in router configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Associate networks with an IGRP routing process.</td>
</tr>
</tbody>
</table>

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

It is not necessary to have a registered autonomous system number to use IGRP. If you do not have a registered number, you are free to create your own. We recommend that if you do have a registered number, you use it to identify the IGRP process.

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 Cisco IOS software to permit this exchange of routing information.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define a neighboring router with which to exchange point-to-point routing information.</td>
<td><code>neighbor ip-address</code></td>
</tr>
</tbody>
</table>

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define the variance associated with a particular path.</td>
<td>variance multiplier</td>
</tr>
</tbody>
</table>

Note: By using the variance feature, the Cisco IOS software 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 Cisco IOS software 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:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribute traffic proportionately to the ratios of metrics, or by the minimum-cost route.</td>
<td>traffic-share { balanced</td>
</tr>
</tbody>
</table>
Adjust the IGRP Metric Weights

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjust the IGRP metric.</td>
<td><code>metric weights tos k1 k2 k3 k4 k5</code></td>
</tr>
</tbody>
</table>

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

Disable Holddown

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable the IGRP holddown period.</td>
<td><code>no metric holddown</code></td>
</tr>
</tbody>
</table>

Enforce a Maximum Network Diameter

The Cisco IOS software enforces a maximum diameter to the IGRP network. Routes whose hop counts exceed this diameter are not 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:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the maximum network diameter.</td>
<td><code>metric maximum-hops hops</code></td>
</tr>
</tbody>
</table>
Validate Source IP Addresses

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable the checking and validation of the source IP address of incoming routing updates.</td>
<td>no validate-update-source</td>
</tr>
</tbody>
</table>

Configure Enhanced IGRP

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

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

Cisco’s Enhanced IGRP Implementation

IP Enhanced IGRP provides the following features:

- Automatic redistribution—IP IGRP routes can be automatically redistributed into Enhanced IGRP, and IP Enhanced IGRP routes can be automatically redistributed into IGRP. If desired, you can turn off redistribution. You can also completely turn off IP Enhanced IGRP and IP IGRP on the router or on individual interfaces.

- Increased network width—With IP RIP, the largest possible width of your network is 15 hops. When IP Enhanced IGRP is enabled, the largest possible width is 224 hops. Because the Enhanced IGRP metric is large enough to support thousands of hops, the only barrier to expanding the network is the transport layer hop counter. Cisco works around this problem by incrementing the transport control field only when an IP packet has traversed 15 routers and the next hop to the destination was learned by way of Enhanced IGRP. When a RIP route is being used as the next hop to the destination, the transport control field is incremented as usual.

Enhanced IGRP offers the following features:

- Fast convergence—The DUAL algorithm allows routing information to converge as quickly as any currently available routing protocol.

- Partial updates—Enhanced IGRP sends incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table. This feature minimizes the bandwidth required for Enhanced IGRP packets.

- Less CPU usage than IGRP—This occurs because full update packets do not have to be processed each time they are received.

- Neighbor discovery mechanism—This is a simple hello mechanism used to learn about neighboring routers. It is protocol-independent.

- Variable-length subnet masks
Configure Enhanced IGRP

- Arbitrary route summarization
- Scaling. Enhanced IGRP scales to large networks

Enhanced IGRP has four basic components:
- Neighbor discovery/recovery
- Reliable transport protocol
- DUAL finite state machine
- Protocol-dependent modules

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

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

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

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

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

- Enable IP Enhanced IGRP
- Transition from IGRP to Enhanced IGRP
- Configure IP Enhanced IGRP-Specific Parameters
- Display System and Network Statistics
- Configure Protocol-Independent Parameters

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

Enable IP Enhanced IGRP

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enable an IP Enhanced IGRP routing process in global configuration mode. <strong>router eigrp process-number</strong></td>
</tr>
<tr>
<td>Step 2</td>
<td>Associate networks with an IP Enhanced IGRP routing process in router configuration mode. <strong>network network-number</strong></td>
</tr>
</tbody>
</table>

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

Transition from IGRP to Enhanced IGRP

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

Configure IP Enhanced IGRP-Specific Parameters

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

- Log Enhanced IGRP Neighbor Adjacency Changes
- Configure the Percentage of Link Bandwidth Used by Enhanced IGRP
- Display System and Network Statistics
- Adjust the IP Enhanced IGRP Metric Weights
- Disable Route Summarization
- Configure Summary Aggregate Addresses
Configure Enhanced IGRP

Log Enhanced IGRP Neighbor Adjacency Changes

You can enable the logging of neighbor adjacency changes to monitor the stability of the routing system and to help you detect problems. By default, adjacency changes are not logged.

To enable logging of Enhanced IGRP neighbor adjacency changes, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable logging of Enhanced IGRP neighbor adjacency changes.</td>
<td>log-neighbor-changes</td>
</tr>
</tbody>
</table>

Configure the Percentage of Link Bandwidth Used by Enhanced IGRP

By default, Enhanced IGRP packets consume a maximum of 50 percent of the link bandwidth, as configured with the `bandwidth` interface subcommand. If a different value is desired, use the `ip eigrp-bandwidth-percent` command. This command may be useful if a different level of link utilization is required or if the configured bandwidth does not match the actual link bandwidth (it may have been configured to influence route metric calculations).

To configure the percentage of bandwidth that may be used by Enhanced IGRP on an interface, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the percentage of bandwidth that may be used by Enhanced IGRP on an interface.</td>
<td>ip eigrp-bandwidth-percent percent</td>
</tr>
</tbody>
</table>

Display System and Network Statistics

You can display specific 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 that packets are taking through the network.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace a branch of a multicast tree for a specific group.</td>
<td>mbranch group-address branch-address [ttl]</td>
</tr>
<tr>
<td>Trace a branch of a multicast tree for a group in the reverse direction.</td>
<td>mrbranch group-address branch-address [ttl]</td>
</tr>
<tr>
<td>Display all BGP routes that contain subnet and supernet network masks.</td>
<td>show ip bgp cidr-only</td>
</tr>
<tr>
<td>Display routes that belong to the specified communities.</td>
<td>show ip bgp community community-number [exact]</td>
</tr>
<tr>
<td>Display routes that are permitted by the community list.</td>
<td>show ip bgp community-list community-list-number [exact]</td>
</tr>
<tr>
<td>Display routes that are matched by the specified autonomous system path access list.</td>
<td>show ip bgp filter-list access-list-number</td>
</tr>
<tr>
<td>Display the routes with inconsistent originating autonomous systems.</td>
<td>show ip bgp inconsistent-as</td>
</tr>
<tr>
<td>Task</td>
<td>Command</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Display the routes that match the specified regular expression entered on the command line.</td>
<td><code>show ip bgp regexp regular-expression</code></td>
</tr>
<tr>
<td>Display the contents of the BGP routing table.</td>
<td><code>show ip bgp [network] [network-mask] [subnets]</code></td>
</tr>
<tr>
<td>Display detailed information on the TCP and BGP connections to individual neighbors.</td>
<td><code>show ip bgp neighbors [address]</code></td>
</tr>
<tr>
<td>Display routes learned from a particular BGP neighbor.</td>
<td>`show ip bgp neighbors address [routes</td>
</tr>
<tr>
<td>Display all BGP paths in the database.</td>
<td><code>show ip bgp paths</code></td>
</tr>
<tr>
<td>Display information about BGP peer groups.</td>
<td><code>show ip bgp peer-group [tag] [summary]</code></td>
</tr>
<tr>
<td>Display the status of all BGP connections.</td>
<td><code>show ip bgp summary</code></td>
</tr>
<tr>
<td>Display the entries in the DVMRP routing table.</td>
<td><code>show ip dvmrp route [ip-address]</code></td>
</tr>
<tr>
<td>Display statistics on EGP connections and neighbors.</td>
<td><code>show ip egp</code></td>
</tr>
<tr>
<td>Display information about interfaces configured for Enhanced IGRP</td>
<td><code>show ip eigrp interfaces [interface] [as-number]</code></td>
</tr>
<tr>
<td>Display the IP Enhanced IGRP discovered neighbors.</td>
<td><code>show ip eigrp neighbors [type number]</code></td>
</tr>
<tr>
<td>Display the IP Enhanced IGRP topology table for a given process.</td>
<td><code>show ip eigrp topology [autonomous-system-number] [ip-address] mask]</code></td>
</tr>
<tr>
<td>Display the number of packets sent and received for all or a specified IP Enhanced IGRP process.</td>
<td><code>show ip eigrp traffic [autonomous-system-number]</code></td>
</tr>
<tr>
<td>Display the multicast groups that are directly connected to the router and that were learned via IGMP.</td>
<td>`show ip igmp groups [group-name</td>
</tr>
<tr>
<td>Display multicast-related information about an interface.</td>
<td><code>show ip igmp interface [type number]</code></td>
</tr>
<tr>
<td>Display IRDP values.</td>
<td><code>show ip irdp</code></td>
</tr>
<tr>
<td>Display the contents of the IP fast switching cache.</td>
<td><code>show ip mcache [group [source]]</code></td>
</tr>
<tr>
<td>Display the contents of the IP multicast routing table.</td>
<td><code>show ip mroute [group] [source] [summary] [count]</code></td>
</tr>
<tr>
<td>Display general information about OSPF routing processes.</td>
<td><code>show ip ospf [process-id]</code></td>
</tr>
<tr>
<td>Display lists of information related to the OSPF database.</td>
<td><code>show ip ospf [process-id area-id] database</code></td>
</tr>
<tr>
<td></td>
<td><code>show ip ospf [process-id area-id] database [router] [link-state-id]</code></td>
</tr>
<tr>
<td></td>
<td><code>show ip ospf [process-id area-id] database [network] [link-state-id]</code></td>
</tr>
<tr>
<td></td>
<td><code>show ip ospf [process-id area-id] database [summary] [link-state-id]</code></td>
</tr>
<tr>
<td></td>
<td><code>show ip ospf [process-id area-id] database [asb-summary] [link-state-id]</code></td>
</tr>
<tr>
<td></td>
<td><code>show ip ospf [process-id] database [external] [link-state-id]</code></td>
</tr>
<tr>
<td></td>
<td><code>show ip ospf [process-id area-id] database [database-summary]</code></td>
</tr>
</tbody>
</table>
Configure Enhanced IGRP

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display the internal OSPF routing table entries to Area Border Router (ABR) and Autonomous System Boundary Router (ASBR).</td>
<td>show ip ospf border-routers</td>
</tr>
<tr>
<td>Display OSPF-related interface information.</td>
<td>show ip ospf interface [interface-name]</td>
</tr>
<tr>
<td>Display OSPF-neighbors information on a per-interface basis.</td>
<td>show ip ospf neighbor [interface-name] [neighbor-id] detail</td>
</tr>
<tr>
<td>Display OSPF-related virtual links information.</td>
<td>show ip ospf virtual-links</td>
</tr>
<tr>
<td>Display information about interfaces configured for PIM.</td>
<td>show ip pim interface [type number]</td>
</tr>
<tr>
<td>List the PIM neighbors discovered by the router.</td>
<td>show ip pim neighbor [type number]</td>
</tr>
<tr>
<td>Display the RP routers associated with a sparse-mode multicast group.</td>
<td>show ip pim rp [group-name] [group-address]</td>
</tr>
<tr>
<td>Display the local policy route map, if any.</td>
<td>show ip local policy</td>
</tr>
<tr>
<td>Display policy route maps.</td>
<td>show ip policy</td>
</tr>
<tr>
<td>Display the parameters and current state of the active routing protocol process.</td>
<td>show ip protocols</td>
</tr>
<tr>
<td>Display the current state of the routing table.</td>
<td>show ip route [address [mask] [longer-prefixes]] [protocol [process-id]]</td>
</tr>
<tr>
<td>Display the current state of the routing table in summary form.</td>
<td>show ip route summary</td>
</tr>
<tr>
<td>Display supernets.</td>
<td>show ip route supernets-only</td>
</tr>
<tr>
<td>Display the contents of the session directory.</td>
<td>show ip sd [group] [session-name] detail</td>
</tr>
<tr>
<td>Display the IS-IS link state database.</td>
<td>show isis database [level-1] [level-2] [l1] [l2] [detail] [lspid]</td>
</tr>
<tr>
<td>Display authentication key information.</td>
<td>show key chain [name]</td>
</tr>
<tr>
<td>Display all route maps configured or only the one specified.</td>
<td>show route-map [map-name]</td>
</tr>
</tbody>
</table>

**Note** By using the variance feature, the Cisco IOS software balances traffic across all feasible paths and immediately converges to a new path if one paths fails.

Adjust the IP Enhanced IGRP Metric Weights

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjust the IP Enhanced IGRP metric.</td>
<td>metric weights tos k1 k2 k3 k4 k5</td>
</tr>
</tbody>
</table>
Configure Enhanced IGRP

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

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

### Disable Route Summarization

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable automatic summarization.</td>
<td>no auto-summary</td>
</tr>
</tbody>
</table>

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

### Configure Summary Aggregate Addresses

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure a summary aggregate address.</td>
<td>ip summary-address eigrp</td>
</tr>
<tr>
<td></td>
<td>autonomous-system-number address mask</td>
</tr>
</tbody>
</table>

### Configure Protocol-Independent Parameters

To configure protocol-independent parameters, perform one or more of the tasks in the following sections:

- Redistribute Routing Information
- Set Metrics for Redistributed Routes
- Filter Routing Information
- Adjust the Interval between Hello Packets and the Hold Time
- Disable Split Horizon
Redistribute Routing Information

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

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

To redistribute routes from one protocol into another, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redistribute routes from one routing protocol into another.</td>
<td>redistribute protocol autonomous-system-number [route-map map-tag]</td>
</tr>
</tbody>
</table>

To define route maps, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define any route maps needed to control redistribution.</td>
<td>route-map map-tag [permit</td>
</tr>
</tbody>
</table>

By default, the redistribution of default information between IP Enhanced IGRP processes is enabled. To disable the redistribution, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable the redistribution of default information between IP Enhanced IGRP processes.</td>
<td>no default-information {in</td>
</tr>
</tbody>
</table>

Set Metrics for Redistributed Routes

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 IP Enhanced 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.

To set metrics for redistributed routes, perform the first task when redistributing from IP Enhanced IGRP, and perform the second task when redistributing into IP Enhanced IGRP. Each task is done in router configuration mode.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cause the current routing protocol to use the same metric value for all redistributed routes.</td>
<td>default-metric number</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cause the IP Enhanced IGRP routing protocol to use the same metric value for all non-IGRP redistributed routes.</td>
<td>default-metric bandwidth delay reliability loading mtu</td>
</tr>
</tbody>
</table>
Filter Routing Information

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

- Suppress the sending of routing updates on a particular interface. Doing so prevents other systems on an interface from learning about routes dynamically.
- Suppress networks from being advertised in routing updates. Doing so prevents other routers from learning a particular interpretation of one or more routes.
- Suppress a routing protocol from both sending and receiving updates on a particular interface. You usually perform this task when a wildcard command has been used to configure the routing protocol for more interfaces than is desirable.
- Suppress networks listed in updates from being accepted and acted upon by a routing process. Doing so keeps a router from using certain routes.
- Filter on the source of routing information. You perform this task to prioritize routing information from different sources, because the accuracy of the routing information can vary.
- Apply an offset to routing metrics. Doing so provides a local mechanism for increasing the value of routing metrics.

Use the information in the following sections to perform these tasks.

Prevent Routing Updates through an Interface

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppress the sending of routing updates</td>
<td><code>passive-interface type number</code></td>
</tr>
<tr>
<td>through an interface.</td>
<td></td>
</tr>
</tbody>
</table>

Control the Advertising of Routes in Routing Updates

To control which routers learn about routes, you can control the advertising of routes in routing updates. To do this, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control the advertising of routes in</td>
<td><code>distribute-list access-list-number [name [interface-name]</code></td>
</tr>
<tr>
<td>routing updates.</td>
<td>` routing-process</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Control the Processing of Routing Updates

To control the processing of incoming updates, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control which incoming route updates are</td>
<td>`distribute-list access-list-number</td>
</tr>
<tr>
<td>processed.</td>
<td><code>[interface-name]</code></td>
</tr>
</tbody>
</table>
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. To do so, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply an offset to routing metrics.</td>
<td>offset-list [access-list-number</td>
</tr>
</tbody>
</table>

Filter Sources of Routing Information
An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. In a large network, some routing protocols and some routers can be more reliable than others as sources of routing information. Also, when multiple routing processes are running in the same device for IP, the same route may be advertised by more than one routing process. Specifying administrative distance values enables the Cisco IOS software to discriminate between sources of routing information. The software always picks the route whose routing protocol has the lowest administrative distance.

There are no general guidelines for assigning administrative distances, because each network has its own requirements. You must determine a reasonable matrix of administrative distances for the network as a whole. Table 4 shows the default administrative distance for various routing information sources.

### Table 4  Default Administrative Distances

<table>
<thead>
<tr>
<th>Route Source</th>
<th>Default Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected interface</td>
<td>0</td>
</tr>
<tr>
<td>Static route</td>
<td>1</td>
</tr>
<tr>
<td>Enhanced IGRP summary route</td>
<td>5</td>
</tr>
<tr>
<td>External BGP</td>
<td>20</td>
</tr>
<tr>
<td>Internal Enhanced IGRP</td>
<td>90</td>
</tr>
<tr>
<td>IGRP</td>
<td>100</td>
</tr>
<tr>
<td>OSPF</td>
<td>110</td>
</tr>
<tr>
<td>IS-IS</td>
<td>115</td>
</tr>
<tr>
<td>RIP</td>
<td>120</td>
</tr>
<tr>
<td>EGP</td>
<td>140</td>
</tr>
<tr>
<td>External Enhanced IGRP</td>
<td>170</td>
</tr>
<tr>
<td>Internal BGP</td>
<td>200</td>
</tr>
<tr>
<td>Unknown</td>
<td>255</td>
</tr>
</tbody>
</table>

For example, consider a router using IP Enhanced IGRP and RIP. Suppose you trust the IP Enhanced IGRP-derived routing information more than the RIP-derived routing information. Because the default IP Enhanced IGRP administrative distance is lower than that for RIP, the Cisco IOS software uses the IP Enhanced IGRP-derived information and ignores the RIP-derived information. However, if you lose the source of the IP Enhanced IGRP-derived information (for example, because of a power shutdown), the software uses the RIP-derived information until the IP Enhanced IGRP-derived information reappears.
Configure Enhanced IGRP

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter on routing information sources.</td>
<td>distance eigrp internal-distance external-distance</td>
</tr>
</tbody>
</table>

Adjust the Interval between Hello Packets and the Hold Time

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

Routing devices periodically send hello packets to each other to dynamically learn of other routers on their directly attached networks. This information is used to discover who their neighbors are, and to learn when their neighbors become unreachable or inoperative.

By default, hello packets are sent every 5 seconds. The exception is on low-speed, nonbroadcast, multiaccess (NBMA) media, where the default hello interval is 60 seconds. Low speed is considered to be a rate of T1 or slower, as specified with the `bandwidth` interface configuration command. The default hello interval remains 5 seconds for high-speed NBMA networks. Note that for the purposes of Enhanced IGRP, Frame Relay and SMDS networks may or may not be considered to be NBMA. These networks are considered NBMA if the interface has not been configured to use physical multicasting; otherwise they are not considered NBMA.

You can configure the hold time on a specified interface for a particular IP Enhanced IGRP routing process designated by the autonomous system number. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is three times the hello interval, or 15 seconds. For slow-speed NBMA networks, the default hold time is 180 seconds.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the hello interval for an IP Enhanced IGRP routing process.</td>
<td>ip hello-interval eigrp autonomous-system-number seconds</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the hold time for an IP Enhanced IGRP routing process.</td>
<td>ip hold-time eigrp autonomous-system-number seconds</td>
</tr>
</tbody>
</table>

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

Disable Split Horizon

Split horizon controls the sending of IP Enhanced IGRP update and query packets. When split horizon is enabled on an interface, these packets are not sent for destinations for which this interface is the next hop. This reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

Split horizon blocks route information from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routing devices, 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 want to disable split horizon.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable split horizon.</td>
<td>no ip split-horizon eigrp autonomous-system-number</td>
</tr>
</tbody>
</table>

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.

We support RFC 1253, Open Shortest Path First (OSPF) MIB, August 1991. The OSPF MIB defines an IP routing protocol that provides management information related to OSPF and is supported by Cisco routers.

