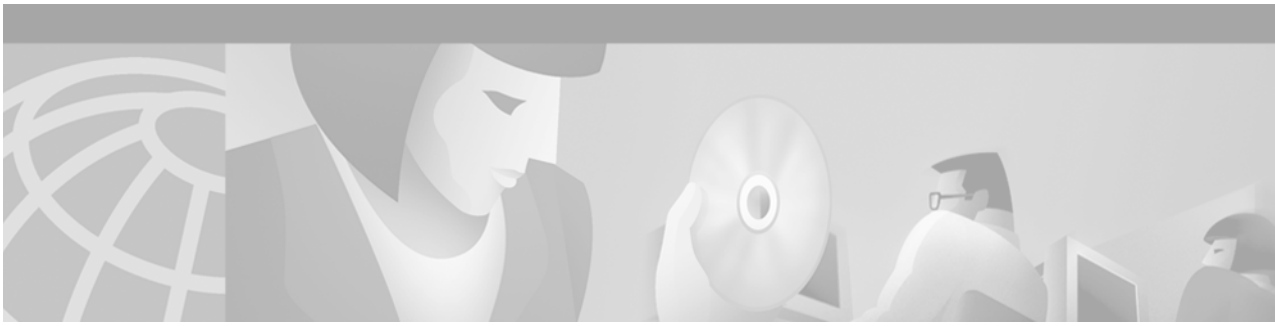


Configuring IP Routing Protocol-Independent Features

This chapter describes how to configure IP routing protocol-independent features. For a complete description of the IP routing protocol-independent commands in this chapter, refer to the "IP Routing Protocol-Independent Commands" chapter of the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols* publication. To locate documentation of other commands in this chapter, use the command reference master index, or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter in this book.

Protocol-Independent Feature Task List

Previous chapters addressed configurations of specific routing protocols. To configure optional protocol-independent features, perform any of the tasks described in the following sections:

- Using Variable-Length Subnet Masks (Optional)
- Configuring Static Routes (Optional)
- Specifying Default Routes (Optional)
- Changing the Maximum Number of Paths (Optional)
- Configuring Multi-Interface Load Splitting (Optional)
- Redistributing Routing Information (Optional)
- Filtering Routing Information (Optional)
- Enabling Policy Routing (PBR) (Optional)
- Managing Authentication Keys (Optional)
- Monitoring and Maintaining the IP Network (Optional)

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

Using Variable-Length Subnet Masks

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

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

Note Consider your decision to use VLSMs carefully. You can easily make mistakes in address assignments and you will generally find it more difficult to monitor your network using VLSMs.

Note 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 "Variable-Length Subnet Mask Example" section at the end of this chapter for an example of using VLSMs.

Configuring 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 useful for specifying a gateway of last resort to which all unroutable packets will be sent.

To configure a static route, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip route prefix mask {ip-address | interface-type interface-number} [distance] [tag tag] [permanent]`	Establishes a static route.

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 9. 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.

Table 9 Dynamic Routing Protocol Default Administrative Distances

Route Source	Default Distance
Connected interface	0
Static route	1

Table 9 Dynamic Routing Protocol Default Administrative Distances (continued)

Route Source	Default Distance
Enhanced IGRP (EIGRP) summary route	5
Exterior Border Gateway Protocol (BGP)	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EIGRP external route	170
Interior BGP	200
Unknown	255

Static routes that point to an interface will be advertised via RIP, IGRP, and other dynamic routing protocols, regardless of whether **redistribute static** router configuration commands were specified for those routing protocols. These static routes are advertised 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 address of the forwarding router in a static route, the static route is removed from the IP routing table.

Specifying 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.

Specifying 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, the router 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 a router can generate its own default 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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip default-network network-number`	Specifies a default network.

Understanding 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** global configuration command. In this usage, the **ip default-network** command 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.