Cisco’s OSPF Implementation

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

- Stub areas—Definition of stub areas is supported.
- Route redistribution—Routes learned via any IP routing protocol can be redistributed into any other IP routing protocol. At the inadomain level, this means that OSPF can import routes learned via IGRP, RIP, and IS-IS. OSPF routes can also be exported into IGRP, RIP, and IS-IS. At the interdomain level, OSPF can import routes learned via EGP and BGP. OSPF routes can be exported into EGP and BGP.
- Authentication—Simple and MD5 authentication among neighboring routers within an area is supported.
- Routing interface parameters—Configurable parameters supported include interface output cost, retransmission interval, interface transmit delay, router priority, router “dead” and hello intervals, and authentication key.
- Virtual links—Virtual links are supported.
- NSSA areas - RFC 1567
- OSPF over demand circuit - RFC 1793
Configure OSPF

OSPF Configuration Task List

OSPF typically requires coordination among many internal routers, area border routers (routers connected to multiple areas), and autonomous system boundary routers. At a minimum, OSPF-based routers or access 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 routers.

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

- Enable OSPF
- Configure OSPF Interface Parameters
- Configure OSPF over Different Physical Networks
- Configure OSPF Area Parameters
- Configure OSPF Not So Stubby Area (NSSA)
- Configure Route Summarization between OSPF Areas
- Configure Route Summarization When Redistributing Routes into OSPF
- Create Virtual Links
- Generate a Default Route
- Configure Lookup of DNS Names
- Force the Router ID Choice with a Loopback Interface
- Disable Default OSPF Metric Calculation Based on Bandwidth
- Configure OSPF on Simplex Ethernet Interfaces
- Configure Route Calculation Timers
- Configure OSPF over On Demand Circuits

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

Enable OSPF

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Enable OSPF routing, which places you in router configuration mode.</td>
<td>router ospf process-id</td>
</tr>
<tr>
<td>Step 2 Define an interface on which OSPF runs and define the area ID for that interface.</td>
<td>network address wildcard-mask area area-id</td>
</tr>
</tbody>
</table>
Configure OSPF Interface Parameters

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicitly specify the cost of sending a packet on an OSPF interface.</td>
<td>ip ospf cost cost</td>
</tr>
<tr>
<td>Specify the number of seconds between link state advertisement retransmissions for adjacencies belonging to an OSPF interface.</td>
<td>ip ospf retransmit-interval seconds</td>
</tr>
<tr>
<td>Set the estimated number of seconds it takes to transmit a link state update packet on an OSPF interface.</td>
<td>ip ospf transmit-delay seconds</td>
</tr>
<tr>
<td>Set priority to help determine the OSPF designated router for a network.</td>
<td>ip ospf priority number</td>
</tr>
<tr>
<td>Specify the length of time, in seconds, between the hello packets that the Cisco IOS software sends on an OSPF interface.</td>
<td>ip ospf hello-interval seconds</td>
</tr>
<tr>
<td>Set the number of seconds that a device’s hello packets must not have been seen before its neighbors declare the OSPF router down.</td>
<td>ip ospf dead-interval seconds</td>
</tr>
<tr>
<td>Assign a specific password to be used by neighboring OSPF routers on a network segment that is using OSPF’s simple password authentication.</td>
<td>ip ospf authentication-key key</td>
</tr>
<tr>
<td>Enable OSPF MD5 authentication.</td>
<td>ip ospf message-digest-key keyid md5 key</td>
</tr>
</tbody>
</table>

Configure OSPF over Different Physical Networks

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

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

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

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

Configure Your OSPF Network Type

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

Configuring nonbroadcast, multiaccess networks as either broadcast or nonbroadcast assumes that
there are virtual circuits from every router to every router or fully meshed network. This is not true
for some cases, for example, because of cost constraints, or when you have only a partially meshed
network. In these cases, you can configure the OSPF network type as a point-to-multipoint network.
Routing between two routers not directly connected will go through the router that has virtual
circuits to both routers. Note that you must not configure neighbors when using this feature.

An OSPF point-to-multipoint interface is defined as a numbered point-to-point interface having one
or more neighbors. It creates multiple host routes. An OSPF point-to-multipoint network has the
following benefits compared to nonbroadcast multiaccess and point-to-point networks:

• Point-to-multipoint is easier to configure because it requires no configuration of neighbor
commands, it consumes only one IP subnet, and it requires no designated router election.

• It costs less because it does not require a fully meshed topology.

• It is more reliable because it maintains connectivity in the event of virtual circuit failure.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the OSPF network type for a specified interface.</td>
<td>`ip ospf network {broadcast</td>
</tr>
</tbody>
</table>

See the “Static Routing Redistribution Example” section at the end of this chapter for an example of
an OSPF point-to-multipoint network.

Configure OSPF for Nonbroadcast Networks

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

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure routers or access servers interconnecting to nonbroadcast networks.</td>
<td><code>neighbor ip-address [priority number] [poll-interval seconds]</code></td>
</tr>
</tbody>
</table>

You can specify the following neighbor parameters, as required:

• Priority for a neighboring router

• Nonbroadcast poll interval

• Interface through which the neighbor is reachable
Configure OSPF Area Parameters

Our OSPF software allows you to configure several area parameters. These area parameters, shown in the following table, include authentication, defining stub areas, and assigning specific costs to the default summary route. Authentication allows password-based protection against unauthorized access to an area.

*Stub areas* are areas into which information on external routes is not sent. Instead, there is a default external route generated by the area border router, into the stub area for destinations outside the autonomous system. To further reduce the number of link state advertisements sent into a stub area, you can configure `no-summary` on the ABR to prevent it from sending summary link advertisement (link state advertisements Type 3) into the stub area.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable authentication for an OSPF area.</td>
<td><code>area area-id authentication</code></td>
</tr>
<tr>
<td>Enable MD5 authentication for an OSPF area.</td>
<td><code>area area-id authentication message-digest</code></td>
</tr>
<tr>
<td>Define an area to be a stub area.</td>
<td><code>area area-id stub [no-summary]</code></td>
</tr>
<tr>
<td>Assign a specific cost to the default summary route used for the stub area.</td>
<td><code>area area-id default-cost cost</code></td>
</tr>
</tbody>
</table>

Configure OSPF Not So Stubby Area (NSSA)

NSSA area is similar to OSPF stub area. NSSA does not flood Type 5 external link state advertisements (LSAs) from the core into the area, but it has the ability of importing AS external routes in a limited fashion within the area.

NSSA allows importing of Type 7 AS external routes within NSSA area by redistribution. These Type 7 LSAs are translated into Type 5 LSAs by NSSA ABR which are flooded throughout the whole routing domain. Summarization and filtering are supported during the translation.

Use NSSA to simplify administration if you are an Internet service provider (ISP), or a network administrator that must connect a central site using OSPF to a remote site that is using a different routing protocol.

Prior to NSSA, the connection between the corporate site border router and the remote router could not be run as OSPF stub area because routes for the remote site cannot be redistributed into stub area. A simple protocol like RIP is usually run and handle the redistribution. This meant maintaining two routing protocols. With NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

In router configuration mode, specify the following area parameters as needed to configure OSPF NSSA:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define an area to be NSSA.</td>
<td><code>area area-id nssa [no-redistribution]</code></td>
</tr>
<tr>
<td></td>
<td><code>[default-information-originate]</code></td>
</tr>
</tbody>
</table>
In router configuration mode on the ABR, specify the following command to control summarization and filtering of Type 7 LSA into Type 5 LSA:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) Control the summarization and</td>
<td>summary address prefix mask [not advertise]</td>
</tr>
<tr>
<td>filtering during the translation.</td>
<td>[tag tag]</td>
</tr>
</tbody>
</table>

Implementation Considerations

Evaluate the following considerations before implementing this feature:

- You can set a Type 7 default route that can be used to reach external destinations. When configured, the router generates a Type 7 default into the NSSA by the NSSA ABR.
- Every router within the same area must agree that the area is NSSA; otherwise, the routers will not be able to communicate with each other.

If possible, avoid using explicit redistribution on NSSA ABR because confusion may result over which packets are being translated by which router.

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 ABR. In OSPF, an ABR 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 ABR to advertise a summary route that covers all the individual networks within the area that fall into the specified range.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify an address range for which a single route will be advertised.</td>
<td>area area-id range address mask</td>
</tr>
</tbody>
</table>

Configure Route Summarization When Redistributing Routes into OSPF

When redistributing routes from other protocols into OSPF (as described in the section “Configure Routing Protocol-Independent Features” later in this chapter), each route is advertised individually in an external link state advertisement (LSA). However, you can configure the Cisco IOS software to advertise a single route for all the redistributed routes that are covered by a specified network address and mask. Doing so helps decrease the size of the OSPF link state database.

To have the software advertise one summary route for all redistributed routes covered by a network address and mask, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify an address and mask that covers redistributed routes, so only one summary route is advertised.</td>
<td>summary-address address mask</td>
</tr>
</tbody>
</table>

Create Virtual Links

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establish a virtual link.</td>
<td>area area-id virtual-link router-id [hello-interval seconds] [retransmit-interval seconds] [transmit-delay seconds] [dead-interval seconds] [[authentication-key key]</td>
</tr>
</tbody>
</table>

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

Generate a Default Route

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force the autonomous system boundary router to generate a default route into the OSPF routing domain.</td>
<td>default-information originate [always] [metric metric-value] [metric-type type-value] [route-map map-name]</td>
</tr>
</tbody>
</table>

See 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 Naming System (DNS) names for use in all OSPF `show` command displays. This feature makes it easier to identify a router, because it is displayed by name rather than by its router ID or neighbor ID.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure DNS name lookup.</td>
<td>ip ospf name-lookup</td>
</tr>
</tbody>
</table>

Force the Router ID Choice with a Loopback Interface

OSPF uses the largest IP address configured on the 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 its routing information out its interfaces.
If a loopback interface is configured with an IP address, the Cisco IOS software will use this IP address as its router ID, even if other interfaces have larger IP addresses. Since loopback interfaces never go down, greater stability in the routing table is achieved.

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Create a loopback interface, which places you in interface configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Assign an IP address to this interface.</td>
</tr>
</tbody>
</table>

1. This command is documented in the “Interface Commands” chapter of the Configuration Fundamentals Command Reference.
2. This command is documented in the “IP Commands” chapter of the Network Protocols Command Reference, Part 1.

Disable Default OSPF Metric Calculation Based on Bandwidth

In Cisco IOS Release 10.2 and earlier, OSPF assigned default OSPF metrics to interfaces regardless of the interface bandwidth. It gave both 64K and T1 links the same metric (1562), and thus required an explicit ip ospf cost command in order to take advantage of the faster link.

In Cisco IOS Release 10.3 and later, by default, OSPF calculates the OSPF metric for an interface according to the bandwidth of the interface. For example, a 64K link gets a metric of 1562, while a T1 link gets a metric of 64. To disable this feature, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable default OSPF metric calculations based on interface bandwidth, resulting in a fixed default metric assignment.</td>
<td>no ospf auto-cost-determination</td>
</tr>
</tbody>
</table>

Configure OSPF on Simplex Ethernet Interfaces

Because simplex interfaces between two devices on an Ethernet represent only one network segment, for OSPF you must configure the transmitting interface to be a passive interface. This prevents OSPF from sending hello packets for the transmitting interface. Both devices 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:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppress the sending of hello packets through the specified interface.</td>
<td>passive-interface type number</td>
</tr>
</tbody>
</table>
Configure OSPF over On Demand Circuits

The OSPF on demand circuit is an enhancement to the OSPF protocol that allows efficient operation over on demand circuits like ISDN, X.25 SVCs and dial-up lines. This feature supports RFC 1793, Extending OSPF to Support Demand Circuits.

Prior to this feature, OSPF periodic hello and link state advertisements (LSAs) updates would be exchanged between routers that connected the on demand link, even when no changes occurred in the hello or LSA information.

With this feature, periodic hellos are suppressed and the periodic refreshes of LSAs are not flooded over the demand circuit. These packets bring up the link only when they are exchanged for the first time, or when a change occurs in the information they contain. This operation allows the underlying datalink layer to be closed when the network topology is stable.

This feature is useful when you want to connect telecommuters or branch offices to an OSPF backbone at a central site. In this case, OSPF for on demand circuits allows the benefits of OSPF over the entire domain, without excess connection costs. Periodic refreshes of hello updates, LSA updates, and other protocol overhead are prevented from enabling the on demand circuit when there is no “real” data to transmit.

Overhead protocols such as hellos and LSAs are transferred over the on demand circuit only upon initial setup and when they reflect a change in the topology. This means that critical changes to the topology that require new SPF calculations are transmitted in order to maintain network topology integrity. Periodic refreshes that do not include changes, however, are not transmitted across the link.

To configure OSPF for on demand circuits, perform the following tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Enable OSPF operation.</td>
<td>router ospf process-id</td>
</tr>
<tr>
<td>Step 2 Configure OSPF on an on demand circuit.</td>
<td>ip ospf demand-circuit</td>
</tr>
</tbody>
</table>

If the router is part of a point-to-point topology, then only one end of the demand circuit must be configured with this command. However, all routers must have this feature loaded.

If the router is part of a point-to-multipoint topology, only the multipoint end must be configured with this command.

Implementation Considerations

Evaluate the following considerations before implementing this feature:

- Because LSAs that include topology changes are flooded over an on demand circuit, it is advised to put demand circuits within OSPF stub areas, or within NSSAs to isolate the demand circuits from as many topology changes as possible.
Configure Stub Routing

A stub router can be thought of as a spoke router in a hub-and-spoke network topology, where the only router to which the spoke is adjacent is the hub router. In such a network topology, the IP routing information required to represent this topology is fairly simple. These stub routers commonly have a WAN connection to the hub router, and a small number of LAN network segments (stub networks) are directly connected to the stub router.

These stub networks might consist only of end systems and the stub router, and thus do not require the stub router to learn any dynamic IP routing information. The stub routers can then be configured with a default route that directs IP traffic to the hub router.

To provide full connectivity, the hub router can be statically configured to know that a particular stub network is reachable via a particular stub router. However, if there are multiple hub routers, many stub networks, or asynchronous connections between hubs and spokes, statically configuring the stub networks on the hub routers becomes a problem.

Stub Routing Task List

Of the following tasks, the first three are required to configure stub routing and the last task is optional:

- Enable On Demand Routing (ODR)
- Filter ODR Information
- Configure Default Route
- Redistribute ODR Information into the Hub’s Dynamic Routing Protocol

Enable On Demand Routing (ODR)

On Demand Routing (ODR) allows you to easily install IP stub networks where the hubs dynamically maintain routes to the stub networks. This is accomplished without requiring the configuration of an IP routing protocol on the stubs.

On stub routers that support the ODR feature, the stub router advertises IP prefixes corresponding to the IP networks configured on all directly connected interfaces. If the interface has multiple logical IP networks configured (via the `ip secondary` command), only the primary IP network is advertised through ODR. Because ODR advertises IP prefixes and not simply IP network numbers, ODR is able to carry Variable Length Subnet Mask (VSLM) information.

To enable ODR, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable ODR on the hub router.</td>
<td><code>router odr process-id</code></td>
</tr>
</tbody>
</table>
Configure Stub Routing

Once ODR is enabled on a hub router, the hub router begins installing stub network routes in the IP forwarding table. The hub router can additionally be configured to redistribute these routes into any configured dynamic IP routing protocols.

On the stub router, no IP routing protocol must be configured. In fact, from the standpoint of ODR, a router is automatically considered to be a stub when no IP routing protocols have been configured.

The routing information that ODR generates is propagated between routers using Cisco’s CDP protocol. This means that the operation of ODR is partially controlled by the configuration of CDP.

Using the global configuration command `no cdp run` disables the propagation of ODR stub routing information entirely. Using the interface configuration command `no cdp enable` disables the propagation of ODR information on a particular interface.

Filter ODR Information

The hub router will attempt to populate the IP routing table with ODR routes, as they are learned dynamically from stub routers. The IP next hop for these routes is the IP address of the neighboring router, as advertised through CDP.

Use IP filtering to limit the network prefixes that the hub router will permit to be learned dynamically through ODR.

To filter ODR information, perform the following task in the router configuration mode:

For example, the following configuration causes the hub router to only accept advertisements for IP prefixes about (or subnets of) the class C network 198.92.110.0.

```
router odr
distribute-list 101 in
access-list 101 permit ip any 198.92.110.0 255.255.255.0
```

Configure Default Route

Although no IP routing protocol must be configured on the stub router, it is still necessary to configure the default route for IP traffic. You can optionally cause traffic for unknown subsets to follow the default route.

To configure the default route for IP traffic, perform the following tasks in global configuration mode:

```
Task: Configure a default route on the stub router.
Command: ip route 0.0.0.0 0.0.0.0 interface-name
```

```
Task: Cause traffic for unknown subnets of directly connected networks to also follow the default route.
Command: ip classless
```

Redistribute ODR Information into the Hub’s Dynamic Routing Protocol

This task may be performed by using the `redistribute` router subcommand. The exact syntax depends upon the routing protocol into which ODR is being redistributed.

See the “Redistribute Routing Information” section later in this chapter.
Reconfigure CDP/ODR Timers

By default, Cisco Discovery Protocol (CDP) sends updates every 60 seconds. This update interval may not be frequent enough to provide speedy reconvergence of IP routes on the hub router side of the network. A faster reconvergence rate may be necessary if the stub connects to one of several hub routers via asynchronous interfaces (such as modem lines). ODR expects to receive periodic CDP updates containing IP prefix information. When ODR fails to receive such updates for routes that it has installed in the routing table, these ODR routes are first marked invalid, and eventually removed from the routing table. (By default, ODR routes are marked invalid after 180 seconds, and are removed from the routing table after 240 seconds.) These defaults are based upon the default CDP update interval. Configuration changes made to either the CDP or ODR timers should be reflected through changes made to both.

To configure CDP/ODR timers, perform the following tasks beginning in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the rate at which CDP updates are</td>
<td>`cdp timer</td>
</tr>
<tr>
<td>sent.</td>
<td><code>seconds</code></td>
</tr>
<tr>
<td>Change the rate at which ODR routes are</td>
<td><code>router odr</code></td>
</tr>
<tr>
<td>expired from the routing table.</td>
<td><code>timers odr</code></td>
</tr>
</tbody>
</table>

1. This command is documented in the “System Management Commands” chapter of the Configuration Fundamentals Command Reference.

Other CDP features are described in the Configuration Fundamentals Configuration Guide, in the “Managing the System” chapter.

Using ODR with Dialer Mappings

For interfaces that specify dialer mappings, CDP packets will make use of dialer map configuration statements that pertain to the IP protocol. Since CDP packets are always broadcast packets, these dialer map statements must handle broadcast packets, typically through use of the dialer map `broadcast` keyword. The dialer string interface configuration command may also be used.

On DDR interfaces, certain kinds of packets can be classified as interesting. These interesting packets can cause a DDR connection to be made, or cause the idle timer of a DDR interface to be reset. For the purposes of DDR classification, CDP packets are considered uninteresting. This is true even while CDP is making use of dialer-map statements for IP, where IP packets are classified as interesting.

Configure RIP

The Routing Information Protocol (RIP) is a relatively old, but still commonly used, IGP created for use in small, homogeneous networks. This is a classical distance-vector routing protocol. RIP is documented in RFC 1058.

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

The metric that RIP uses to rate the value of different routes is `hop count`. The hop count is the number of routers that can be traversed in a route. A directly connected network has a metric of zero; an unreachable network has a metric of 16. This small range of metrics makes RIP an unsuitable routing protocol for large networks.
Configure RIP

If the router has a default network path, RIP advertises a route that links the router to the pseudonetwork 0.0.0.0. The network 0.0.0.0 does not exist; RIP treats 0.0.0.0 as a network to implement the default routing feature. The Cisco IOS software will advertise the default network if a default was learned by RIP, or if the router has a gateway of last resort and RIP is configured with a default metric.

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

Cisco’s implementation of RIP Version 2 supports plain text and MD5 authentication, route summarization, classless interdomain routing (CIDR), and variable-length subnet masks (VLSMs).

RIP Configuration Task List

To configure RIP, complete the tasks in the following sections. You must enable RIP. The remaining tasks are optional.

- Enable RIP
- Allow Point-to-Point Updates for RIP
- Specify a RIP Version
- Enable RIP Authentication
- Disable Route Summarization
- Run IGRP and RIP Concurrently
- Disable the Validation of Source IP Addresses

For information about filtering RIP information, see the “Filter Routing Information” section later in this chapter. For information about RIP Version 2 key management or VLSM, see the “Configure Routing Protocol-Independent Features” section later in this chapter.

Enable RIP

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enable a RIP routing process, which places you in router configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Associate a network with a RIP routing process.</td>
</tr>
</tbody>
</table>

See the “Key Management Examples” section at the end of this chapter for key management examples.

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 Cisco IOS software to permit this exchange of routing information. To do so, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define a neighboring router with which to exchange point-to-point routing information.</td>
<td>neighbor ip-address</td>
</tr>
</tbody>
</table>
To control the set of interfaces with which you want to exchange routing updates, you can disable the sending of routing updates on specified interfaces by configuring the `passive-interface` command. See the discussion on filtering in the “Filter Routing Information” section later in this chapter.

**Specify a RIP Version**

Cisco’s implementation of RIP Version 2 supports authentication, key management, route summarization, classless interdomain routing (CIDR), and variable-length subnet masks (VLSMs). Key management and VLSM are described in the section “Configure Routing Protocol-Independent Features” later in this chapter.

By default, the software receives RIP Version 1 and Version 2 packets, but sends only Version 1 packets. You can configure the software to receive and send only Version 1 packets. Alternatively, you can configure the software to receive and send only Version 2 packets. To do so, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the software to receive and send only RIP Version 1 or only RIP Version 2 packets.</td>
<td>`version {1</td>
</tr>
</tbody>
</table>

The preceding task controls the default behavior of RIP. You can override that behavior by configuring a particular interface to behave differently. To control which RIP version an interface sends, perform one of the following tasks in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure an interface to send only RIP Version 1 packets.</td>
<td><code>ip rip send version 1</code></td>
</tr>
<tr>
<td>Configure an interface to send only RIP Version 2 packets.</td>
<td><code>ip rip send version 2</code></td>
</tr>
<tr>
<td>Configure an interface to send RIP Version 1 and Version 2 packets.</td>
<td><code>ip rip send version 1 2</code></td>
</tr>
</tbody>
</table>

Similarly, to control how packets received from an interface are processed, perform one of the following tasks in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure an interface to accept only RIP Version 1 packets.</td>
<td><code>ip rip receive version 1</code></td>
</tr>
<tr>
<td>Configure an interface to accept only RIP Version 2 packets.</td>
<td><code>ip rip receive version 2</code></td>
</tr>
<tr>
<td>Configure an interface to accept either RIP Version 1 or 2 packets.</td>
<td><code>ip rip receive version 1 2</code></td>
</tr>
</tbody>
</table>

**Enable RIP Authentication**

RIP Version 1 does not support authentication. If you are sending and receiving RIP Version 2 packets, you can enable RIP authentication on an interface.
Configure RIP

The key chain determines the set of keys that can be used on the interface. If a key chain is not configured, no authentication is performed on that interface, not even the default authentication. Therefore, you must also perform the tasks in the section “Manage Authentication Keys” later in this chapter.

We support two modes of authentication on an interface for which RIP authentication is enabled: plain text authentication and MD5 authentication. The default authentication in every RIP Version 2 packet is plain text authentication.

**Note** Do not use plain text authentication in RIP packets for security purposes, because the unencrypted authentication key is sent in every RIP Version 2 packet. Use plain text authentication when security is not an issue, for example, to ensure that misconfigured hosts do not participate in routing.

To configure RIP authentication, perform the following tasks in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enable RIP authentication.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Configure the interface to use MD5 digest authentication (or let it default to plain text authentication).</td>
</tr>
<tr>
<td>Step 3</td>
<td>Perform the authentication key management tasks.</td>
</tr>
</tbody>
</table>

See the “Key Management Examples” section at the end of this chapter for key management examples.

Disable Route Summarization

RIP Version 2 supports automatic route summarization by default. The software summarizes subprefixes to the classful network boundary when crossing classful network boundaries.

If you have disconnected subnets, disable automatic route summarization to advertise the subnets. When route summarization is disabled, the software transmits subnet and host routing information across classful network boundaries. To disable automatic summarization, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable automatic summarization.</td>
<td>no auto-summary</td>
</tr>
</tbody>
</table>

Run IGRP and RIP Concurrently

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

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

Disable the Validation of Source IP Addresses

By default, the software validates the source IP address of incoming RIP routing updates. If that
source address is not valid, the software discards the routing update.

You might want to disable this feature if you have a router that is “off network” and you want to
receive its updates. However, disabling this feature is not recommended under normal
circumstances. To disable the default function that validates the source IP addresses of incoming
routing updates, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable the validation of the source IP address of incoming RIP routing updates.</td>
<td>no validate-update-source</td>
</tr>
</tbody>
</table>

Configure IS-IS

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

IS-IS Configuration Task List

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

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

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

Enable IS-IS

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

Network entity titles (NETs) define the area addresses for the IS-IS area. Multiple NETs per router
are allowed (up to a maximum of three). Refer to the “Configuring ISO CLNS” chapter in Network
To enable IS-IS, perform the following tasks starting in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enable IS-IS routing and specify an IS-IS process for IP, which places you in router configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Configure NETs for the routing process; you can specify a name for a NET as well as an address.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Enter interface configuration mode.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Specify the interfaces that should be actively routing IS-IS.</td>
</tr>
</tbody>
</table>

See the “Static Routing Redistribution Example” section at the end of this chapter for an example of configuring IS-IS as an IP routing protocol.

Configure IS-IS Interface Parameters

Our IS-IS implementation allows you to alter certain interface-specific IS-IS parameters. You can perform these tasks, described in the following sections:

- Configure IS-IS Link-State Metrics
- Set the Advertised Hello Interval
- Set the Advertised CSNP Interval
- Set the Retransmission Interval
- Specify Designated Router Election
- Specify the Interface Circuit Type
- Assign a Password for an Interface

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

Configure IS-IS Link-State Metrics

You can configure a cost for a specified interface. The only configurable metric supported by the Cisco IOS software is the default-metric, which you can configure for Level 1 or Level 2 routing. The other metrics currently are not supported.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the metric (or cost) for the specified interface.</td>
<td>isis metric default-metric [delay-metric [expense-metric [error-metric]]] [level-1</td>
</tr>
</tbody>
</table>

Set the Advertised Hello Interval

You can specify the length of time (in seconds) between hello packets that the Cisco IOS software sends on the interface.
To specify the length of time between hello packets for the specified interface, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the length of time, in seconds, between hello packets the Cisco IOS software sends on the specified interface.</td>
<td>`isis hello-interval seconds [level-1</td>
</tr>
</tbody>
</table>

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

### Set the Advertised CSNP Interval

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the IS-IS CSNP interval for the specified interface.</td>
<td>`isis csnp-interval seconds [level-1</td>
</tr>
</tbody>
</table>

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

### Set the Retransmission Interval

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the number of seconds between retransmission of IS-IS LSPs for point-to-point links.</td>
<td><code>isis retransmit-interval seconds</code></td>
</tr>
</tbody>
</table>

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

### Specify Designated Router Election

You can configure the priority to use for designated router election. Priorities can be configured for Level 1 and Level 2 individually.
To specify the designated router election, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the priority to use for designated router election.</td>
<td>`isis priority value {level-1</td>
</tr>
</tbody>
</table>

Specify the Interface Circuit Type

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the type of adjacency desired for neighbors on the specified interface (the interface circuit type).</td>
<td>`isis circuit-type {level-1</td>
</tr>
</tbody>
</table>

Assign a Password for an Interface

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the authentication password for a specified interface.</td>
<td>`isis password password {level-1</td>
</tr>
</tbody>
</table>

Configure Miscellaneous IS-IS Parameters

You can configure optional IS-IS parameters as described in the following sections:

- Generate a Default Route
- Specify Router-Level Support
- Configure IS-IS Authentication Passwords
- Summarize Address Ranges

Generate a Default Route

You can force a default route into an IS-IS routing domain. Whenever you specifically configure redistribution of routes into an IS-IS routing domain, the Cisco IOS software does not, by default, generate a default route into the IS-IS routing domain. The following feature allows you to force the boundary router do this.
To generate a default route, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force a default route into the IS-IS routing domain.</td>
<td><strong>default-information originate</strong> [metric metric-value] [metric-type type-value] {level-1</td>
</tr>
</tbody>
</table>

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

Specify Router-Level Support

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the level at which the router should operate.</td>
<td><strong>is-type</strong> {level-1</td>
</tr>
</tbody>
</table>

Configure IS-IS Authentication Passwords

You can assign passwords to areas and domains.

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the area authentication password.</td>
<td><strong>area-password</strong> password</td>
</tr>
<tr>
<td>Configure the routing domain authentication password.</td>
<td><strong>domain-password</strong> password</td>
</tr>
</tbody>
</table>

Summarize Address Ranges

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a summary of addresses for a given level.</td>
<td><strong>summary-address</strong> address mask {level-1</td>
</tr>
</tbody>
</table>
Configure BGP

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

Cisco’s BGP Implementation

In BGP, each route consists of a network number, a list of autonomous systems that information has passed through (called the autonomous system path), and a list of other path attributes. We support BGP Versions 2, 3, and 4, as defined in RFCs 1163, 1267, and 1771, respectively.

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

You can configure the value for the Multi Exit Discriminator (MED) metric attribute using route maps. (The name of this metric for BGP Versions 2 and 3 is INTER_AS_METRIC.) When an update is sent to an IBGP peer, the MED is passed along without any change. This action enables all the peers in the same autonomous system to make a consistent path selection.

A third-party next-hop router address is used in the NEXT_HOP attribute, regardless of the autonomous system of that third-party router. The Cisco IOS software automatically calculates the value for this attribute.

Transitive, optional path attributes are passed along to other BGP-speaking routers.

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

See the “BGP Route Map Examples” section at the end of this chapter for examples of how to use route maps to redistribute BGP Version 4 routes.

How BGP Selects Paths

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

• If the next hop is inaccessible, do not consider it.
• Consider larger BGP administrative weights first.
• If the routers have the same weight, consider the route with higher local preference.
• If the routes have the same local preference, prefer the route that the local router originated.
• If no route was originated, prefer the shorter autonomous system path.
• If all paths are of the same autonomous system path length, prefer the lowest origin code (IGP < EGP < INCOMPLETE).
• If origin codes are the same and all the paths are from the same autonomous system, prefer the path with the lowest Multi Exit Discriminator (MED) metric. A missing metric is treated as zero.
• If the MEDs are the same, prefer external paths over internal paths.
• If IGP synchronization is disabled and only internal paths remain, prefer the path through the closest neighbor.

• Prefer the route with the lowest IP address value for the BGP router ID.

When a BGP speaker learns two identical EBGP paths for a prefix from a neighboring AS, it will pick the path with the least route-id as the best path. This best path with the least router-id will be installed in the IP routing table. If BGP multipath support is enabled, instead of picking one best path, if the EBGP paths are learned from the same neighboring AS, multiple paths will be installed in the IP routing table.

During packet switching depending on the mode of switching, either per-packet or per-destination load balancing will be done among the multiple paths. Up to a maximum of 6 paths is supported. The `maximum-paths` router configuration command is used to control the number of paths allowed. By default, BGP will install only one path to the IP routing table.

BGP Configuration Task List

The tasks in this section are divided into basic and advanced BGP configuration tasks. The first three basic tasks are required to configure BGP; the remaining basic and advanced tasks are optional.

Basic BGP Configuration Tasks

Basic BGP configuration tasks are discussed in the following sections:

• Enable BGP Routing
• Configure BGP Neighbors
• Configure BGP Soft Configuration
• Reset BGP Connections
• Configure BGP Interactions with IGPs
• Configure BGP Administrative Weights
• Configure BGP Route Filtering by Neighbor
• Configure BGP Path Filtering by Neighbor
• Disable Next-Hop Processing on BGP Updates
• Configure the BGP Version
• Set the Network Weight
• Configure the Multi Exit Discriminator Metric

Advanced BGP Configuration Tasks

Advanced BGP configuration tasks are discussed in the following sections:

• Use Route Maps to Modify Updates
• Configure BGP Neighbor Templates
• Reset EBGP Connections Immediately upon Link Failure
• Configure Aggregate Addresses
• Disable Automatic Summarization of Network Numbers
Configure BGP

- Configure BGP Community Filtering
- Configure a Routing Domain Confederation
- Configure a Route Reflector
- Configure Neighbor Options
- Configure BGP Peer Groups
- Indicate Backdoor Routes
- Modify Parameters While Updating the IP Routing Table
- Set Administrative Distance
- Adjust BGP Timers
- Change the Local Preference Value
- Redistribute Network 0.0.0.0
- Base Path Selection on MEDs from Other Autonomous Systems

Basic BGP Tasks

This section describes the basic BGP configuration tasks.

Enable BGP Routing

To enable BGP routing, establish a BGP routing process by performing the following steps starting in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enable a BGP routing process, which places you in router configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Flag a network as local to this autonomous system and enter it to the BGP table.</td>
</tr>
</tbody>
</table>

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

**Note** The `network` command is used to inject IGP routes into the BGP table. The `network-mask` portion of the command allows supernetting and subnetting. A maximum of 200 entries of the command are accepted. Alternatively, you could use the `redistribute` command to achieve the same result.
Configure BGP Neighbors

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a BGP neighbor.</td>
<td>`neighbor {ip-address</td>
</tr>
<tr>
<td></td>
<td>remote-as number</td>
</tr>
</tbody>
</table>

See the “BGP Neighbor Configuration Examples” section at the end of this chapter for an example of configuring BGP neighbors.

Configure BGP Soft Configuration

When ever there is a change in the policy, BGP session has to be cleared for the new policy to take effect. Clearing a BGP session causes cache invalidation and results in a tremendous impact on the operation of networks.

Soft-reconfiguration allows policies to be configured and activated without clearing the BGP session. Soft reconfiguration can be done on a per-neighbor basis. When soft reconfiguration is used to generate inbound updates from a neighbor, it is called inbound soft reconfiguration. When soft reconfiguration is used to send a new set of updates to a neighbor, it is called outbound soft reconfiguration.

Performing inbound reconfiguration enables the new inbound policy to take effect. Performing outbound reconfiguration will make the new local outbound policy take effect without resetting the BGP session. As a new set of updates is sent during outbound policy reconfiguration, a new inbound policy of the neighbor can also take effect.

In order to generate new inbound updates without resetting the BGP session, the local BGP speaker should store all the received updates without modification regardless of whether it is accepted or denied by the current inbound policy. This is memory intensive and where ever possible it should be avoided. On the other hand, outbound soft reconfiguration does not have any memory overhead. One could trigger a outbound reconfiguration in the other side of the BGP session to make the new inbound policy take effect.

To allow inbound reconfiguration, BGP should be informed to store all received updates. Outbound reconfiguration does not require pre-configuration.

You can configure the Cisco IOS software to start storing received updates, which is required for in-bound BGP soft reconfiguration. Outbound reconfiguration does not require in-bound soft reconfiguration to be enabled.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a BGP neighbor.</td>
<td>`neighbor {ip-address</td>
</tr>
<tr>
<td></td>
<td>reconfiguration</td>
</tr>
</tbody>
</table>

Our implementation of BGP supports BGP Versions 2, 3, and 4. If the neighbor does not accept default Version 4, dynamic version negotiation is implemented to negotiate down to Version 2.
Configure BGP

If you specify a BGP peer group by using the peer-group-name argument, all members of the peer group will inherit the characteristic configured with this command.

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 must reset BGP connections for the configuration change to take effect. Perform either of the following tasks in EXEC mode to reset BGP connections:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset a particular BGP connection.</td>
<td>clear ip bgp address</td>
</tr>
<tr>
<td>Reset all BGP connections.</td>
<td>clear ip bgp *</td>
</tr>
</tbody>
</table>

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

In some cases, you do not need synchronization. If you will not be passing traffic from a different autonomous system through your autonomous system, or if all routers in your autonomous system will be running BGP, you can disable synchronization. Disabling this feature can allow you to carry fewer routes in your IGP and allow BGP to converge more quickly. To disable synchronization, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable synchronization between BGP and an IGP.</td>
<td>no synchronization</td>
</tr>
</tbody>
</table>

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

See the “BGP Path Filtering by Neighbor Example” section at the end of this chapter for an example of BGP synchronization.

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 speaker generate a default route for your autonomous system. When redistributing from BGP into IGP, only the routes learned using EBGP get redistributed.

In most circumstances, you also will not want to redistribute your IGP into BGP. Just list the networks in your autonomous system with network router configuration commands and your networks will be advertised. Networks that are listed this way are referred to as local networks and have a BGP origin attribute of “IGP.” They must appear in the main IP routing table and can have any source; for example, they can be directly connected or learned via an IGP. The BGP routing process periodically scans the main IP routing table to detect the presence or absence of local networks, updating the BGP routing table as appropriate.
If you do perform redistribution into BGP, you must be very careful about the routes that can be in your IGP, especially if the routes were redistributed from BGP into the IGP elsewhere. This creates a situation where BGP is potentially injecting information into the IGP and then sending such information back into BGP, and vice versa.

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

Configure BGP Administrative Weights

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a weight for all routes from a neighbor.</td>
<td>`neighbor ip-address</td>
</tr>
</tbody>
</table>

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

To assign weights based on autonomous system path access lists, perform the following tasks starting in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Define a BGP-related access list.</td>
<td><code>ip as-path access-list</code> <code>access-list-number</code> `{permit</td>
</tr>
<tr>
<td>Step 2 Enter router configuration mode.</td>
<td><code>router bgp autonomous-system</code></td>
</tr>
<tr>
<td>Step 3 Configure administrative weight on all incoming routes matching an autonomous system path filter.</td>
<td><code>neighbor ip-address filter-list access-list-number weight</code></td>
</tr>
</tbody>
</table>

Configure BGP Route Filtering by Neighbor

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter BGP routing updates to/from neighbors as specified in an access list.</td>
<td>`neighbor ip-address</td>
</tr>
</tbody>
</table>
Configure BGP Path Filtering by Neighbor

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

To configure BGP path filtering, perform the following tasks starting in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Define a BGP-related access list.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Enter router configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Establish a BGP filter.</td>
</tr>
</tbody>
</table>

See the “BGP Path Filtering by Neighbor Example” section at the end of this chapter for an example of BGP path filtering by neighbor.

Disable Next-Hop Processing on BGP Updates

You can configure the Cisco IOS software to disable next-hop processing for BGP updates to a neighbor. This might be useful in nonmeshed networks such as Frame Relay or X.25, where BGP neighbors might not have direct access to all other neighbors on the same IP subnet.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable next-hop processing on BGP updates to a neighbor.</td>
<td>`neighbor {ip-address</td>
</tr>
</tbody>
</table>

Configuring this command causes the current router to advertise itself as the next hop for the specified neighbor. Therefore, other BGP neighbors will forward to it packets for that address. This is useful in a nonmeshed environment, since you know that a path exists from the present router to that address. In a fully meshed environment, this is not useful, since it will result in unnecessary extra hops and because there might be a direct access through the fully meshed cloud with fewer hops.

Configure the BGP Version

By default, BGP sessions begin using BGP Version 4 and negotiating downward to earlier versions if necessary. To prevent negotiation and force the BGP version used to communicate with a neighbor, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the BGP version to use when communicating with a neighbor.</td>
<td>`neighbor {ip-address</td>
</tr>
</tbody>
</table>
Configure BGP

Set the Network Weight

Weight is a parameter that affects the best path selection process. To set the absolute weight for a network, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the weight for a network.</td>
<td><code>network address weight weight</code></td>
</tr>
</tbody>
</table>

Configure the Multi Exit Discriminator Metric

BGP uses the Multi Exit Discriminator (MED) metric as a hint to external neighbors about preferred paths. (The name of this metric for BGP Versions 2 and 3 is INTER_AS_METRIC.) You can set the MED of the redistributed routes by performing the following task. All the routes without a MED will also be set to this value. Perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set a multi exit discriminator.</td>
<td><code>default-metric number</code></td>
</tr>
</tbody>
</table>

Alternatively, you can set the MED using the `route-map` command. See the “BGP Route Map Examples” section at the end of this chapter for examples of using BGP route maps.

Advanced BGP Configuration Tasks

This section contains advanced BGP configuration tasks.

Use Route Maps to Modify Updates

You can use a route map on a per-neighbor basis to filter updates and modify various attributes. A route map can be applied to either inbound or outbound updates. Only the routes that pass the route map are sent or accepted in updates.

On both the inbound and the outbound updates, we support matching based on autonomous system path, community, and network numbers. Autonomous system path matching requires the `as-path access-list` command, community based matching requires the `community-list` command and network-based matching requires the `ip access-list` command. Perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply a route map to incoming or outgoing routes.</td>
<td>`neighbor [ip-address</td>
</tr>
</tbody>
</table>

See the “BGP Route Map Examples” section at the end of this chapter for BGP route-map examples.