Changing the Maximum Number of Paths

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

The range of maximum paths is one to six paths. To change the maximum number of parallel paths allowed, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# maximum-paths maximum`	Configures the maximum number of parallel paths allowed in a routing table.

Configuring Multi-Interface Load Splitting

Multi-interface load splitting allows you to efficiently control traffic that travels across multiple interfaces to the same destination. The **traffic-share min** router configuration command specifies that if multiple paths are available to the same destination, only paths with the minimum metric will be installed in the routing table. The number of paths allowed is never more than six. For dynamic routing

protocols, the number of paths is controlled by the **maximum-paths** router configuration command. The static route source can always install six paths. If more paths are available, the extra paths are discarded. If some installed paths are removed from the routing table, pending routes are added automatically.

When the **traffic-share min** command is used with the **across-interfaces** keyword, an attempt is made to use as many different interfaces as possible to forward traffic to the same destination. When the maximum path limit has been reached and a new path is installed, the router compares the installed paths. For example, if path X references the same interface as path Y and the new path uses a different interface, path X is removed and the new path is installed.

To configure traffic that is distributed among multiple routes of unequal cost for equal cost paths across multiple interfaces, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# traffic-share min {across-interfaces}`	Configures multi-interface load splitting across different interfaces with equal cost paths.

Redistributing 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 IGRP-derived routes using RIP, or to readvertise static routes using the IGRP protocol. Redistributing information from one routing protocol to another applies to all of the IP-based routing protocols.

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

The following four tables list tasks associated with route redistribution. 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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# route-map map-tag [permit | deny] [sequence-number]`	Defines any route maps needed to control redistribution.

One or more **match** commands and one or more **set** commands typically follow a **route-map** global configuration 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, use at least one of the following commands in route-map configuration mode, as needed:

Command	Purpose
`Router(config-route-map)# match as-path path-list-number`	Matches a BGP autonomous system path access list.
`Router(config-route-map)# match community-list community-list-number [exact]`	Matches a BGP community list.

Command	Purpose
Router(config-route-map)# **match ip address** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*]	Matches a standard access list.
Router(config-route-map)# **match metric** *metric-value*	Matches the specified metric.
Router(config-route-map)# **match ip next-hop** {*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*]	Matches a next-hop router address passed by one of the access lists specified.
Router(config-route-map)# **match tag** *tag-value* [*tag-value*]	Matches the specified tag value.
Router(config-route-map)# **match interface** *interface-type interface-number* [*interface-type interface-number*]	Matches the specified next hop route out one of the interfaces specified.
Router(config-route-map)# **match ip route-source** {*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*]	Matches the address specified by the specified advertised access lists.
Router(config-route-map)# **match route-type** {**local** | **internal** | **external** [**type-1** | **type-2**] | **level-1** | **level-2**}	Matches the specified route type.

One or more **match** commands and one or more **set** commands should follow a **route-map** router configuration command. To define conditions for redistributing routes from one routing protocol into another, use at least one of the following commands in route-map configuration mode as needed:

Command	Purpose
Router(config-route-map)# **set community** {*community-number* [**additive**]} | **none**	Sets the communities attribute.
Router(config-route-map)# **set dampening** *halflife reuse suppress max-suppress-time*	Sets BGP route dampening factors.
Router(config-route-map)# **set local-preference** *number-value*	Assigns a value to a local BGP path.
Router(config-route-map)# **set weight** *weight*	Specifies the BGP weight for the routing table.
Router(config-route-map)# **set origin** {**igp** | **egp** *as-number* | **incomplete**}	Sets the BGP origin code.
Router(config-route-map)# **set as-path** {**tag** | **prepend** *as-path-string*}	Modifies the BGP autonomous system path.
Router(config-route-map)# **set next-hop** *next-hop*	Specifies the address of the next hop.
Router(config-route-map)# **set automatic-tag**	Enables automatic computing of the tag table.
Router(config-route-map)# **set level** {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**}	Specifies the areas in which to import routes.
Router(config-route-map)# **set metric** *metric-value*	Sets the metric value to give the redistributed routes (for any protocol except IGRP or Enhanced IGRP [EIGRP]).
Router(config-route-map)# **set metric** *bandwidth delay reliability loading mtu*	Sets the metric value to give the redistributed routes (for IGRP or EIGRP only).
Router(config-route-map)# **set metric-type** {**internal** | **external** | **type-1** | **type-2**}	Sets the metric type to give redistributed routes.