Configure BGP Neighbor Templates

You can configure neighbor templates that use a word argument rather than an IP address to configure BGP neighbors. During the initiation of a BGP session, the IP address of the neighbor is checked against the access list, and it must pass the access list to start the BGP session. You must keep in mind that the router configured with the template name is in a passive state for these neighbors, which means it will not initiate sessions; rather it waits until a session is initiated to it, and then verifies based on the criteria previously mentioned.
If you use the `neighbor configure-neighbors` command, all the neighbors that initiate a session to the router and pass the `neighbor-list` are permanently entered into the configuration (NVRAM) upon issuing the `write memory` command. (Note that the `write memory` command has been replaced by the `copy running-config startup-config` command.)

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support anonymous neighbor peers by configuring a neighbor template.</td>
<td>`neighbor template-name neighbor-list [access-list-number</td>
</tr>
<tr>
<td>Treat neighbors that have been accepted by a template as if they were configured by hand.</td>
<td><code>neighbor template-name configure-neighbors</code></td>
</tr>
</tbody>
</table>

See the “Using Access Lists to Specify Neighbors” section at the end of this chapter for an example of using access lists to specify neighbors.

**Reset EBGP Connections Immediately upon Link Failure**

Normally, when a link between external neighbors goes down, the BGP session will not be reset immediately. If you want the EBGP session to be reset as soon as an interface goes down, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically reset EBGP sessions.</td>
<td><code>bgp fast-external-fallover</code></td>
</tr>
</tbody>
</table>

**Configure Aggregate Addresses**

Classless interdomain routing (CIDR) enables you to create aggregate routes (or supernets) to minimize the size of routing tables. You can configure aggregate routes in BGP either by redistributing an aggregate route into BGP or by using the conditional aggregation feature described in the following task table. An aggregate address will be added to the BGP table if there is at least one more specific entry in the BGP table.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an aggregate entry in the BGP routing table.</td>
<td><code>aggregate-address address mask</code></td>
</tr>
<tr>
<td>Generate an aggregate with AS-SET.</td>
<td><code>aggregate-address address mask as-set</code></td>
</tr>
<tr>
<td>Advertise summary addresses only.</td>
<td><code>aggregate-address address-mask summary-only</code></td>
</tr>
<tr>
<td>Suppress selected, more specific routes.</td>
<td><code>aggregate-address address-mask suppress-map map-name</code></td>
</tr>
</tbody>
</table>

See the “BGP Aggregate Route Examples” section at the end of this chapter for examples of using BGP aggregate routes.
Configure BGP

Disable Automatic Summarization of Network Numbers

In BGP Version 3, when a subnet is redistributed from an IGP into BGP, only the network route is injected into the BGP table. By default, this automatic summarization is enabled. To disable automatic network number summarization, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable automatic network summarization.</td>
<td>no auto-summary</td>
</tr>
</tbody>
</table>

Configure BGP Community Filtering

BGP supports transit policies via controlled distribution of routing information. The distribution of routing information is based on one of the following three values:

- IP address (see the “Configure BGP Route Filtering by Neighbor” section earlier in this chapter).
- The value of the AS_PATH attribute (see the “Configure BGP Path Filtering by Neighbor” section earlier in this chapter).
- The value of the COMMUNITIES attribute (as described in this section).

The COMMUNITIES attribute is a way to group destinations into communities and apply routing decisions based on the communities. This method simplifies a BGP speaker’s configuration that controls distribution of routing information.

A community is a group of destinations that share some common attribute. Each destination can belong to multiple communities. Autonomous system administrators can define to which communities a destination belongs. By default, all destinations belong to the general Internet community. The community is carried as the COMMUNITIES attribute.

The COMMUNITIES attribute is an optional, transitive, global attribute in the numerical range from 1 to 4,294,967,200. Along with Internet community, there are a few predefined, well-known communities, as follows:

- **internet**—Advertise this route to the Internet community. All routers belong to it.
- **no-export**—Do not advertise this route to EBGP peers.
- **no-advertise**—Do not advertise this route to any peer (internal or external).

Based on the community, you can control which routing information to accept, prefer, or distribute to other neighbors. A BGP speaker can set, append, or modify the community of a route when you learn, advertise, or redistribute routes. When routes are aggregated, the resulting aggregate has a COMMUNITIES attribute that contains all communities from all the initial routes.

You can use community lists to create groups of communities to use in a match clause of a route map. Just like an access list, a series of community lists can be created. Statements are checked until a match is found. As soon as one statement is satisfied, the test is concluded.

To create a community list, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a community list.</td>
<td>ip community-list community-list-number {permit</td>
</tr>
</tbody>
</table>

To set the COMMUNITIES attribute and match clauses based on communities, see the **match community-list** and **set community** commands in the “Redistribute Routing Information” section later in this chapter.
Configure BGP

By default, no COMMUNITIES attribute is sent to a neighbor. You can specify that the COMMUNITIES attribute be sent to the neighbor at an IP address by performing the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify that the COMMUNITIES attribute be sent to the neighbor at this IP address.</td>
<td>`neighbor {ip-address</td>
</tr>
</tbody>
</table>

Configure a Routing Domain Confederation

One way to reduce the IBGP mesh is to divide an autonomous system into multiple autonomous systems and group them into a single confederation. To the outside world, the confederation looks like a single autonomous system. Each autonomous system is fully meshed within itself, and has a few connections to other autonomous systems in the same confederation. Even though the peers in different autonomous systems have EBGP sessions, they exchange routing information as if they were IBGP peers. Specifically, the next-hop, MED, and local preference information is preserved. This enables us to retain a single Interior Gateway Protocol (IGP) for all of the autonomous systems.

To configure a BGP confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems will look like a single autonomous system with the confederation identifier as the autonomous system number. To configure a BGP confederation identifier, perform the following tasks in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure a BGP confederation.</td>
<td><code>bgp confederation identifier autonomous-system</code></td>
</tr>
</tbody>
</table>

In order to treat the neighbors from other autonomous systems within the confederation as special EBGP peers, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the autonomous systems that belong to the confederation.</td>
<td><code>bgp confederation peers autonomous-system [{autonomous-system ...}]</code></td>
</tr>
</tbody>
</table>

See the “BGP Confederation Example” section at the end of this chapter for an example configuration of several peers in a confederation.

For an alternative way to reduce the IBGP mesh, see the following section, “Configure a Route Reflector,” in this chapter.

Configure a Route Reflector

BGP requires that all of the IBGP speakers be fully meshed. However, this requirement does not scale when there are many IBGP speakers. Instead of configuring a confederation, another way to reduce the IBGP mesh is to configure a route reflector.

Figure 19 illustrates a simple IBGP configuration with three IBGP speakers (Routers A, B, and C). Without route reflectors, when Router A receives a route from an external neighbor, it must advertise it to both Routers B and C. Routers B and C do not readvertise the IBGP learned route to other IBGP speakers because the routers do not pass routes learned from internal neighbors on to other internal neighbors, thus preventing a routing information loop.
With route reflectors, all IBGP speakers need not be fully meshed because there is a method to pass learned routes to neighbors. In this model, an internal BGP peer is configured to be a route reflector responsible for passing IBGP learned routes to a set of IBGP neighbors. In Figure 20, Router B is configured as a route reflector. When the route reflector receives routes advertised from Router A, it advertises them to Router C, and vice versa. This scheme eliminates the need for the IBGP session between Routers A and C.
The internal peers of the route reflector are divided into two groups: client peers and all the other routers in the autonomous system (nonclient peers). A route reflector reflects routes between these two groups. The route reflector and its client peers form a cluster. The nonclient peers must be fully meshed with each other, but the client peers need not be fully meshed. The clients in the cluster do not communicate with IBGP speakers outside their cluster.

Figure 21 illustrates a more complex route reflector scheme. Router A is the route reflector in a cluster with Routers B, C, and D. Routers E, F, and G are fully meshed, nonclient routers.
When the route reflector receives an advertised route, depending on the neighbor, it does the following:

- A route from an external BGP speaker is advertised to all clients and nonclient peers.
- A route from a nonclient peer is advertised to all clients.
- A route from a client is advertised to all clients and nonclient peers. Hence, the clients need not be fully meshed.

To configure a route reflector and its clients, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the local router as a BGP route reflector and the specified neighbor as a client.</td>
<td><code>neighbor ip-address route-reflector-client</code></td>
</tr>
</tbody>
</table>

Along with route reflector-aware BGP speakers, it is possible to have BGP speakers that do not understand the concept of route reflectors. They can be members of either client or nonclient groups. This allows easy, gradual migration from the old BGP model to the route reflector model. Initially, you could create a single cluster with a route reflector and a few clients. All the other IBGP speakers could be nonclient peers to the route reflector and then more clusters could be created gradually.

An autonomous system can have multiple route reflectors. A route reflector treats other route reflectors just like other IBGP speakers. A route reflector can be configured to have other route reflectors in a client group or nonclient group. In a simple configuration, the backbone could be
Configure BGP

divided into many clusters. Each route reflector would be configured with other route reflectors as nonclient peers (thus, all the route reflectors will be fully meshed). The clients are configured to maintain IBGP sessions with only the route reflector in their cluster.

Usually a cluster of clients will have a single route reflector. In that case, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure, a cluster might have more than one route reflector. In this case, all route reflectors in the cluster must be configured with the 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster. All the route reflectors serving a cluster should be fully meshed and all of them should have identical sets of client and nonclient peers.

If the cluster has more than one route reflector, configure the cluster ID by performing the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the cluster ID.</td>
<td>bgp cluster-id cluster-id</td>
</tr>
</tbody>
</table>

Use the `show ip bgp` command to display the originator ID and the cluster-list attributes.

By default, the clients of a route reflector are not required to be fully meshed and the routes from a client are reflected to other clients. However, if the clients are fully meshed, the route reflector does not need to reflect routes to clients. To disable client-to-client route reflection, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable client-to-client route reflection.</td>
<td>no bgp client-to-client reflection</td>
</tr>
</tbody>
</table>

**Note** If client-to-client reflection is enabled, the clients of a route reflector cannot be members of a peer group.

As the IBGP learned routes are reflected, it is possible for routing information to loop. The route reflector model has the following mechanisms to avoid routing loops:

- **Originator-ID** is an optional, nontransitive BGP attribute. This is a 4-byte attributed created by a route reflector. The attribute carries the router ID of the originator of the route in the local autonomous system. Therefore, if a misconfiguration causes routing information to come back to the originator, the information is ignored.

- **Cluster-list** is an optional, nontransitive BGP attribute. It is a sequence of cluster IDs that the route has passed. When a route reflector reflects a route from its clients to nonclient peers, it appends the local cluster ID to the cluster-list. If the cluster-list is empty, it creates a new one. Using this attribute, a route reflector can identify if routing information is looped back to the same cluster due to misconfiguration. If the local cluster ID is found in the cluster-list, the advertisement is ignored.

- Using `set` clauses in outbound route maps modifies attributes, possibly creating routing loops. To avoid this, `set` clauses of outbound route maps are ignored for routes reflected to IBGP peers.

Configure Neighbor Options

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

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

For internal BGP, you might want to allow your BGP connections to stay up regardless of which
interface is used to reach a neighbor. 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. Now the IBGP session will be
up as long as there is a route, regardless of any interface.

You can set the minimum interval of time between BGP routing updates.

You can invoke MD5 authentication between two BGP peers, meaning that each segment sent on the
TCP connection between them is verified. This feature must be configured with the same password
on both BGP peers; otherwise, the connection between them will not be made. The authentication
feature uses the MD5 algorithm. Invoking authentication causes the Cisco IOS software to generate
and check the MD5 digest of every segment sent on the TCP connection. If authentication is invoked
and a segment fails authentication, then a message appears on the console.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify an access list of BGP neighbors.</td>
<td>`neighbor any [access-list-number</td>
</tr>
<tr>
<td>Allow internal BGP sessions to use any operational interface for TCP connections.</td>
<td>`neighbor {ip-address</td>
</tr>
<tr>
<td>Allow BGP sessions even when the neighbor is not on a directly connected segment.</td>
<td>`neighbor {ip-address</td>
</tr>
<tr>
<td>Set the minimum interval between sending BGP routing updates.</td>
<td>`neighbor {ip-address</td>
</tr>
<tr>
<td>Invoke MD5 authentication on a TCP connection to a BGP peer.</td>
<td>`neighbor {ip-address</td>
</tr>
</tbody>
</table>

**Configure BGP Peer Groups**

Often, in a BGP speaker, many neighbors are configured with the same update policies (that is, the
same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the
same update policies can be grouped into peer groups to simplify configuration and make updating
more efficient.

The three steps to configuring a BGP peer group are: creating the peer group, assigning configuration
options to the peer group, and making neighbors members of the peer group.

To create a BGP peer group, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a BGP peer group.</td>
<td><code>neighbor peer-group-name peer-group</code></td>
</tr>
</tbody>
</table>

After you create a peer group, you configure the peer group with `neighbor` commands. By default,
members of the peer group inherit all the configuration options of the peer group. Members can also
be configured to override the options that do not affect outbound updates.
Peer group members will always inherit the following: remote-as (if configured), version, update-source, out-route-map, out-filter-list, out-dist-list, minimum-advertisement-interval, and next-hop-self. All the peer group members will inherit changes made to the peer group.

To assign configuration options to the peer group, perform any of the following tasks in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify that the COMMUNITIES attribute be sent to the neighbor at this IP address.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Allow internal BGP sessions to use any operational interface for TCP connections.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Allow BGP sessions, even when the neighbor is not on a directly connected segment.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Set the minimum interval between sending BGP routing updates.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Invoke MD5 authentication on a TCP connection to a BGP peer.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Specify a BGP neighbor.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Specify a weight for all routes from a neighbor.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Filter BGP routing updates to/from neighbors, as specified in an access list.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Establish a BGP filter.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Disable next-hop processing on the BGP updates to a neighbor.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Specify the BGP version to use when communicating with a neighbor.</td>
<td>neighbor {ip-address</td>
</tr>
<tr>
<td>Apply a route map to incoming or outgoing routes.</td>
<td>neighbor {ip-address</td>
</tr>
</tbody>
</table>

If a peer group is not configured with a remote-as, the members can be configured with the neighbor ip-address | peer-group-name remote-as command. This allows you to create peer groups containing EBGP neighbors.

Finally, to configure a BGP neighbor to be a member of that BGP peer group, perform the following task in router configuration mode, using the same peer group name:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make a BGP neighbor a member of the peer group.</td>
<td>neighbor ip-address peer-group peer-group-name</td>
</tr>
</tbody>
</table>

See the “BGP Peer Group Examples” section at the end of this chapter for examples of IBGP and EBGP peer groups.
Indicate Backdoor Routes

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indicate reachable networks through backdoor routes.</td>
<td>network address backdoor</td>
</tr>
</tbody>
</table>

Modify Parameters While Updating the IP Routing Table

By default, when a BGP route is put into the IP routing table, the MED is converted to an IP route metric, the BGP next hop is used as the next hop for the IP route, and the tag is not set. However, you can use a route map to perform mapping. To modify metric and tag information when the IP routing table is updated with BGP learned routes, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply a route map to routes when updating the IP routing table.</td>
<td>table-map route-map name</td>
</tr>
</tbody>
</table>

Set Administrative Distance

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign a BGP administrative distance.</td>
<td>distance bgp external-distance internal-distance local-distance</td>
</tr>
</tbody>
</table>

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 Cisco IOS software 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 hold time and the configured keepalive time. To adjust BGP timers, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjust BGP timers.</td>
<td>timers bgp keepalive holdtime</td>
</tr>
</tbody>
</table>
Configure EGP

Change the Local Preference Value

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the default local preference value.</td>
<td>bgp default local-preference value</td>
</tr>
</tbody>
</table>

You can use route maps to change the default local preference of specific paths. See the “BGP Route Map Examples” section at the end of this chapter for examples when used with BGP route maps.

Redistribute Network 0.0.0.0

By default, you are not allowed to redistribute network 0.0.0.0. To permit the redistribution of network 0.0.0.0, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow the redistribution of network 0.0.0.0 into BGP</td>
<td>default-information originate</td>
</tr>
</tbody>
</table>

Base Path Selection on MEDs from Other Autonomous Systems

The MED is one of the parameters that is considered when selecting the best path among many alternative paths. The path with a lower MED is preferred over a path with a higher MED.

By default, during the best-path selection process, MED comparison is done only among paths from the same autonomous system. You can allow comparison of MEDs among paths regardless of the autonomous system from which the paths are received. To do so, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow the comparison of MEDs for paths from neighbors in different autonomous systems</td>
<td>bgp always-compare-med</td>
</tr>
</tbody>
</table>

Configure EGP

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

An exterior router uses EGP to advertise its knowledge of routes to networks within its autonomous system. It sends these advertisements to the core routers, which then readvertise their collected routing information to the exterior router. A neighbor or peer router is any router with which the router communicates using EGP.
Cisco’s EGP Implementation
Cisco’s implementation of EGP supports three primary functions, as specified in RFC 904:
• Routers running EGP establish a set of neighbors, and these neighbors share reachability information.
• EGP routers poll their neighbors periodically to see if they are “alive.”
• EGP routers send update messages containing information about the reachability of networks within their autonomous systems.

EGP Configuration Task List
To enable EGP routing on your router, complete the tasks in the following sections. The tasks in the first two sections are mandatory; the remaining tasks are optional.
• Enable EGP Routing
• Configure EGP Neighbor Relationships
• Adjust EGP Timers
• Configure Third-Party EGP Support
• Configure Backup Routers
• Configure Default Routes
• Define a Central Routing Information Manager (Core Gateway)

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

To enable EGP routing, perform the following tasks starting in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Specify the autonomous system in which the router resides for EGP.</td>
<td>autonomous-system local-as</td>
</tr>
<tr>
<td>Step 2 Enable an EGP routing process, which places you in router configuration mode.</td>
<td>router egp remote-as</td>
</tr>
<tr>
<td>Step 3 Specify a network to be advertised to the EGP peers of an EGP routing process.</td>
<td>network network-number</td>
</tr>
</tbody>
</table>

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

Configure EGP Neighbor Relationships
A router using EGP cannot dynamically determine its neighbor or peer routers. You must, therefore, provide a list of neighbor routers.
Configure EGP

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify an EGP neighbor.</td>
<td>neighbor ip-address</td>
</tr>
</tbody>
</table>

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 Cisco IOS software sends hello messages to its peer. The poll time specifies how frequently to exchange updates. Our implementation of EGP allows these timers to be adjusted by the user.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjust EGP timers.</td>
<td>timers egp hello polltime</td>
</tr>
</tbody>
</table>

Configure Third-Party EGP Support

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a third-party through which certain destinations can be achieved.</td>
<td>neighbor ip-address third-party third-party-ip-address [internal</td>
</tr>
</tbody>
</table>

See the “Third-Party EGP Support Example” section at the end of this chapter for an example of configuring third-party EGP support.

Configure Backup Routers

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

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

See the “Backup EGP Router Example” section at the end of this chapter for an example of configuring backup EGP routers.
Configure Default Routes

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure EGP to generate a default route.</td>
<td>default-information originate</td>
</tr>
</tbody>
</table>

Define a Central Routing Information Manager (Core Gateway)

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

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

To define a core gateway process, perform the following tasks starting in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Allow a specific router to act as a peer with any reachable autonomous system.</td>
<td>router egp 0</td>
</tr>
<tr>
<td>Step 2 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.</td>
<td>neighbor any [access-list-number</td>
</tr>
</tbody>
</table>

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

See the “EGP Core Gateway Example” section at the end of this chapter for an example of configuring an EGP core gateway.

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

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

**Note** In future Cisco IOS software releases, GDP will not be supported.

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

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

Figure 22 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 22 GDP Report Message Packet Format**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>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.</td>
</tr>
<tr>
<td>Opcode</td>
<td>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.</td>
</tr>
<tr>
<td>Count</td>
<td>Number of addresses, priorities, and hold times.</td>
</tr>
<tr>
<td>Reserved</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>Address</td>
<td>IP address of the router sending the report.</td>
</tr>
<tr>
<td>Priority</td>
<td>Priority of the router sending the report.</td>
</tr>
<tr>
<td>Hold time</td>
<td>Hold time of the router sending the report.</td>
</tr>
<tr>
<td></td>
<td>Address / priority / Hold time fields repeated count times</td>
</tr>
</tbody>
</table>
Configure IRDP

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable GDP processing on an interface.</td>
<td>ip gdp</td>
</tr>
<tr>
<td>Set the relative quality of the associated address.</td>
<td>ip gdp priority number</td>
</tr>
<tr>
<td>Set the GDP report period.</td>
<td>ip gdp reporttime seconds</td>
</tr>
<tr>
<td>Set the length of time the associated address should be used as a router without hearing further report messages regarding that address.</td>
<td>ip gdp holdtime seconds</td>
</tr>
</tbody>
</table>

Configure IRDP

Like GDP, the ICMP Router Discovery Protocol (IRDP) allows hosts to locate routers. When operating as a client, router discovery packets are generated. When operating as a host, router discovery packets are received.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable IRDP processing on an interface.</td>
<td>ip irdp</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send IRDP advertisements to the all-systems multicast address (224.0.0.1) on a specified interface.</td>
<td>ip irdp multicast</td>
</tr>
<tr>
<td>Set the IRDP period for which advertisements are valid.</td>
<td>ip irdp holdtime seconds</td>
</tr>
</tbody>
</table>
Configure Resource Reservation Protocol (RSVP)

The Resource Reservation Protocol (RSVP) permits end systems to request Quality of Service (QOS) guarantees from the network. The need for network resource reservations differs as follows for data traffic versus for real-time traffic:

- Data traffic seldom needs reserved bandwidth since internetworks provide datagram services for data traffic. This asynchronous packet switching does not need guarantees of service quality. Routers can operate in a first-in, first out (FIFO) manner for data traffic packets. End-to-end controls between data traffic senders and receivers help ensure adequate transmission of bursts of information.

- Real-time traffic (that is, voice or video information) experiences problems when operating over datagram services. Since real-time traffic sends an almost constant flow of information that send and receive, the network “pipes” must be consistent. Some guarantee must be provided that service between real-time hosts will not vary. Routers operating on a FIFO basis risk unrecoverable disruption of the real-time information that is being transmitted.

Data applications (with little need for resource guarantees) frequently demand relatively lower bandwidth than real-time traffic. The almost constant high bit-rate demands of a video conferencing application, and the bursty low bit-rate demands of an interactive data application, share available network resources.

RSVP prevents the demands of real-time traffic from impairing the bandwidth resources necessary for bursty data traffic. To do this, the routers sort and prioritize packets much like a statistical time division multiplexor would sort and prioritize several signal sources that shares a single channel.

RSVP mechanisms enable real-time traffic to reserve resources necessary for consistent latency. A video conferencing application can use settings in the router to propagate a request for a path with the required bandwidth and delay for video conferencing destinations. RSVP will check and repeat reservations at regular intervals. By this process, RSVP can adjust and alter the path between RSVP end systems to recover from router changes.

Real-time traffic (unlike data traffic) requires a guaranteed network consistency. Without consistent QOS, real-time traffic faces the following problems:

- Jitter—A slight time or phase movement in a transmission signal can introduce loss of synchronization or other errors.

- Insufficient bandwidth—Voice calls use a digital signal level 0 (DS0 at 64Kbps); video conferencing uses T1/E1 (1.544 Mbs or 2.048 Mbps); and higher-fidelity video uses much more.
• Delay variations—If the wait time between when signal elements are sent and when they arrive varies, the real-time traffic will no longer be synchronized and may fail.

• Information loss—When signal elements drop or arrive too late lost audio causes distortions with noise or crackle sounds. The lost video causes image blurring, distortions, or blackouts.

RSVP works in conjunction with weighted fair queuing (WFQ). This conjunction of reservation setting with packet queuing uses two key concepts: end-to-end flows with RSVP and router-to-router conversations with WFQ.

• RSVP Flow—This is a stream that operates “multidestination simplex,” since data travels across it in only one direction (from the origin to the targets). Flows travel from a set of senders to a set of receivers. The flows can be merged or left unmerged, and the method of merging them varies according to the attributes of application using the flow.

• WFQ Conversation—This is the traffic for a single transport layer session or network layer flow that crosses a given interface. This conversation is called from the source and destination address, protocol type, port number, or other attributes in the relevant communications layer.

RSVP allows for hosts to send packets to a subset of all hosts (or multicasting). RSVP assumes that resource reservation applies primarily to multicast applications (such as video conferencing).

Although the primary target for RSVP is multimedia traffic, a clear interest exists for the reservation of bandwidth for unicast traffic (such as NFS and virtual private network management). A unicast transmission involves a host sending packets to a single host.

RSVP Reservation Types

Two types of multicast flows are a flow that originates from exactly one sender (called a distinct reservation), and a flow that originates from one or more senders (called a shared reservation).

RSVP describes these reservations as having certain algorithmic attributes.

An example of a distinct reservation is a video application, in which each sender emits a distinct data stream that requires admission and management in a queue. Such a flow, therefore, requires a separate reservation per sender on each transmission facility it crosses (such as Ethernet, an HDLC line, a Frame Relay DLCI, or an ATM virtual channel). RSVP refers to this distinct reservation as explicit, and installs it using a Fixed Filter style of reservation.

Use of RSVP for unicast applications is generally a degenerate case of a distinct flow.

An example of a shared reservation is an audio application, in which each sender also emits a distinct data stream that requires admission and management in a queue. However, because of the nature of the application, a limited number of senders are transmitting data at any given time. Such a flow, therefore, does not require a separate reservation per sender. Instead, a single reservation that can be applied to any sender within a set, as needed.

RSVP installs a shared reservation using a Wild Card or Shared Explicit style of reservation, with the difference between the two being determined by the scope of application (which is either wild or explicit). The Wild Card Filter reserves bandwidth and delay characteristics for any sender, and is limited by the list of source addresses carried in the reservation message. The Shared Explicit reservation style identifies the flows for specific network resources.
Planning for RSVP Configuration

You must plan carefully to successfully configure and use RSVP on your network. At a minimum, RSVP must reflect your assessment of bandwidth needs on router interfaces. Consider the following questions as you plan for RSVP configuration:

- How much bandwidth should RSVP allow per end-user application flow? You must understand the “feeds and speeds” of your applications. By default, the amount reservable by a single flow can be the entire reservable bandwidth. You can, however, limit individual reservations to smaller amounts using the single flow bandwidth parameter. This value may not exceed the interface reservable amount, and no one flow may reserve more than the amount specified.

- How much bandwidth is available for RSVP? By default, 75 percent of the bandwidth available on an interface is reservable. If you are using a tunnel interface, RSVP can make a reservation for the tunnel whose bandwidth is the sum of the bandwidths reserved within the tunnel.

- How much bandwidth must be excluded from RSVP so that it can fairly provide the timely service required by low-volume data conversations? End-to-end controls for data traffic assumes that all sessions will behave so as to avoid congestion dynamically. Real-time demands do not follow this behavior. Determine the bandwidth to set aside so bursty data traffic will not be deprived as a side effect of the RSVP QOS configuration.

Plan for RSVP before entering the details needed as RSVP configuration parameters.

RSVP Task List

After you have planned for your RSVP configuration, continue by entering the Cisco IOS commands that will implement your reservation configuration plan. The following sections discuss how to configure RSVP. You must enable RSVP on an interface, and enter the senders and receivers into the RSVP database in global configuration mode. The other tasks are optional.

- Enable RSVP on an Interface
- Enter Multicast Addresses
- Set Up Access-List Controls
- Enter Senders in the RSVP Database
- Enter Receivers in the RSVP Database
- Monitor RSVP

Enable RSVP on an Interface

The interface configuration command `ip rsvp bandwidth` enables the use of the RSVP protocol for IP on an interface.

To enable RSVP on an interface, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable RSVP for IP on an interface.</td>
<td><code>ip rsvp bandwidth [interface-kbps] [single-flow-kbps]</code></td>
</tr>
</tbody>
</table>

You must use the `ip rsvp bandwidth` command because the default is for the protocol to be disabled; this is backward compatible with systems that do not implement RSVP.
This command starts RSVP and sets the bandwidth and single-flow limits. The default maximum bandwidth is up to 75 percent of the bandwidth available on the interface. By default, the amount reservable by a flow can be up to the entire reservable bandwidth.

On subinterfaces, this applies the more restrictive of the available bandwidths of the physical interface and the subinterface. For example, a Frame Relay interface might have a T1 connector nominally capable of 1.536 Mbps, and 64 sub-interfaces on 128 Kbps circuits (64K CIR), with 1200 and 100 Kbps, respectively.

Reservations on individual circuits that do not exceed 100 Kbps normally succeed. If, however, reservations have been made on other circuits adding up to 1.2 Mbps, and a reservation is made on a subinterface which itself has enough remaining bandwidth, it will still be refused because the physical interface lacks supporting bandwidth.

Enter Multicast Addresses

If RSVP neighbors are discovered to be using UDP encapsulation, the router will automatically generate UDP encapsulated messages for consumption by the neighbors. This step is optional.

To enter multicast addresses, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter any multicast addresses necessary if you use UDP.</td>
<td><code>ip rsvp udp-multicast [multicast-address]</code></td>
</tr>
</tbody>
</table>

However, in some cases, a host will not originate such a message until it has first heard from the router, which it can only do via UDP. You must instruct the router to generate UDP encapsulated RSVP multicasts whenever it generates an IP encapsulated multicast.

Set Up Access-List Controls

If no limits are specified, any RSVP neighbor may offer a reservation.

To set up access-list controls, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set up any access-list controls to limit which routers may offer reservations.</td>
<td><code>ip rsvp neighbors access-list-number</code></td>
</tr>
</tbody>
</table>

If an access list is specified, only neighbors conforming to the access list are accepted. This access list is applied to the IP header.

Enter Senders in the RSVP Database

This interface configuration command forces the router to behave as though it is periodically receiving an RSVP PATH message from the sender or previous hop routes containing the indicated attributes.
Configure Resource Reservation Protocol (RSVP)

To enter senders in the RSVP database, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter the senders in the RSVP database.</td>
<td>ip rsvp sender session-ip-address sender-ip-address [UDP</td>
</tr>
</tbody>
</table>

Enter Receivers in the RSVP Database

This interface configuration command forces the router to behave as though it is continuously receiving an RSVP RESV message from the originator containing the indicated attributes.

To enter receivers in the RSVP database, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter the receivers in the RSVP database.</td>
<td>ip rsvp reservation session-ip-address [TCP</td>
</tr>
</tbody>
</table>

Monitor RSVP

After you configure the RSVP reservations that reflect your network resource policy, you can verify the resulting RSVP operations.

To verify RSVP operations, perform the following tasks in EXEC mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display RSVP-related interface information.</td>
<td>show ip rsvp interface [interface]</td>
</tr>
<tr>
<td>Display RSVP-related filters and bandwidth information.</td>
<td>show ip rsvp interface installed [interface]</td>
</tr>
<tr>
<td>Display current RSVP neighbors.</td>
<td>show ip rsvp neighbor [interface]</td>
</tr>
<tr>
<td>Display RSVP sender information.</td>
<td>show ip rsvp sender [interface]</td>
</tr>
<tr>
<td>Display RSVP request information.</td>
<td>show ip rsvp request [interface]</td>
</tr>
<tr>
<td>Display RSVP receiver information.</td>
<td>show ip rsvp reservation [interface]</td>
</tr>
</tbody>
</table>

RSVP Implementation Considerations

You should be aware of RSVP implementation considerations as you design your reservation system. RSVP does not model all data links likely to be present on the internetwork. RSVP models an interface as having a queuing system that completely determines the mix of traffic on the interface; bandwidth or delay characteristics are only deterministic to the extent that this model holds.

Unfortunately, data links are often imperfectly modeled this way. Use the following guidelines:

- Serial line interfaces—PPP, HDLC, LAPB, HSSI, and similar serial line interfaces are well modeled by RSVP. The device can, therefore, make guarantees on these interfaces. With NBMA interfaces, these are also the most in need of reservations.
Configure IP Multicast Routing

Multicast LANs—These data links are not modeled well by RSVP interfaces, because the LAN itself represents a queuing system that is not under the control of the device making the guarantees. The device guarantees what load it will offer, but cannot guarantee what competing loads or timings of loads that neighboring LAN systems will offer. The network administrator can use admission controls to control how much traffic is placed on the LAN. The network administrator, however, should focus on the use of admission in network design in order to use RSVP effectively.

Public X.25 networks—It is not clear that rate or delay reservations can be usefully made on public X.25 networks.

You must use a specialized configuration on Frame Relay and ATM networks.

The following RSVP implementation considerations apply as you design your reservation system for a Frame Relay internetwork:

- Reservations are made for an interface or subinterface. If subinterfaces contain more than one DLC, the bandwidth required and the bandwidth reserved may differ. Therefore, RSVP subinterfaces of frame relay circuits must contain exactly one DLC to operate correctly.
- In addition, Frame Relay DLCs have rates (CIR) and burst controls (Be and Be) that may not be reflected in the configuration, and may differ markedly from the interface speed (either adding up to exceed it or being significantly smaller). Therefore, the `ip rsvp bandwidth` interface configuration command must be entered for both the interface and the subinterface. Both bandwidths are used as admission criteria.

For example, suppose that a frame relay interface runs at a T1 rate (1.544 Mbps) and supports several DLCs to remote offices served by 128 and 56 Kbps lines. One must configure the amount of the total interface (75 percent of which being 1.158 Mbps) and the amount of each receiving interface (75 percent of which would be 96 and 42 Kbps, respectively) that may be reserved. Admission succeeds if and only if enough bandwidth is available on the DLC (the subinterface) and on the aggregate interface.

The following RSVP implementation considerations apply as you design your reservation system for an ATM internetwork:

- When ATM is configured, it most likely uses a usable bit rate (UBR) or an available bit rate (ABR) virtual channel (VC) connecting individual routers. With these classes of service, the ATM network makes a “best effort” to meet the traffic’s bit-rate requirements, and assumes that the end-stations are responsible for information that does not get through the network.
- This ATM service has the capability of opening separate channels for reserved traffic having the necessary characteristics. RSVP should open these VCs and adjust the cache to make effective use of the VC for this purpose.

Configure IP Multicast Routing

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

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

The multicast environment consists of senders and receivers. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

A multicast address is chosen for the receivers in a multicast group. Senders use that address as the destination address of a datagram to reach all members of the group.
Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.

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

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

Cisco’s Implementation of IP Multicast Routing

The Cisco IOS software supports three protocols to implement IP multicast routing:

- Internet Group Management Protocol (IGMP) is used between hosts on a LAN and the router(s) on that LAN to track of which multicast groups the hosts are members.
- Protocol-Independent Multicast (PIM) is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- Distance Vector Multicast Routing Protocol (DVMRP) is the protocol used on the MBONE (the multicast backbone of the Internet. The Cisco IOS software supports PIM-to-DVMRP interaction.

Internet Group Management Protocol (IGMP)

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

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

Protocol-Independent Multicast (PIM) Protocol

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

PIM is defined in the following IETF Internet drafts:

- Protocol Independent Multicast (PIM): Motivation and Architecture
- Protocol Independent Multicast (PIM), Dense Mode Protocol Specification
- Protocol Independent Multicast (PIM), Sparse Mode Protocol Specification
- IGMP Router Extensions for Routing to Dense Multicast Groups
- IGMP Router Extensions for Routing to Sparse Multicast Groups

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

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

Distance Vector Multicast Routing Protocol (DVMRP)
DVMRP uses a flood-and-prune approach to multicast packet delivery. It assumes that all other routers want to forward multicast packets for a group, like dense-mode PIM. DVMRP is implemented in the equipment of many vendors and is based on the public domain mrouted program.

The Cisco IOS software supports dynamic discovery of DVMRP routers and can interoperate with them over traditional media (such as Ethernet and FDDI), or over DVMRP-specific tunnels.

IP Multicast Routing Configuration Task List
To configure IP multicast routing, perform the required tasks described in the following sections:
- Enable IP Multicast Routing
- Enable PIM on an Interface
You can also perform the optional tasks described in the following sections:
- Configure a Router to Be a Member of a Group
- Modify the IGMP Host-Query Message Interval
- Control Access to IP Multicast Groups
- Modify the PIM Router-Query Message Interval
- Configure the TTL Threshold
- Configure DVMRP Interoperability
- Advertise Network 0.0.0.0 to DVMRP Neighbors
- Configure a DVMRP Tunnel
- Configure an IP Multicast Static Route
- Disable Fast Switching of IP Multicast
- Control the Transmission Rate to a Multicast Group
- Enable PIM Nonbroadcast, Multiaccess (NBMA) Mode
- Enable sd Listener Support

See the “IP Multicast Routing Configuration Examples” section at the end of this chapter for examples of multicast routing configurations.
Enable IP Multicast Routing

Enabling IP multicast routing allows the Cisco IOS software to forward multicast packets. To enable IP multicast routing on the router, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable IP multicast routing.</td>
<td>ip multicast-routing</td>
</tr>
</tbody>
</table>

Enable PIM on an Interface

Enabling PIM on an interface also enables IGMP operation on that interface. An interface can be configured to be in dense mode or sparse mode. The mode describes how the router populates its multicast routing table and how the router forwards multicast packets it receives from its directly connected LANs.

In populating the multicast routing table, dense-mode interfaces are always added to the table. Sparse-mode interfaces are added to the table only when periodic join messages are received from downstream routers, or when there is a directly connected member on the interface. When forwarding from a LAN, sparse-mode operation occurs if there is an RP known for the group. If so, the packets are encapsulated and sent toward the RP. When no RP is known, the packet is flooded in a dense-mode fashion. If the multicast traffic from a specific source is sufficient, the receiver’s first-hop router may send joins toward the source to build a source-based distribution tree.

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable dense-mode PIM on the interface.</td>
<td>ip pim dense-mode</td>
</tr>
</tbody>
</table>

See the “Configure to Operate in Dense Mode Example” section later in this chapter for an example of how to configure a PIM interface in dense mode.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable sparse-mode PIM on the interface.</td>
<td>ip pim sparse-mode</td>
</tr>
</tbody>
</table>

See the “Configure to Operate in Sparse Mode Example” section later in this chapter for an example of how to configure a PIM interface in sparse mode.

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

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

The RP address is used by first-hop routers to send PIM register messages on behalf of a host sending a packet to the group. The RP address is also used by last-hop routers to send PIM join/prune messages to the RP to inform it about group membership. The RP does not need to know it is an RP. You must configure the RP address only on first-hop and last-hop routers (leaf routers).
A PIM router can be an RP for more than one group. A group can have more than one RP. The conditions specified by the access list determine for which groups the router is an RP.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the address of a PIM rendezvous</td>
<td>ip pim rp-address ip-address [access-list-number]</td>
</tr>
</tbody>
</table>

Configure a Router to Be a Member of a Group

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

To have a router join a multicast group and turn on IGMP, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Join a multicast group.</td>
<td>ip igmp join-group group-address</td>
</tr>
</tbody>
</table>

Modify the IGMP Host-Query Message Interval

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

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

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the frequency at which</td>
<td>ip igmp query-interval</td>
</tr>
<tr>
<td>the designated router sends IGMP</td>
<td>seconds</td>
</tr>
<tr>
<td>host-query messages.</td>
<td></td>
</tr>
</tbody>
</table>

Control Access to IP Multicast Groups

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control the multicast groups that hosts on the subnet serviced by an interface can join.</td>
<td><code>ip igmp access-group access-list-number</code></td>
</tr>
</tbody>
</table>

Modify the PIM Router-Query Message Interval

Route-query messages are used to elect a PIM designated router. The designated router is responsible for sending IGMP host-query messages. By default, multicast routers send PIM router-query messages every 30 seconds. To modify this interval, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the frequency at which multicast routers send PIM router-query messages.</td>
<td><code>ip pim query-interval seconds</code></td>
</tr>
</tbody>
</table>

Configure the TTL Threshold

The time-to-live (TTL) value controls whether packets are forwarded out of an interface. You specify the TTL value in hops. Only multicast packets with a TTL greater than the interface TTL threshold are forwarded on the interface. The default value is 0, which means that all multicast packets are forwarded on the interface. To change the default TTL threshold value, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the TTL threshold of packets being forwarded out an interface.</td>
<td><code>ip multicast ttl-threshold ttl</code></td>
</tr>
</tbody>
</table>

Configure DVMRP Interoperability

Cisco multicast routers using PIM can interoperate with non-Cisco multicast routers that use the Distance Vector Multicast Routing Protocol (DVMRP).