Command	Purpose
`Router(config-route-map)# set metric-type internal`	Sets the Multi Exit Discriminator (MED) value on prefixes advertised to Exterior BGP neighbor to match the Interior Gateway Protocol (IGP) metric of the next hop.
`Router(config-route-map)# set tag tag-value`	Sets the tag value to associate with the redistributed routes.

See the "BGP Route Map Examples" section in the "Configuring BGP" chapter for examples of BGP route maps. See the "BGP Community with Route Maps Examples" section in the "Configuring BGP" chapter for examples of BGP communities and route maps.

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

	Command	Purpose
Step 1	`Router(config-router)# redistribute protocol [process-id] {level-1 | level-1-2 | level-2} [metric metric-value] [metric-type type-value] [match internal | external type-value] [tag tag-value] [route-map map-tag] [subnets]`	Redistributes routes from one routing protocol to another routing protocol.
Step 2	`Router(config-router)# default-metric number`	Causes the current routing protocol to use the same metric value for all redistributed routes (BGP, OSPF, RIP).
Step 3	`Router(config-router)# default-metric bandwidth delay reliability loading mtu`	Causes the IGRP or Enhanced IGRP (EIGRP) routing protocol to use the same metric value for all non-IGRP redistributed routes.
Step 4	`Router(config-router)# no default-information {in | out}`	Disables the redistribution of default information between IGRP processes, which is enabled by default.

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.

Understanding 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).
- 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.

Filtering Routing Information

To filter routing protocol information performing the tasks in the following sections. The tasks in the first section are required; the tasks in the remaining sections are optional:

- Preventing Routing Updates Through an Interface (Required)
- Controlling the Advertising of Routes in Routing Updates (Optional)
- Controlling the Processing of Routing Updates (Optional)
- Filtering Sources of Routing Information (Optional)

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

Preventing 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. Keeping routing update messages from being sent through a router interface prevents other systems on the interface from learning about routes dynamically. This feature applies to all IP-based routing protocols except BGP.

OSPF and IS-IS behave somewhat differently. 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, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# passive-interface` *interface-type interface-number*	Suppresses the sending of routing updates through the specified interface.

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

Configuring Default Passive Interfaces

In Internet service provider (ISP) and large enterprise networks, many of the distribution routers have more than 200 interfaces. Before the introduction of the Default Passive Interface feature, there were two possibilities for obtaining routing information from these interfaces:

- Configure a routing protocol such as OSPF on the backbone interfaces and redistribute connected interfaces.
- Configure the routing protocol on all interfaces and manually set most of them as passive.

Network managers may not always be able to summarize type 5 link-state advertisements (LSAs) at the router level where redistribution occurs, as in the first possibility. Thus, a large number of type 5 LSAs can be flooded over the domain.

In the second possibility, large type 1 LSAs might be flooded into the area. The Area Border Router (ABR) creates type 3 LSAs, one for each type 1 LSA, and floods them to the backbone. It is possible, however, to have unique summarization at the ABR level, which will inject only one summary route into the backbone, thereby reducing processing overhead.

The prior solution to this problem was to configure the routing protocol on all interfaces and manually set the **passive-interface** router configuration command on the interfaces where adjacency was not desired. But in some networks, this solution meant coding 200 or more passive interface statements. With the Default Passive Interface feature, this problem is solved by allowing all interfaces to be set as passive by default using a single **passive-interface default** command, then configuring individual interfaces where adjacencies are desired using the **no passive-interface** command.