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

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

The mrouted protocol is a public domain implementation of DVMRP. It is necessary to use mrouted version 3.8 (which implements a nonpruning version of DVMRP). When Cisco routers are directly connected to DVMRP routers or interoperate with DVMRP routers over an MBONE tunnel, DVMRP advertisements produced by the Cisco IOS software can cause older versions of mrouted to corrupt their routing tables and those of their neighbors. Any router connected to the MBONE should have an access-list to limit the number of unicast routes that are advertised via DVMRP.
To configure the sources that are advertised and the metrics that are used when transmitting DVMRP report messages, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the metric associated with a set of destinations for DVMRP reports.</td>
<td>ip dvmrp metric metric [list access-list-number] [protocol process-id] [dvmrp]</td>
</tr>
</tbody>
</table>

Responding to MRINFO Requests

The Cisco IOS software answers mrinfo requests sent by mrouted systems and cisco routers. The software returns information about neighbors on DVMRP tunnels and all of the router’s interfaces. This information includes the metric, which is always set to 1, the configured TTL threshold, the status of the interface, and various flags. The mrinfo command can also be used to query the router itself as in the following example:

```
mml-7kd# mrinfo
171.69.214.27 (mml-7kd.cisco.com) [version cisco 11.1] [flags: PMS]:
171.69.214.27 -> 171.69.214.26 (mml-r7kb.cisco.com) [1/0/pim/querier]
171.69.214.27 -> 171.69.214.25 (mml-45a.cisco.com) [1/0/pim/querier]
171.69.214.33 -> 171.69.214.34 (mml-45c.cisco.com) [1/0/pim]
171.69.214.137 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
171.69.214.203 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
171.69.214.18 -> 171.69.214.20 (mml-45e.cisco.com) [1/0/pim]
171.69.214.18 -> 171.69.214.19 (mml-45c.cisco.com) [1/0/pim]
171.69.214.18 -> 171.69.214.17 (mml-45a.cisco.com) [1/0/pim]
```

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

Advertise Network 0.0.0.0 to DVMRP Neighbors

The mrouted protocol is a public domain implementation of DVMRP. If your router is a neighbor to an mrouted version 3.6 machine, you can configure the Cisco IOS software to advertise network 0.0.0.0 to the DVMRP neighbor. Do not advertise the DVMRP default into the mbone. You must specify whether only route 0.0.0.0 is advertised, or other routes can also be specified.

To advertise network 0.0.0.0 to DVMRP neighbors on an interface, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertise network 0.0.0.0 to DVMRP neighbors.</td>
<td>ip dvmrp default-information {originate</td>
</tr>
</tbody>
</table>

Configure a DVMRP Tunnel

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

When a Cisco router runs DVMRP over a tunnel, it advertises sources in DVMRP report messages much as it does on real networks. In addition, DVMRP report messages received are cached by the Cisco IOS software and are used as part of its Reverse Path Forwarding (RPF) calculation. This allows multicast packets received over the tunnel to be forwarded by the software.
When you configure a DVMRP tunnel, you should assign a tunnel an address for two reasons:

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

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

To configure a DVMRP tunnel, perform the following tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Specify a tunnel interface in global configuration mode. This puts the router into interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Set the tunnel interface’s source address. This is the IP address of the interface on the router.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Set the tunnel interface’s destination address. This is the IP address of the mrouted multitask router.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Configure a DVMRP tunnel.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Assign an IP address to the interface.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Configure PIM on the interface.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Configure an acceptance filter for incoming DVMRP reports.</td>
</tr>
</tbody>
</table>

1. This command is documented in the “Interface Commands” chapter of the *Configuration Fundamentals Command Reference*.
2. This command is documented in the “IP Commands” chapter of the *Network Protocols Command Reference, Part 1*.

See the “Configure DVMRP Interoperability Examples” section later in this chapter for an example of how to configure a DVMRP interoperability over a tunnel interface.

### Configure an IP Multicast Static Route

IP multicast static routes (mroutes) allow you to have multicast paths diverge from the unicast paths. When using PIM, the router expects to receive packets on the same interface where it sends unicast packets back to the source. This is beneficial if your multicast and unicast topologies are congruent. However, you might want unicast packets to take one path and multicast packets to take another.

The most common reason for using separate unicast and multicast paths is tunneling. When a path between a source and a destination does not support multicast routing, a solution is to configure two routers with a GRE tunnel between them. In Figure 23, the UR routers support unicast packets only; the MR routers support multicast packets.
In Figure 23, Source delivers multicast packets to Destination by using MR1 and MR2. MR2 accepts the multicast packet only if it thinks it can reach Source over the tunnel. If this is true, when Destination sends unicast packets to Source, MR2 sends them over the tunnel. This could be slower than natively sending the unicast packet through UR2, UR1, and MR1.

Prior to multicast static routes, the configuration in Figure 24 was used to overcome the problem of both unicasts and multicasts using the tunnel. In this figure, MR1 and MR2 are used as multicast routers only. When Destination sends unicast packets to Source, it uses the (UR3,UR2,UR1) path. When Destination sends multicast packets, the UR routers do not understand or forward them. However, the MR routers forward the packets.

To make the configuration in Figure 24 work, MR1 and MR2 must run another routing protocol (typically a different instantiation of the same protocol running in the UR routers), so that paths from sources are learned dynamically.

A multicast static route allows you to use the configuration in Figure 23 by configuring a static multicast source. The Cisco IOS software uses the configuration information instead of the unicast routing table. This allows multicast packets to use the tunnel without having unicast packets use the tunnel. Static mroutes are local to the router they are configured on and not advertised or redistributed in any way to any other router.

To configure a multicast static route, perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure an IP multicast static route.</td>
<td><strong>ip mroute</strong> source mask [protocol as-number] [rpf-address</td>
</tr>
</tbody>
</table>
Configure IP Multicast Routing

Disable Fast Switching of IP Multicast

Fast switching of IP multicast packets is enabled by default on all interfaces (including GRE and DVMRP tunnels), with one exception: It is disabled and not supported over X.25 encapsulated interfaces.

- If fast switching is disabled on an incoming interface for a multicast routing table entry, the packet is sent at process level for all interfaces in the outgoing interface list.
- If fast switching is disabled on an outgoing interface for a multicast routing table entry, the packet is process-level switched for that interface, but might be fast switched for other interfaces in the outgoing interface list.

You disable fast switching if you want to log debug messages, because when fast switching is enabled, debug messages are not logged.

To disable fast switching of IP multicast, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable fast switching of IP multicast</td>
<td>no ip mroute-cache</td>
</tr>
</tbody>
</table>

Control the Transmission Rate to a Multicast Group

By default, you have no limit as to how fast a sender can transmit packets to a multicast group. You can control the rate that the sender from the source list can send to a multicast group in the group list by performing the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control transmission rate to a multicast group</td>
<td>ip multicast rate-limit [in</td>
</tr>
</tbody>
</table>

Enable PIM Nonbroadcast, Multiaccess (NBMA) Mode

PIM nonbroadcast, multiaccess (NBMA) mode allows the Cisco IOS software to replicate packets for each neighbor on the NBMA network. Traditionally, the software replicates multicast and broadcast packets to all “broadcast” configured neighbors. This might be inefficient when not all neighbors want packets for certain multicast groups. NBMA mode allows you to reduce bandwidth on links leading into the NBMA network, as well as CPU cycles in switches and attached neighbors.

Configure this feature on ATM, Frame Relay, SMDS, PRI ISDN, or X.25 networks only, especially when these media do not have native multicast available. Do not use this feature on multicast-capable LANs such as Ethernet or FDDI.

We encourage you to use sparse-mode PIM with this feature. Therefore, when each join is received from NBMA neighbors, PIM stores each neighbor IP address/interface in the outgoing interface list for the group. When a packet is destined for the group, the software replicates the packet and unicasts (data-link unicasts) it to each neighbor that has joined the group.

To enable PIM nonbroadcast, multicaccess mode on your serial link, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable PIM nonbroadcast, multiaccess mode</td>
<td>ip pim nbma-mode</td>
</tr>
</tbody>
</table>
Consider the following two factors before enabling PIM NBMA mode:

- If the number of neighbors grows, the outgoing interface list will get large. This will cost memory and replication time.
- If the network (Frame Relay, SMDS, or ATM) supports multicast natively, we encourage you to use it so that replication is performed at optimal points in the network.

Enable sd Listener Support

The multicast backbone (MBONE) allows efficient, many-to-many communication and is widely used for multimedia conferencing. To help announce multimedia conference sessions and provide the necessary conference setup information to potential participants, the session directory (sd) tool was written. A session directory client announcing a conference session periodically multicasts an announcement packet on a well-known multicast address and port.

To enable session directory listener support, perform the following task in interface configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable sd listener support.</td>
<td>ip sd listen</td>
</tr>
</tbody>
</table>

Configure Routing Protocol-Independent Features

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

- Use Variable-Length Subnet Masks (VLSMs)
- Configure Static Routes
- Specify Default Routes
- Redistribute Routing Information
- Filter Routing Information
- Enable Policy Routing
- Adjust Timers
- Enable or Disable Split Horizon
- Manage Authentication Keys

Use Variable-Length Subnet Masks (VLSMs)

Enhanced IGRP, OSPF, RIP Version 2, static routes, and IS-IS support variable-length subnet masks (VLSMs). With VLSMs, you can use different masks for the same network number on different interfaces, which allows you to conserve IP addresses and more efficiently use available address space. However, using VLSMs also presents address assignment challenges for the network administrator and ongoing administrative challenges.

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

Note Consider your decision to use VLSMs carefully. You can easily make mistakes in address assignments and you will generally find it 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 “OSPF Point-to-Multipoint Example” 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 Cisco IOS software 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:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establish a static route.</td>
<td>`ip route prefix {address</td>
</tr>
</tbody>
</table>

See the “Overriding Static Routes with Dynamic Protocols Example” section at the end of this chapter for an example of configuring static routes.

The software 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 5. If you would like a static route to be overridden by information from a dynamic routing protocol, simply ensure that the administrative distance of the static route is higher than that of the dynamic protocol.

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

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

<table>
<thead>
<tr>
<th>Route Source</th>
<th>Default Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected interface</td>
<td>0</td>
</tr>
<tr>
<td>Static route</td>
<td>1</td>
</tr>
<tr>
<td>Enhanced IGRP summary route</td>
<td>5</td>
</tr>
<tr>
<td>External BGP</td>
<td>20</td>
</tr>
<tr>
<td>Internal Enhanced IGRP</td>
<td>90</td>
</tr>
<tr>
<td>IGRP</td>
<td>100</td>
</tr>
<tr>
<td>OSPF</td>
<td>110</td>
</tr>
<tr>
<td>IS-IS</td>
<td>115</td>
</tr>
<tr>
<td>RIP</td>
<td>120</td>
</tr>
<tr>
<td>EGP</td>
<td>140</td>
</tr>
</tbody>
</table>
Specify Default Routes

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

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

Specify a Default Network

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

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a default network</td>
<td>ip default-network</td>
</tr>
<tr>
<td></td>
<td>network-number</td>
</tr>
</tbody>
</table>

Gateway of Last Resort

When default information is being passed along through a dynamic routing protocol, no further configuration is required. The system periodically scans its routing table to choose the optimal default network as its default route. In the case of RIP, there is only one choice, network 0.0.0.0. In the case of IGRP, there might be several networks that can be candidates for the system default. The Cisco IOS software 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 software, candidates for the default route are specified with the `ip default-network` command. In this usage, `ip default-network` takes an unconnected 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.

If the router has no interface on the default network, but does have a route to it, it considers this network as a candidate default path. The route candidates are examined and the best one is chosen, based on administrative distance and metric. The gateway to the best default path becomes the gateway of last resort.
Configure Routing Protocol-Independent Features

Change the Maximum Number of Paths

By default, most IP routing protocols install a maximum of four parallel routes in a routing table. The exception is BGP, which by default allows only one path to a destination.

The range of maximum paths is 1 to 6 paths. To change the maximum number of parallel paths allowed, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the maximum number of parallel paths allowed in a routing table.</td>
<td><code>maximum-paths</code> maximum</td>
</tr>
</tbody>
</table>

Redistribute Routing Information

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

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

The following five tables list tasks associated with route redistribution. Although redistribution is a protocol-independent feature, some of the match and set commands are specific to a particular protocol.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define any route maps needed to control redistribution.</td>
<td><code>route-map</code> map-tag [permit</td>
</tr>
</tbody>
</table>

One or more match commands and one or more set commands typically follow a route-map command. If there are no match commands, then everything matches. If there are no set commands, nothing is done (other than the match). Therefore, you need at least one match or set command. To define conditions for redistributing routes from one routing protocol into another, perform at least one of the following tasks in route-map configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match a BGP autonomous system path access list.</td>
<td><code>match as-path</code> path-list-number</td>
</tr>
<tr>
<td>Match a BGP community list.</td>
<td><code>match community-list</code> community-list-number [exact]</td>
</tr>
</tbody>
</table>
| Match a standard access list.     | `match ip address` {access-list-number | name ...
|                                   | access-list-number | name...name} |
| Match the specified metric.       | `match metric` metric-value    |
| Match a next-hop router address passed by one of the access lists specified. | `match ip next-hop` {access-list-number | name ...
|                                   | access-list-number | name...name} |
| Match the specified tag value.    | `match tag` tag-value...tag-value |
| Match the specified next-hop route out one of the interfaces specified. | `match interface` type number...type number |
| Match the address specified by the specified advertised access lists. | `match ip route-source` {access-list-number | name ...
|                                   | access-list-number | name...name} |
Configure Routing Protocol-Independent Features

Configuring IP Routing Protocols

To define conditions for redistributing routes from one routing protocol into another, perform at least one of the following tasks in route-map configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match the specified route type.</td>
<td>`match route-type {local</td>
</tr>
</tbody>
</table>

One or more `match` commands and one or more `set` commands must follow a `route-map` command.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the COMMUNITIES attribute.</td>
<td><code>set community community-number [additive]</code></td>
</tr>
<tr>
<td>Assign a value to a local BGP path.</td>
<td><code>set local-preference value</code></td>
</tr>
<tr>
<td>Specify the BGP weight for the routing table.</td>
<td><code>set weight weight</code></td>
</tr>
<tr>
<td>Set the BGP origin code.</td>
<td>`set origin {igp</td>
</tr>
<tr>
<td>Modify the BGP autonomous system path.</td>
<td>`set as-path {tag</td>
</tr>
<tr>
<td>Specify the address of the next hop.</td>
<td><code>set next-hop next-hop</code></td>
</tr>
<tr>
<td>Enable automatic computing of tag table.</td>
<td><code>set automatic-tag</code></td>
</tr>
<tr>
<td>For routes that are advertised into the specified area of the routing domain.</td>
<td>`set level {level-1</td>
</tr>
<tr>
<td>Set the metric value to give the redistributed routes.</td>
<td><code>set metric metric-value</code></td>
</tr>
<tr>
<td>Set the metric type to give the redistributed routes.</td>
<td>`set metric-type {internal</td>
</tr>
<tr>
<td>Set the tag value to associate with the redistributed routes.</td>
<td><code>set tag tag-value</code></td>
</tr>
</tbody>
</table>

See the “BGP Route Map Examples” section at the end of this chapter for examples of BGP route maps. See the “BGP Community Examples with Route Maps” section at the end of this chapter for examples of BGP communities and route maps.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redistribute routes from one routing protocol to another routing protocol.</td>
<td>`redistribute protocol [process-id] {level-1</td>
</tr>
<tr>
<td>Cause the current routing protocol to use the same metric value for all redistributed routes (BGP, EGP, OSPF, RIP).</td>
<td><code>default-metric number</code></td>
</tr>
<tr>
<td>Cause the IGRP or Enhanced IGRP routing protocol to use the same metric value for all redistributed routes.</td>
<td><code>default-metric bandwidth delay reliability loading mtu</code></td>
</tr>
<tr>
<td>Disable the redistribution of default information between IGRP processes. This is enabled by default.</td>
<td>`no default-information [in</td>
</tr>
</tbody>
</table>
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.

Supported Metric Translations

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

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

Filter Routing Information

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

- Suppress the sending of routing updates on a particular router interface. This is done to prevent other systems on an interface from learning about routes dynamically.
- Suppress networks from being advertised in routing updates. This is done to prevent other routers from learning a particular device’s interpretation of one or more routes.
- Suppress networks listed in updates from being accepted and acted upon by a routing process. This is done to keep a router from using certain routes.
- Apply an offset to routing metrics. This is done to provide a local mechanism for increasing the value of routing metrics.
- 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.

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

The following sections describe these tasks.

Suppress Routing Updates through an Interface

To prevent other routers on a local network from learning about routes dynamically, you can keep routing update messages from being sent through a router interface. This feature applies to all IP-based routing protocols except BGP and EGP.
OSPF and IS-IS behaviors are somewhat different. In OSPF, the interface address you specify as passive appears as a stub network in the OSPF domain. OSPF routing information is neither sent nor received through the specified router interface. In IS-IS, the specified IP addresses are advertised without actually running IS-IS on those interfaces.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppress the sending of routing updates through the specified interface.</td>
<td>passive-interface type number</td>
</tr>
</tbody>
</table>

See the “Passive Interface Examples” section at the end of this chapter for examples of configuring passive interfaces.

### Suppress Routes from Being Advertised in Routing Updates

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permit or deny routes from being advertised in routing updates depending upon the action listed in the access list.</td>
<td>distribute-list {access-list-number</td>
</tr>
</tbody>
</table>

### Suppress Routes Listed in Updates from Being Processed

You might want to avoid processing certain routes listed in incoming updates. This feature does not apply to OSPF or IS-IS. Perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppress routes listed in updates from being processed.</td>
<td>distribute-list {access-list-number</td>
</tr>
</tbody>
</table>

### Apply Offsets to Routing Metrics

An offset list is the mechanism for increasing incoming and outgoing metrics to routes learned via RIP and IGRP. Optionally, you can limit the offset list with either an access list or an interface. To increase the value of routing metrics, perform the following task in router configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply an offset to routing metrics.</td>
<td>offset-list {access-list-number</td>
</tr>
</tbody>
</table>

### Filter Sources of Routing Information

An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. In a large network, some routing protocols and some routers can be more reliable than others as sources of routing information. Also, when multiple
Configure Routing Protocol-Independent Features

Routing processes are running in the same router for IP, it is possible for the same route to be advertised by more than one routing process. By specifying administrative distance values, you enable the router to intelligently discriminate between sources of routing information. The router will always pick the route whose routing protocol has the lowest administrative distance.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter on routing information sources.</td>
<td>distance weight [address-mask [access-list-number</td>
</tr>
</tbody>
</table>

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

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

Assigning administrative distances is a problem unique to each network and is done in response to the greatest perceived threats to the network. Even when general guidelines exist, the network manager must ultimately determine a reasonable matrix of administrative distances for the network as a whole. Table 5 shows the default administrative distance for various sources of routing information.

See the “Administrative Distance Examples” section at the end of this chapter for examples of setting administrative distances.

**Enable Policy Routing**

Policy routing is a more flexible mechanism for routing packets than destination routing. It is a process whereby the router puts packets through a route map before routing them. The route map determines which packets are routed to which router next. You might enable policy routing if you want certain packets to be routed some way other than the obvious shortest path. Some possible applications for policy routing are to provide equal access, protocol-sensitive routing, source-sensitive routing, routing based on interactive versus batch traffic, or routing based on dedicated links.

To enable policy routing, you must identify which route map to use for policy routing and create the route map. The route map itself specifies the match criteria and the resulting action if all of the match clauses are met. These steps are described in the following three task tables.
To enable policy routing on an interface, indicate which route map the router should use by performing the following task in interface configuration mode. All packets arriving on the specified interface will be subject to policy routing. This command disables fast switching of all packets arriving on this interface.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify the route map to use for policy routing.</td>
<td><code>ip policy route-map map-tag</code></td>
</tr>
</tbody>
</table>

You must also define the route map to be used for policy routing. Perform the following task in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define a route map to control where packets are output.</td>
<td>`route-map map-tag [permit</td>
</tr>
</tbody>
</table>

The next step is to define the criteria by which packets are examined to see if they will be policy-routed. No match clause in the route map indicates all packets. Perform one or more of the following tasks in route-map configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match the Level 3 length of the packet.</td>
<td><code>match length min max</code></td>
</tr>
<tr>
<td>Match the destination IP address that is permitted by one or more standard or extended access lists.</td>
<td>`match ip address { access-list-number</td>
</tr>
</tbody>
</table>

The last step is to specify where the packets that pass the match criteria are output. To do so, perform one or more of the following tasks in route-map configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the next hop to which to route the packet (it need not be adjacent).</td>
<td><code>set ip next-hop ip-address [... ip-address]</code></td>
</tr>
<tr>
<td>Specify the output interface for the packet.</td>
<td><code>set interface type number [... type number]</code></td>
</tr>
<tr>
<td>Specify the next hop to which to route the packet, if there is no explicit route for this destination.</td>
<td><code>set ip default next-hop ip-address [... ip-address]</code></td>
</tr>
<tr>
<td>Specify the output interface for the packet, if there is no explicit route for this destination.</td>
<td><code>set default interface type number [... type number]</code></td>
</tr>
</tbody>
</table>

The `set` commands can be used in conjunction with each other. They are evaluated in the order shown in the previous task table. A usable next hop implies an interface. Once the local router finds a next hop and a usable interface, it routes the packet.

See the “Policy Routing Example” section at the end of this chapter for an example of policy routing.
Enable Local Policy Routing

Packets that are generated by the router are not normally policy-routed. To enable local policy routing for such packets, indicate which route map the router should use by performing the following task in global configuration mode. All packets originating on the router will then be subject to local policy routing.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify the route map to use for local policy routing.</td>
<td>ip local policy route-map map-tag</td>
</tr>
</tbody>
</table>

Use the `show ip local policy` command to display the route map used for local policy routing, if one exists.

Adjust Timers

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

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

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

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

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

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

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

Perform the following task in router configuration mode:

Perform the following the following task in interface configuration mode:

**Task** | **Command** |
---|---|
Adjust the frequency with which the Cisco IOS software sends messages to itself (Ethernet and Token Ring) or to the other end (HDLC-serial and PPF-serial links) to ensure that a network interface is alive for a specified interface. | `keepalive [seconds]` |

1. This command is documented in the “Interface Commands” chapter of the *Configuration Fundamentals Command Reference.*
You can also configure the keepalive interval, the frequency at which the Cisco IOS software sends messages to itself (Ethernet and Token Ring) or to the other end (hdlc-serial, ppp-serial) to ensure that a network interface is alive. The interval in some previous software versions was 10 seconds; it is now adjustable in one-second increments down to one second. An interface is declared down after three update intervals have passed without receiving a keepalive packet.

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

**Enable or Disable Split Horizon**

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

If an interface is configured with secondary IP addresses and split horizon is enabled, updates might not be sourced by every secondary address. One routing update is sourced per network number unless split horizon is disabled.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable split horizon.</td>
<td>ip split-horizon</td>
</tr>
<tr>
<td>Disable split horizon.</td>
<td>no ip split-horizon</td>
</tr>
</tbody>
</table>

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

See the “Split Horizon Examples” section at the end of this chapter for examples of using split horizon.

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

**Manage Authentication Keys**

*Key management* is a method of controlling authentication keys used by routing protocols. RIP Version 2 is the first protocol that is able to use key management.

Before you manage RIP authentication keys, RIP authentication must be enabled. See the “Enable RIP Authentication” section earlier in this chapter.
To manage authentication keys, define a key chain, identify the keys that belong to the key chain, and specify how long each key is valid. You can configure multiple keys with lifetimes, and the software will rotate through them. Note that the router must know the time. Refer to the NTP and calendar commands in the “Managing the System” chapter of the Configuration Fundamentals Configuration Guide.

To manage authentication keys, perform the following tasks beginning in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify a key chain.</td>
<td>key chain name-of-chain</td>
</tr>
<tr>
<td>In key chain configuration mode, identify the key number.</td>
<td>key number</td>
</tr>
<tr>
<td>In key chain key configuration mode, identify the key string.</td>
<td>key-string text</td>
</tr>
<tr>
<td>Specify the time period during which the key can be received.</td>
<td>accept-lifetime start-time {infinite</td>
</tr>
<tr>
<td>Specify the time period during which the key can be sent.</td>
<td>send-lifetime start-time {infinite</td>
</tr>
</tbody>
</table>

If authentication is enabled, the software sends a RIP packet for every active key on the key chain. Therefore, if two keys on the key chain happen to be active based on the send-lifetime values, the software sends two RIP packets every 30 seconds, one authenticated with each key.

Use the show key chain command to display key chain information. For examples of key management, see the “Key Management Examples” section at the end of this chapter.

Monitor and Maintain the IP Network

You can remove all contents of a particular cache, table, or database. You also can display specific 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:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear the IP ARP cache and the fast-switching cache.</td>
<td>clear arp-cache</td>
</tr>
<tr>
<td>Reset a particular BGP connection.</td>
<td>clear ip bgp address</td>
</tr>
<tr>
<td>Reset all BGP connections.</td>
<td>clear ip bgp *</td>
</tr>
<tr>
<td>Remove all members of a BGP peer group.</td>
<td>clear ip bgp peer-group tag</td>
</tr>
</tbody>
</table>
| Delete routes from the DVMRP routing table.    | clear ip dvmrp route [ * | route]
| Delete neighbors from the neighbor table.      | clear ip eigrp neighbors [ip-address | interface]
| Delete entries from the IGMP cache.            | clear ip igmp group [group-name | group-address | interface] |
Display System and Network Statistics

You can display specific 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 device’s packets are taking through the network.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display the contents of the IP multicast routing table.</td>
<td>clear ip mroute { *</td>
</tr>
<tr>
<td>Clear one or more routes from the IP routing table.</td>
<td>clear ip route { network [mask] } *</td>
</tr>
<tr>
<td>Display a branch of a multicast tree for a specific group.</td>
<td>mbranch group-address branch-address [ttl]</td>
</tr>
<tr>
<td>Display a branch of a multicast tree in the reverse direction.</td>
<td>mrbranch group-address branch-address [ttl]</td>
</tr>
<tr>
<td>Display all BGP routes that contain subnet and supernet network masks.</td>
<td>show ip bgp cidr-only</td>
</tr>
<tr>
<td>Display routes that belong to the specified communities.</td>
<td>show ip bgp community community-number [exact]</td>
</tr>
<tr>
<td>Display routes that are permitted by the community list.</td>
<td>show ip bgp community-list community-list-number [exact]</td>
</tr>
<tr>
<td>Display routes that are matched by the specified autonomous system path access list.</td>
<td>show ip bgp filter-list access-list-number</td>
</tr>
<tr>
<td>Display the routes with inconsistent originating autonomous systems.</td>
<td>show ip bgp inconsistent-as</td>
</tr>
<tr>
<td>Display the routes that match the specified regular expression entered on the command line.</td>
<td>show ip bgp regexp regular-expression</td>
</tr>
<tr>
<td>Display the contents of the BGP routing table.</td>
<td>show ip bgp [network] [network-mask] [subnets]</td>
</tr>
<tr>
<td>Display detailed information on the TCP and BGP connections to individual neighbors.</td>
<td>show ip bgp neighbors [address]</td>
</tr>
<tr>
<td>Display routes learned from a particular BGP neighbor.</td>
<td>show ip bgp neighbors address [routes</td>
</tr>
<tr>
<td>Display all BGP paths in the database.</td>
<td>show ip bgp paths</td>
</tr>
<tr>
<td>Display information about BGP peer groups.</td>
<td>show ip bgp peer-group [tag] [summary]</td>
</tr>
<tr>
<td>Display the status of all BGP connections.</td>
<td>show ip bgp summary</td>
</tr>
<tr>
<td>Display the entries in the DVMRP routing table.</td>
<td>show ip dvmrp route [ip-address]</td>
</tr>
<tr>
<td>Display statistics on EGP connections and neighbors.</td>
<td>show ip egp</td>
</tr>
<tr>
<td>Display the IP Enhanced IGRP discovered neighbors.</td>
<td>show ip eigrp neighbors [type number]</td>
</tr>
<tr>
<td>Display the IP Enhanced IGRP topology table for a given process.</td>
<td>show ip eigrp topology [autonomous-system-number</td>
</tr>
</tbody>
</table>
### Task

Display the number of packets sent and received for all or a specified IP Enhanced IGRP process.

**Command**

```
show ip eigrp traffic [autonomous-system-number]
```

Display the multicast groups that are directly connected to the router and that were learned via IGMP.

**Command**

```
show ip igmp groups [group-name | group-address | type number]
```

Display multicast-related information about an interface.

**Command**

```
show ip igmp interface [type number]
```

Display IRDP values.

**Command**

```
show ip irdp
```

Display the contents of the IP fast-switching cache.

**Command**

```
show ip mcache [group [source]]
```

Display the contents of the IP multicast routing table.

**Command**

```
show ip mroute [group] [source] [summary] [count]
```

Display general information about OSPF routing processes.

**Command**

```
show ip ospf [process-id]
```

Display lists of information related to the OSPF database.

**Command**

```
show ip ospf [process-id area-id] database
show ip ospf [process-id area-id] database [router]
[link-state-id]
show ip ospf [process-id area-id] database [network]
[link-state-id]
show ip ospf [process-id area-id] database [summary]
[link-state-id]
show ip ospf [process-id area-id] database
[asb-summary] [link-state-id]
show ip ospf [process-id] database [external]
[link-state-id]
show ip ospf [process-id area-id] database
[database-summary]
```

Display the internal OSPF routing table entries to Area Border Router (ABR) and Autonomous System Boundary Router (ASBR).

**Command**

```
show ip ospf border-routers
```

Display OSPF-related interface information.

**Command**

```
show ip ospf interface [interface-name]
```

Display OSPF-neighbor information on a per-interface basis.

**Command**

```
show ip ospf neighbor [interface-name] [neighbor-id] detail
```

Display OSPF-related virtual links information.

**Command**

```
show ip ospf virtual-links
```

Display information about interfaces configured for PIM.

**Command**

```
show ip pim interface [type number]
```

List the PIM neighbors discovered by the router.

**Command**

```
show ip pim neighbor [type number]
```

Display the RP routers associated with a sparse-mode multicast group.

**Command**

```
show ip pim rp [group-name | group-address]
```

Display the local policy route map, if any.

**Command**

```
show ip local policy
```

Display policy route maps.

**Command**

```
show ip policy
```

Display the parameters and current state of the active routing protocol process.

**Command**

```
show ip protocols
```

Display the current state of the routing table.

**Command**

```
show route [address [mask] [longer-prefixes]] [protocol [process-id]]
```
IP Routing Protocol Configuration Examples

The following sections provide IP routing protocol configuration examples:

- Static Routing Redistribution Example
- Key Management Examples
- Static Routing Redistribution Example
- BGP Route Map Examples
- BGP Neighbor Configuration Examples
  - Using Access Lists to Specify Neighbors
- BGP Path Filtering by Neighbor Example
- BGP Path Filtering by Neighbor Example
- BGP Aggregate Route Examples
- BGP Confederation Example
- BGP Peer Group Examples
  - IBGP Peer Group Example
  - EIBGP Peer Group Example
- Third-Party EGP Support Example
- Backup EGP Router Example
- EGP Core Gateway Example
- IP Multicast Routing Configuration Examples
  - Configure to Operate in Dense Mode Example
  - Configure to Operate in Sparse Mode Example
  - Configure DVMRP Interoperability Examples
- OSPF Point-to-Multipoint Example
- Overriding Static Routes with Dynamic Protocols Example
- BGP Community Examples with Route Maps
- Passive Interface Examples

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display the current state of the routing table in summary form.</td>
<td><code>show ip route summary</code></td>
</tr>
<tr>
<td>Display supernets.</td>
<td><code>show ip route supernets-only</code></td>
</tr>
<tr>
<td>Display the contents of the session directory.</td>
<td>`show ip sd [group</td>
</tr>
<tr>
<td>Display the IS-IS link state database.</td>
<td><code>show isis database [level-1] [level-2] [l1] [l2] [detail] [lspid]</code></td>
</tr>
<tr>
<td>Display authentication key information.</td>
<td><code>show key chain [name]</code></td>
</tr>
<tr>
<td>Display all route maps configured or only the one specified.</td>
<td><code>show route-map [map-name]</code></td>
</tr>
</tbody>
</table>
• Administrative Distance Examples
• Policy Routing Example
• Split Horizon Examples

OSPF Point-to-Multipoint Example

In Figure 25, Mollie uses DLCI 201 to communicate with Neon, DLCI 202 to Jelly, and DLCI 203 to Platty. Neon uses DLCI 101 to communicate with Mollie and DLCI 102 to communicate with Platty. Platty communicates with Neon (DLCI 401) and Mollie (DLCI 402). Jelly communicates with Mollie (DLCI 301).

Figure 25  OSPF Point-to-Multipoint Example

Mollie's Configuration

```
hostname mollie
!
interface serial 1
ip address 10.0.0.2 255.0.0.0
ip ospf network point-to-multipoint
encapsulation frame-relay
frame-relay map ip 10.0.0.1 201 broadcast
frame-relay map ip 10.0.0.3 202 broadcast
frame-relay map ip 10.0.0.4 203 broadcast
!
router ospf 1
network 10.0.0.0 0.0.0.255 area 0
```

Neon's Configuration

```
hostname neon
!
interface serial 0
ip address 10.0.0.1 255.0.0.0
ip ospf network point-to-multipoint
encapsulation frame-relay
frame-relay map ip 10.0.0.2 101 broadcast
frame-relay map ip 10.0.0.3 202 broadcast
frame-relay map ip 10.0.0.4 203 broadcast
!
router ospf 1
network 10.0.0.0 0.0.0.255 area 0
```
Platty's Configuration

hostname platty
!
interface serial 3
ip address 10.0.0.4 255.0.0.0
ip ospf network point-to-multipoint
encapsulation frame-relay
clockrate 1000000
frame-relay map ip 10.0.0.1 401 broadcast
frame-relay map ip 10.0.0.2 402 broadcast
!
router ospf 1
network 10.0.0.0 0.0.0.255 area 0

Jelly's Configuration

hostname jelly
!
interface serial 2
ip address 10.0.0.3 255.0.0.0
ip ospf network point-to-multipoint
encapsulation frame-relay
clockrate 2000000
frame-relay map ip 10.0.0.2 301 broadcast
!
router ospf 1
network 10.0.0.0 0.0.0.255 area 0

Key Management Examples

The following example configures a key chain called trees. In this example, the software will always accept and send willow as a valid key. The key chestnut will be accepted from 1:30 p.m. to 3:30 p.m. and be sent from 2:00 p.m. to 3:00 p.m. The overlap allows for migration of keys or discrepancies in the router’s time. Likewise, the key birch immediately follows chestnut, and there is a half hour leeway on each side to handle time-of-day differences.

interface ethernet 0
  ip rip authentication key-chain trees
  ip rip authentication mode md5
!
routing rip
  network 172.19.0.0
  version 2
!
key chain trees
  key 1
    key-string willow
  key 2
    key-string chestnut
    accept-lifetime 13:30:00 Jan 25 1996 duration 7200
    send-lifetime 14:00:00 Jan 25 1996 duration 3600
  key 3
    key-string birch
    accept-lifetime 14:30:00 Jan 25 1996 duration 7200
    send-lifetime 15:00:00 Jan 25 1996 duration 3600

The following example configures a key chain called flintstone.

key chain flintstone
  key 1
    key-string fred
  key 2
IP Routing Protocol Configuration Examples

key-string barney
accept-lifetime 00:00:00 Dec 5 1995 23:59:59 Dec 5 1995
send-lifetime 06:00:00 Dec 5 1995 18:00:00 Dec 5 1995

interface Ethernet0
ip address 172.19.104.75 255.255.255.0 secondary
ip address 171.69.232.147 255.255.255.240
ip rip authentication key-chain flintstone
media-type 10BaseT

interface Ethernet1
no ip address
shutdown
media-type 10BaseT

interface Fddi0
ip address 2.1.1.1 255.255.255.0
no keepalive

interface Fddi1
ip address 3.1.1.1 255.255.255.0
ip rip send version 1
ip rip receive version 1
no keepalive

router rip
version 2
network 172.19.0.0
network 2.0.0.0
network 3.0.0.0

IS-IS as an IP Routing Protocol Example

The following example shows how you would configure three routers to run IS-IS as an IP routing protocol. Figure 26 illustrates the example configuration.

Figure 26  IS-IS Routing

Configuration for Router A

router isis
net 49.0001.0000.0000.000a.00
interface ethernet 0
ip router isis
interface serial 0
ip router isis

Configuration for Router B
router isis
net 49.0001.0000.0000.000b.00
interface ethernet 0
ip router isis
interface ethernet 1
ip router isis
interface serial 0
ip router isis

Configuration for Router C
router isis
net 49.0001.0000.0000.000c.00
interface ethernet 1
ip router isis
interface ethernet 2
ip router isis

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

```
router bgp 100
neighbor 140.222.1.1 route-map fix-weight in
neighbor 140.222.1.1 remote-as 1

route-map fix-weight permit 10
match as-path 200
set local-preference 250
set weight 200

ip as-path access-list 200 permit ^690$
ip as-path access-list 200 permit ^1800
```

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

```
router bgp 100
neighbor 1.1.1.1 route-map freddy out

ip as-path access-list 1 permit ^690_
ip as-path access-list 2 permit .*

route-map freddy permit 10
match as-path 1
set metric 127

route-map freddy permit 20
match as-path 2
```

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

```
router bgp 100
redistribute igrp 109 route-map igrp2bgp

route-map igrp2bgp
match ip address 1
set local-preference 25
set metric 127
set weight 30000
set next-hop 192.92.68.24
set origin igp

access-list 1 permit 131.108.0.0 0.0.255.255
access-list 1 permit 160.89.0.0 0.0.255.255
access-list 1 permit 198.112.0.0 0.0.127.255
```

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

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

route-map fnord permit 20
match as-path 2
```

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

```
router bgp 100
table-map set_ospf_tag

route-map set_ospf_tag
match as-path 1
set automatic-tag

ip as-path access-list 1 permit .*
```

In the following example, the route map called `set-as-path` is applied to outbound updates to the neighbor 200.69.232.70. The route map will prepend the autonomous system path “100 100” to routes that pass access list 1. The second part of the route map is to permit the advertisement of other routes.

```
router bgp 100
network 171.60.0.0
network 172.60.0.0
neighbor 200.69.232.70 remote-as 200
neighbor 200.69.232.70 route-map set-as-path out

route-map set-as-path 10 permit
match address 1
set as-path prepend 100 100

route-map set-as-path 20 permit
match address 2

access-list 1 permit 171.60.0.0 0.0.255.255
access-list 1 permit 172.60.0.0 0.0.255.255
```
Inbound route-maps could do prefix-based matching and set various parameters of the update. Inbound prefix matching is available in addition to as-path and community-list matching. In the following example, the `set local preference` command sets the local preference of the inbound prefix 140.10.0.0/16 to 120.

```plaintext
access-list 2 permit 0.0.0.0 255.255.255.255

BGP Neighbor Configuration Examples
In the following example, a BGP router is assigned to autonomous system 109, and two networks are listed as originating in the autonomous system. Then the addresses of three remote routers (and their autonomous systems) are listed. The router being configured will share information about networks 131.108.0.0 and 192.31.7.0 with the neighbor routers. The first router listed is in a different autonomous system; the second `neighbor` command specifies an internal neighbor (with the same autonomous system number) at address 131.108.234.2; and the third `neighbor` command specifies a neighbor on a different autonomous system.

```
router bgp 109
network 131.108.0.0
network 192.31.7.0
neighbor 131.108.1.1 remote-as 200
neighbor 131.108.1.1 route-map set-local-pref in
route-map set-local-pref permit 10
match ip address 2
set local preference 120
!
route-map set-local-pref permit 20
!
access-list 2 permit 140.10.0.0 0.0.255.255 access-list 2 deny any
```

In Figure 27, Router A is being configured. The internal BGP neighbor is not directly linked to Router A. External neighbors (in autonomous system 167 and autonomous system 99) must be linked directly to Router A.
Using Access Lists to Specify Neighbors

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

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

BGP Synchronization Example

In the configuration shown in Figure 28, with synchronization on, Router B does not advertise network 198.92.68.0 to Router A until an IGRP route for network 198.92.68.0 exists. If you specify the **no synchronization** router configuration command, Router B advertises network 198.92.68.0 as soon as possible. However, because routing information still must be sent to interior peers, you must configure a full internal BGP mesh.
BGP Path Filtering by Neighbor Example

The following is an example of BGP path filtering by neighbor. The routes that pass as-path access list 1 will get weight 100. Only the routes that pass as-path access list 2 will be sent to 193.1.12.10. Similarly, only routes passing access list 3 will be accepted from 193.1.12.10.

```
router bgp 200
neighbor 193.1.12.10 remote-as 100
neighbor 193.1.12.10 filter-list 1 weight 100
neighbor 193.1.12.10 filter-list 2 out
neighbor 193.1.12.10 filter-list 3 in
ip as-path access-list 1 permit _109_
ip as-path access-list 2 permit _200$
ip as-path access-list 2 permit ^100$
ip as-path access-list 3 deny _6905
ip as-path access-list 3 permit .*
```

BGP Aggregate Route Examples

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

In the following example, the `redistribute static` command is used to redistribute aggregate route 193.0.0.0.

```
ip route 193.0.0.0 255.0.0.0 null 0
! router bgp 100
redistribute static
```
The following configuration creates an aggregate entry in the BGP routing table when there is at least one specific route that falls into the specified range. The aggregate route will be advertised as coming from your autonomous system and has the atomic aggregate attribute set to show that information might be missing. (By default, atomic aggregate is set unless you use the `as-set` keyword in the `aggregate-address` command.)