Thus, the Default Passive Interface feature simplifies the configuration of distribution routers and allows the network manager to obtain routing information from the interfaces in large ISP and enterprise networks.

To set all interfaces as passive by default and then activate only those interfaces that need to have adjacencies set, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router protocol`	Configures the routing protocol on the network.
Step 2	`Router(config-router)# passive-interface default`	Sets all interfaces as passive by default.
Step 3	`Router(config-router)# no passive-interface interface-type`	Activates only those interfaces that need to have adjacencies set.
Step 4	`Router(config-router)# network network-address [options]`	Specifies the list of networks for the routing process. The *network-address* argument is an IP address written in dotted decimal notation—172.24.101.14, for example.

See the section "Default Passive Interface Example" at the end of this chapter for an example of a default passive interface.

To verify that interfaces on your network have been set to passive, you could enter a network monitoring command such as the **show ip ospf interface** EXEC command, or you could verify the interfaces you enabled as active using a command such as the **show ip interface** EXEC command.

Controlling the Advertising of Routes in Routing Updates

To prevent other routers from learning one or more routes, you can suppress routes from being advertised in routing updates. Suppressing routes in route updates prevents other routers from learning the interpretation of a particular device of one or more routes. 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, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# distribute-list {access-list-number | access-list-name} out [interface-name | routing-process | as-number]`	Permits or denies routes from being advertised in routing updates depending upon the action listed in the access list.

Controlling the Processing of Routing Updates

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

Command	Purpose
`Router(config-router)# distribute-list {access-list-number | access-list-name} in [interface-type interface-number]`	Suppresses routes listed in updates from being processed.

Filtering Sources of Routing Information

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

To filter sources of routing information, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# distance {ip-address {wildcard-mask}} [ip-standard-list] [ip-extended]`	Filters on routing information sources.

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 9 shows the default administrative distance for various routing information sources.

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.

For an example of filtering on sources of routing information, see the section "Administrative Distance Examples" later in this chapter.

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.

Note The weight of a route can no longer be set with the **distance** command. To set the weight for a route, use a route-map.

Enabling Policy Routing (PBR)

Policy routing (or "policy-based routing" [PBR]) 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. Possible applications for policy routing are to provide equal access, protocol-sensitive routing, source-sensitive routing, routing based on interactive versus batch traffic, and 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 task tables.

To enable policy routing on an interface, indicate which route map the router should use by using the following command in interface configuration mode. A packet arriving on the specified interface will be subject to policy routing except when its destination IP address is the same as the IP address of the router's interface. This command disables fast switching of all packets arriving on this interface.

Command	Purpose
`Router(config-if)# ip policy route-map map-tag`	Identifies the route map to use for policy routing.

To define the route map to be used for policy routing, use the following command in global configuration mode:

Command	Purpose
`Router(config)# route-map map-tag [permit | deny] [sequence-number]`	Defines a route map to control where packets are output.

To define the criteria by which packets are examined to learn if they will be policy-routed, use either one or both of the following commands in route-map configuration mode. No match clause in the route map indicates all packets.

Command	Purpose
`Router(config-route-map)# match length minimum-length maximum-length`	Matches the Level 3 length of the packet.
`Router(config-route-map)# match ip address {access-list-number | access-list-name} [access-list-number | access-list-name]`	Matches the destination IP address that is permitted by one or more standard or extended access lists.