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

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

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

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

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

### BGP Confederation Example

The following is a sample configuration from several peers in a confederation. The confederation consists of three internal autonomous systems with autonomous system numbers 6001, 6002, and 6003. To the BGP speakers outside the confederation, the confederation looks like a normal autonomous system with autonomous system number 666 (specified via the `bgp confederation identifier` command).

In a BGP speaker in autonomous system 6001, the `bgp confederation peers` command marks the peers from autonomous systems 6002 and 6003 as special EBGP peers. Hence peers 171.69.232.55 and 171.69.232.56 will get the local-preference, next-hop and MED unmodified in the updates. The router at 160.69.69.1 is a normal EBGP speaker and the updates received by it from this peer will be just like a normal EBGP update from a peer in autonomous system 666.

```
router bgp 6001
  bgp confederation identified 666
  bgp confederation peers 6002 6003
  neighbor 171.69.232.55 remote-as 6002
  neighbor 171.69.232.56 remote-as 6003
  neighbor 160.69.69.1 remote-as 777
```

In a BGP speaker in autonomous system 6002, the peers from autonomous systems 6001 and 6003 are configured as special EBGP peers. 170.70.70.1 is a normal IBGP peer and 199.99.99.2 is a normal EBGP peer from autonomous system 700.

```
router bgp 6002
  bgp confederation identified 666
  bgp confederation peers 6001 6003
  neighbor 170.70.70.1 remote-as 6002
  neighbor 171.69.232.57 remote-as 6001
  neighbor 171.69.232.56 remote-as 6003
  neighbor 199.99.99.2 remote-as 700
```

In a BGP speaker in autonomous system 6003, the peers from autonomous systems 6001 and 6002 are configured as special EBGP peers. 200.200.200.200.200 is a normal EBGP peer from autonomous system 701.

```
router bgp 6003
  bgp confederation identified 666
```
IP Routing Protocol Configuration Examples

Configuring IP Routing Protocols

V-177

```
bgp confederation peers 6001 6002
neighbor 171.69.232.57 remote-as 6001
neighbor 171.69.232.55 remote-as 6002
neighbor 200.200.200.200 remote-as 701

The following is a part of the configuration from the BGP speaker 200.200.200.205 from autonomous system 701. Neighbor 171.69.232.56 is configured as a normal EBGP speaker from autonomous system 666. The internal division of the autonomous system into multiple autonomous systems is not known to the peers external to the confederation.

router bgp 701
neighbor 171.69.232.56 remote-as 666
neighbor 200.200.200.205 remote-as 701
```

BGP Peer Group Examples

This section contains an IBGP peer group example and an EBGP peer group example.

**IBGP Peer Group Example**

In the following example, the peer group named `internal` configures the members of the peer group to be IBGP neighbors. By definition, this is an IBGP peer group because the `router bgp` command and the `neighbor remote-as` command indicate the same autonomous system (in this case, AS 100).

All the peer group members use loopback 0 as the update source and use `set-med` as the outbound route-map. The example also shows that, except for the neighbor at address 171.69.232.55, all the neighbors have filter-list 2 as the inbound filter list.

```
router bgp 100
neighbor internal peer-group
neighbor internal remote-as 100
neighbor internal update-source loopback 0
neighbor internal route-map set-med out
neighbor internal filter-list 1 out
neighbor internal filter-list 2 in
neighbor 171.69.232.53 peer-group internal
neighbor 171.69.232.54 peer-group internal
neighbor 171.69.232.55 peer-group internal
neighbor 171.69.232.55 filter-list 3 in
```

**EBGP Peer Group Example**

In the following example, the peer group `external-peers` is defined without the `neighbor remote-as` command. This is what makes it an EBGP peer group. Each member of the peer group is configured with its respective autonomous system number separately. Thus, the peer group consists of members from autonomous systems 200, 300, and 400. All the peer group members have `set-metric` route map as an outbound route map and filter-list 99 as an outbound filter list. Except for neighbor 171.69.232.110, all have 101 as the inbound filter list.

```
router bgp 100
neighbor external-peers peer-group
neighbor external-peers route-map set-metric out
neighbor external-peers filter-list 99 out
neighbor external-peers filter-list 101 in
neighbor 171.69.232.90 remote-as 200
neighbor 171.69.232.90 peer-group external-peers
neighbor 171.69.232.100 remote-as 300
neighbor 171.69.232.100 peer-group external-peers
neighbor 171.69.232.110 remote-as 400
```
Third-Party EGP Support Example

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

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

Backup EGP Router Example

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

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

EGP Core Gateway Example

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

Figure 29 illustrates an environment with three routers (designated C1, C2, and C3) attached to a common X.25 network. The routers are intended to route information using EGP. With the following configuration (on the router designated Core), C1, C2, and C3 cannot route traffic directly to each other via the X.25 network:

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

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

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

```
access-list 2 permit 10.0.0.0 0.255.255.255
! global access list assignment
router egp 0
neighbor any 2
neighbor any third-party 10.1.1.1
```
IP Multicast Routing Configuration Examples

This section provides the following IP multicast routing configuration examples:

- Configure to Operate in Dense Mode Example
- Configure to Operate in Sparse Mode Example
- Configure DVMRP Interoperability Examples

Configure to Operate in Dense Mode Example

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

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

Configure to Operate in Sparse Mode Example

The following example configures the Cisco IOS software to operate in sparse-mode PIM. The RP router is the router whose address is 10.8.0.20.

```
ip multicast-routing
ip pim rp-address 10.8.0.20 1
interface ethernet 1
ip pim sparse-mode
```
Configure DVMRP Interoperability Examples

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

interface ethernet 0
ip address 131.119.244.244 255.255.255.0
ip pim dense-mode
ip dvmrp metric 1 list 1
ip dvmrp metric 0 list 2
access-list 1 permit 198.92.35.0 0.0.0.255
access-list 1 permit 198.92.36.0 0.0.0.255
access-list 1 permit 198.92.37.0 0.0.0.255
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 1 permit 150.136.0.0 0.0.255.255
access-list 1 deny 0.0.0.0 255.255.255.255
access-list 2 permit 0.0.0.0 255.255.255.255

The following example configures DVMRP interoperability over a tunnel interface:

! ip multicast-routing
!
interface tunnel 0
ip unnumbered ethernet 0
ip pim dense-mode
tunnel source ethernet 0
tunnel destination 192.70.92.133
tunnel mode dvmrp
!
interface ethernet 0
description Universitat DMZ-ethernet
ip address 192.76.243.2 255.255.255.0
ip pim dense-mode

Variable-Length Subnet Masks Example

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

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

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

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

! Router is configured for OSPF and assigned AS 107
router ospf 107
! Specifies network directly connected to the router
network 131.107.0.0 0.0.255.255 area 0.0.0.0
Overriding Static Routes with Dynamic Protocols Example

In the following example, packets for network 10.0.0.0 from Router B (where the static route is installed) will be routed through 131.108.3.4 if a route with an administrative distance less than 110 is not available. Figure 30 illustrates this point. The route learned by a protocol with an administrative distance of less than 110 might cause Router B to send traffic destined for network 10.0.0.0 via the alternate path—through Router D.

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

![Figure 30 Overriding Static Routes](image)

BGP Community Examples with Route Maps

This section contains three examples of the use of BGP communities with route maps.

In the first example, the route map `set-community` is applied to the outbound updates to the neighbor 171.69.232.50. The routes that pass access list 1 have the special community attribute value “no-export.” The remaining routes are advertised normally. This special community value automatically prevents the advertisement of those routes by the BGP speakers in autonomous system 200.

```
router bgp 100
neighbor 171.69.232.50 remote-as 200
neighbor 171.69.232.50 send-community
neighbor 171.69.232.50 route-map set-community out
!
route-map set-community 10 permit
match address 1
set community no-export
!
route-map set-community 20 permit
match address 2
```

In the second example, the route map `set-community` is applied to the outbound updates to neighbor 171.69.232.90. All the routes that originate from AS 70 have the community values 200 200 added to their already existing values. All other routes are advertised as normal.

```
route-map bgp 200
```
neighbor 171.69.232.90 remote-as 100
neighbor 171.69.232.90 send-community
neighbor 171.69.232.90 route-map set-community out
  route-map set-community 10 permit
  match as-path 1
  set community 200 200 additive
  !
  route-map set-community 20 permit
  match as-path 2
  !
  ip as-path access-list 1 permit 705
  ip as-path access-list 2 permit .*

In the third example, community-based matching is used to selectively set MED and local-preference for routes from neighbor 171.69.232.55. All the routes that match community list 1 get the MED set to 8000. This includes any routes that have the communities “100 200 300” or “900 901.” These routes could have other community values also.

All the routes that pass community list 2 get the local preference set to 500. This includes the routes that have community values 88 or 90. If they belong to any other community, they will not be matched by community list 2.

All the routes that match community list 3 get the local-preference set to 50. Community list 3 will match all the routes because all the routes are members of the Internet community. Thus, all the remaining routes from neighbor 171.69.232.55 get a local preference 50.

router bgp 200
neighbor 171.69.232.55 remote-as 100
neighbor 171.69.232.55 route-map filter-on-community in
  route-map filter-on-community 10 permit
  match community 1
  set metric 8000
  !
  route-map filter-on-community 20 permit
  match community 2 exact-match
  set local-preference 500
  !
  route-map filter-on-community 30 permit
  match community 3
  set local-preference 50
  !
  ip community-list 1 permit 100 200 300
  ip community-list 1 permit 900 901
  !
  ip community-list 2 permit 88
  ip community-list 2 permit 90
  !
  ip community-list 3 permit internet

Passive Interface Examples

The following example sends IGRP updates to all interfaces on network 131.108.0.0 except interface Ethernet 1. Figure 31 shows this configuration.

router igrp 109
network 131.108.0.0
passive-interface ethernet 1
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.

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

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

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

```plaintext
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:

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

Administrative Distance Examples

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

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

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

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

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

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

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

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

```
router isis
distance 90 ip
```

### Policy Routing Example

The following example provides two sources with equal access to two different service providers. Packets arriving on async interface 1 from the source 1.1.1.1 are sent to the router at 6.6.6.6 if the router has no explicit route for the packet’s destination. Packets arriving from the source 2.2.2.2 are sent to the router at 7.7.7.7 if the router has no explicit route for the packet’s destination. All other packets for which the router has no explicit route to the destination are discarded.

```
access-list 1 permit ip 1.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface async 1
ip policy route-map equal-access
!
route-map equal-access permit 10
match ip address 1
set ip default next-hop 6.6.6.6
route-map equal-access permit 20
match ip address 2
set ip default next-hop 7.7.7.7
route-map equal-access permit 30
set default interface null0
```

### Split Horizon Examples

Two examples of configuring split horizon are provided.
Example 1
The following sample configuration illustrates a simple example of disabling split horizon on a serial link. In this example, the serial link is connected to an X.25 network.

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

Example 2
In the next example, Figure 32 illustrates a typical situation in which the **no ip split-horizon** interface configuration command would be useful. This figure depicts two IP subnets that are both accessible via a serial interface on Router C (connected to Frame Relay network). In this example, the serial interface on Router C accommodates one of the subnets via the assignment of a secondary IP address.

The Ethernet interfaces for Router A, Router B, and Router C (connected to IP networks 12.13.50.0, 10.20.40.0, and 20.155.120.0) all have split horizon *enabled* by default, while the serial interfaces connected to networks 128.125.1.0 and 131.108.1.0 all have split horizon *disabled* by default. The partial interface configuration specifications for each router that follow Figure 32 illustrate that the **ip split-horizon** command is *not* explicitly configured under normal conditions for any of the interfaces.

**Figure 32  Disabled Split Horizon Example for Frame Relay Network**

In this example, split horizon must be disabled in order for network 128.125.0.0 to be advertised into network 131.108.0.0, and vice versa. These subnets overlap at Router C, interface S0. If split horizon were enabled on serial interface S0, it would not advertise a route back into the Frame Relay network for either of these networks.
Configuration for Router A

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

Configuration for Router B

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

Configuration for Router C

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

Static Routing Redistribution Example

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

```bash
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
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 Cisco IOS software must run a separate IGRP process and maintain a separate routing database for each autonomous system it services. However, you can transfer routing information between these routing databases.

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

```bash
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:

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

RIP and IGRP Redistribution Example

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

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

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

The `distribute-list` router configuration command instructs the Cisco IOS software 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.

IP Enhanced IGRP Redistribution Examples

Each IP Enhanced IGRP routing process provides routing information to only one autonomous system. The Cisco IOS software must run a separate IP Enhanced IGRP process and maintain a separate routing database for each autonomous system it services. However, you can transfer routing information between these routing databases.

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

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

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

```plaintext
router eigrp 71
redistribute eigrp 109 route-map 109-to-71
route-map 109-to-71 permit
match ip address 3
set metric 10000 100 1 255 1500
access-list 3 permit 192.31.7.0
```
The following example is an alternative way to transfer a route to 192.31.7.0 into autonomous system 71. Unlike the previous configuration, this one does not allow you to arbitrarily set the metric.

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

### RIP and IP Enhanced IGRP Redistribution Examples

This sections provides two examples of RIP and IP Enhanced IGRP redistribution, a simple one and a complex one.

**Example 1: Simple Redistribution**

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

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

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

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

**Example 2: Complex Redistribution**

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

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

```
! Configuration for router R1:
router bgp 1
network 131.108.0.0
neighbor 192.5.10.1 remote-as 2
neighbor 192.5.10.15 remote-as 1
neighbor 192.5.10.24 remote-as 3
redistribute eigrp 1
distribute-list 1 out eigrp 1
!
! All networks that should be advertised from R1 are controlled with access lists:
```
IP Routing Protocol Configuration Examples

Configuring IP Routing Protocols

OSPF Routing and Route Redistribution Examples

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

Three examples follow:

- The first is a simple configuration illustrating basic OSPF commands.
- The second example illustrates a configuration for an internal router, ABR, and ASBRs within a single, arbitrarily assigned, OSPF autonomous system.
- The third example illustrates a more complex configuration and the application of various tools available for controlling OSPF-based routing environments.

Example 1: Basic OSPF Configuration

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

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

Example 2: Another Basic OSPF Configuration

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

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

Each **network** router configuration command is evaluated sequentially, so the specific order of these commands in the configuration is important. The Cisco IOS software sequentially evaluates the *address/wildcard-mask* pair for each interface. See the “IP Routing Protocols Commands” chapter of the *Network Protocols Command Reference, Part 1* for more information.

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

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

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

**Example 3: Internal Router, ABR, and ASBRs**

The following example outlines a configuration for several routers within a single OSPF autonomous system. Figure 33 provides a general network map that illustrates this example configuration.
In this configuration, five routers are configured in OSPF autonomous system 109:

- Router A and Router B are both internal routers within Area 1.
- Router C is an OSPF area border router. Note that for Router C, Area 1 is assigned to E3 and Area 0 is assigned to S0.
- Router D is an internal router in Area 0 (backbone area). In this case, both network router configuration commands specify the same area (Area 0, or the backbone area).
- Router E is an OSPF autonomous system boundary router. Note that BGP routes are redistributed into OSPF and that these routes are advertised by OSPF.
Note  It is not necessary to include definitions of all areas in an OSPF autonomous system in the configuration of all routers in the autonomous system. You must only define the directly connected areas. In the example that follows, routes in Area 0 are learned by the routers in Area 1 (Router A and Router B) when the area border router (Router C) injects summary link state advertisements (LSAs) into Area 1.

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

Router A—Internal Router
interface ethernet 1
ip address 131.108.1.1 255.255.255.0
router ospf 109
network 131.108.0.0 0.0.255.255 area 1

Router B—Internal Router
interface ethernet 2
ip address 131.108.1.2 255.255.255.0
router ospf 109
network 131.108.0.0 0.0.255.255 area 1

Router C—ABR
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 Router
interface ethernet 4
ip address 10.0.0.4 255.0.0.0

interface serial 1
ip address 131.108.2.4 255.255.255.0

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

Router E—ASBR
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
Example 4: Complex OSPF Configuration

The following example configuration accomplishes several tasks in setting up an ABR. 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 34 illustrates the network address ranges and area assignments for the interfaces.

Figure 34  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:

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

OSPF is on network 131.119.0.0:

```plaintext
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 ospf 201 metric 1

IGRP autonomous system 200 is on 131.119.0.0:

```plaintext
router igrp 200
network 131.119.0.0
!
! RIP for 192.42.110
!
router rip
network 192.42.110.0
redistribute igrp 200 metric 1
redistribute ospf 201 metric 1
```
BGP Route Advertisement and Redistribution Examples

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

Example 1: Simple BGP Route Advertisement

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

Configuration for Router R1

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

Configuration for Router R2

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

Example 2: Mutual Route Redistribution

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

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

Configuration for Router R1

```plaintext
router bgp 1
 network 131.108.0.0
 neighbor 192.5.10.1 remote-as 2
 ! External peer or neighbor
 neighbor 192.5.10.15 remote-as 1
 ! Same AS; therefore internal neighbor
 neighbor 192.5.10.24 remote-as 3
```
! A second External neighbor
redistribute igrp 1
distribute-list 1 out igrp 1
!
! All networks that should be
! advertised from R1 are
! controlled with access lists:
!
access-list 1 permit 131.108.0.0
access-list 1 permit 150.136.0.0
access-list 1 permit 128.125.0.0
!
router igrp 1
network 131.108.0.0
network 192.5.10.0
redistribute bgp 1

Default Metric Values Redistribution Example
The following example shows a router in autonomous system 109 using both RIP and IGRP. The
eexample advertises IGRP-derived routes using the RIP protocol and assigns the IGRP-derived routes
a RIP metric of 10.
router rip
default-metric 10
redistribute igrp 109

Route Map Examples
The examples in this section illustrate the use of redistribution, with and without route maps.
Examples from both the IP and CLNS routing protocols are given.
The following example redistributes all OSPF routes into IGRP:
router igrp 109
redistribute ospf 110

The following example redistributes RIP routes with a hop count equal to 1 into OSPF. These routes
will be redistributed into OSPF as external link state advertisements with a metric of 5, metric type
of Type 1, and a tag equal to 1.
router ospf 109
redistribute rip route-map rip-to-ospf
!
route-map rip-to-ospf permit
match metric 1
set metric 5
set metric-type type1
set tag 1

The following example redistributes OSPF learned routes with tag 7 as a RIP metric of 15:
router rip
redistribute ospf 109 route-map 5
!
route-map 5 permit
match tag 7
set metric 15
The following example redistributes OSPF intra-area and interarea routes with next-hop routers on serial interface 0 into BGP with an INTER_AS metric of 5:

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

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

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

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

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

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

```
router rip
redistribute ospf 109 route-map default
!
route-map default permit
match ip address 1
set metric 1
!
access-list 1 permit 140.222.0.0 0.0.255.255
access-list 2 permit 0.0.0.0 0.0.0.0
```
Given the following configuration, a RIP learned route for network 160.89.0.0 and an ISO-IGRP learned route with prefix 49.0001.0002 will be redistributed into an IS-IS Level 2 LSP with a metric of 5:

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

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

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

See more route map examples in the sections “BGP Route Map Examples” and “BGP Community Examples with Route Maps.”

**Route Summarization Example**

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

```
interface Ethernet 0
  ip summary-address eigrp 1 10.0.0.0 255.0.0.0
!
router eigrp 1
  network 172.16.0.0
  no auto-summary
```
IGRP Feasible Successor Relationship Example

In Figure 35, 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 35 Assigning Metrics for IGRP Path Feasibility](image)

The feasibility test would work as follows:

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

- If both of the following two conditions are met, the route to network A through C2 will be included in C1’s routing table:
  - If \(m\) is greater than \(p\).
  - If the multiplier (value specified by the `variance` router configuration command) times \(m\) is greater than or equal to \(n\).

- The configuration for Router C1 would be as follows:

  ```
  router igrp 109
  variance 10
  ```

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

TCP MD5 Authentication for BGP Example

The following example specifies that the router and its BGP peer at 145.2.2.2 invoke MD5 authentication on the TCP connection between them:

```
router bgp 109
neighbor 145.2.2.2 password v61ne0qkel33&
```
Autonomous System within EGP Example

The following example illustrates a typical configuration for an EGP router process. The router is in autonomous system 109 and is peering with routers in autonomous system 164, as shown in Figure 36. It will advertise the networks 131.108.0.0 and 192.31.7.0 to the router in autonomous system 164, 10.2.0.2. The information sent and received from peer routers can be filtered in various ways, including blocking information from certain routers and suppressing the advertisement of specific routes.

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

**Figure 36** Router in AS 164 Peers with Router in AS 109