To set the precedence and specify where the packets that pass the match criteria are output, use the following commands in route-map configuration mode:

	Command	Purpose
Step 1	`Router(config-route-map)# set ip precedence number | name`	Sets the precedence value in the IP header.
Step 2	`Router(config-route-map)# set ip next-hop ip-address [ip-address]`	Specifies the next hop to which to route the packet. (It must be an adjacent router).
Step 3	`Router(config-route-map)# set interface interface-type interface-number [... interface-type interface-number]`	Specifies the output interface for the packet.
Step 4	`Router(config-route-map)# set ip default next-hop ip-address [ip-address]`	Specifies the next hop to which to route the packet, if there is no explicit route for this destination. **Note** Like the **set ip next-hop** command, the **set ip default next-hop** command needs to specify an adjacent router.
Step 5	`Router(config-route-map)# set default interface interface-type interface-number [... interface-type interface-number]`	Specifies the output interface for the packet, if there is no explicit route for this destination.

Note The **set ip next-hop** and **set ip default next-hop** are similar commands but have a different order of operations. Configuring the **set ip next-hop** command causes the system to use policy routing first and then use the routing table. Configuring the **set ip default next-hop** causes the system to use the routing table first and then policy route the specified next hop.

The precedence setting in the IP header determines whether, during times of high traffic, the packets will be treated with more or less precedence than other packets. By default, the Cisco IOS software leaves this value untouched; the header remains with the precedence value it had.

The precedence bits in the IP header can be set in the router when policy routing is enabled. When the packets containing those headers arrive at another router, the packets are ordered for transmission according to the precedence set, if the queueing feature is enabled. The router does not honor the precedence bits if queueing is not enabled; the packets are sent in FIFO order.

You can change the precedence setting, using either a number or name. The names came from RFC 791, but are evolving. You can enable other features that use the values in the **set ip precedence** route-map configuration command to determine precedence. Table 10 lists the possible numbers and their corresponding name, from least important to most important.

Table 10 IP Precedence Values

Number	Name
0	routine
1	priority
2	immediate
3	flash
4	flash-override
5	critical
6	internet
7	network

The **set** commands can be used 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.

Preverifying Next-Hop Availability

If the router is policy routing packets to the next hop and the next hop happens to be down, the router will try unsuccessfully to use Address Resolution Protocol (ARP) for the next hop (which is down). This behavior will continue forever.

To prevent this situation, you can configure the router to first verify that the next hops of the route map are CDP neighbors of the router before routing to that next hop.

This task is optional because some media or encapsulations do not support CDP, or it may not be a Cisco device that is sending the router traffic

To configure the route-map policy to verify that the next hop is available before the router attempts to route traffic to it, use the following command in route-map configuration mode:

Command	Purpose
`Router(config-route-map)# set ip next-hop verify-availability`	Causes the router to confirm that the next hop, specified in the route map configuration, are active and available. • This command relies on CDP to determine if the next hop is an active CDP neighbor. • If this command is not used, and the next hop is not available, the traffic will remain forever unrouted. • If this command is used, and the next hop is not a CDP neighbor, the router looks to the subsequent next hop, if there is one. If a subsequent next-hop is not defined, the packets simply are not policy routed

The **set ip next-hop verify-availability** has the following restrictions:

- It can cause some performance degradation due to CDP database lookup overhead per packet.
- CDP must be enabled on the interface.

- The directly connected next hop must be a Cisco device with CDP enabled.
- It is not supported for use in conjunction with dCEF, due to the dependency of the CDP neighbor database.

If you want to selectively verify availability of only some next hops, you can configure different route-map entries (under the same route map name) with different criteria (using access list matching or packet size matching), and use the **set ip next-hop verify-availability** configuration command selectively.

Displaying Route-Map Policy Information

To display the cache entries in the policy route cache, use the **show ip cache policy** EXEC command.

To display the route map Inter Processor Communication (IPC) message statistics in the Route Processor (RP) or Versatile Interface Processor (VIP), use the following command in EXEC mode:

Command	Purpose
`Router# show route-map ipc`	Displays the route map IPC message statistics in the RP or VIP.

If you want policy routing to be fast switched, see the following section "Enabling Fast-Switched Policy Routing."

See the "Policy Routing Example" section at the end of this chapter for an example of policy routing.

Note For new policy-based routing (PBR) features in 12.4, see the following modules:

- PBR Support for Multiple Tracking Options
- PBR Recursive Next Hop

Enabling Fast-Switched Policy Routing

IP policy routing can now be fast switched. Prior to fast-switched policy routing, policy routing could only be process switched, which meant that on most platforms, the switching rate was approximately 1000 to 10,000 packets per second. Such rates were not fast enough for many applications. Users that need policy routing to occur at faster speeds can now implement policy routing without slowing down the router.

Fast-switched policy routing supports all of the **match** commands and most of the **set** commands, except for the following restrictions:

- The **set ip default** command is not supported.
- The **set interface** command is supported only over point-to-point links, unless a route cache entry exists using the same interface specified in the **set interface** command in the route map. Also, at the process level, the routing table is consulted to determine if the interface is on a reasonable path to the destination. During fast switching, the software does not make this check. Instead, if the packet matches, the software blindly forwards the packet to the specified interface.

Policy routing must be configured before you configure fast-switched policy routing. Fast switching of policy routing is disabled by default. To have policy routing be fast switched, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip route-cache policy`	Enables fast switching of policy routing.

Enabling 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 using the following command in global configuration mode. All packets originating on the router will then be subject to local policy routing.

Command	Purpose
`Router(config)# ip local policy route-map map-tag`	Identifies the route map to use for local policy routing.

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

Managing Authentication Keys

Key management is a method of controlling authentication keys used by routing protocols. Not all protocols can use key management. Authentication keys are available for Director Response Protocol (DRP) Agent, Enhanced IGRP (EIGRP), and RIP Version 2.

Before you manage authentication keys, authentication must be enabled. See the appropriate protocol chapter to learn how to enable authentication for that protocol.

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. Each key has its own key identifier (specified with the **key chain** configuration command), which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and Message Digest 5 (MD5) authentication key in use.

You can configure multiple keys with lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in order from lowest to highest, and uses the first valid key it encounters. The lifetimes allow for overlap during key changes. Note that the router must know the time. Refer to the Network Time Protocol (NTP) and calendar commands in the "Performing Basic System Management" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

To manage authentication keys, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#key chain name-of-chain`	Identifies a key chain.
Step 2	`Router(config-keychain)# key number`	Identifies the key number in key chain configuration mode.
Step 3	`Router(config-keychain-key)# key-string text`	Identifies the key string in key chain configuration mode.

	Command	Purpose
Step 4	`Router(config-keychain-key)# accept-lifetime start-time {infinite | end-time | duration seconds}]`	Specifies the time period during which the key can be received.
Step 5	`Router(config-keychain-key)# send-lifetime start-time {infinite | end-time | duration seconds}`	Specifies the time period during which the key can be sent.

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

Monitoring and Maintaining 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.

Clearing Routes from the IP Routing Table

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

To clear one or more routes from the IP routing table, use the following command in EXEC mode:

Command	Purpose
`Router# clear ip route {network [mask] | *}`	Clears one or more routes from the IP routing table.

Displaying 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 packets leaving your device are taking through the network.

To display various routing statistics, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show ip cache policy`	Displays the cache entries in the policy route cache.
`Router# show ip local policy`	Displays the local policy route map if one exists.
`Router# show ip policy`	Displays policy route maps.
`Router# show ip protocols`	Displays the parameters and current state of the active routing protocol process.
`Router# show ip route [address [mask] [longer-prefixes]] | [protocol [process-id]]`	Displays the current state of the routing table.
`Router# show ip route summary`	Displays the current state of the routing table in summary form.
`Router# show ip route supernets-only`	Displays supernets.

Command	Purpose
Router# **show key chain** [*name-of-chain*]	Displays authentication key information.
Router# **show route-map** [*map-name*]	Displays all route maps configured or only the one specified.

IP Routing Protocol-Independent Configuration Examples

The following sections provide routing protocol-independent configuration examples:

- Variable-Length Subnet Mask Example
- Overriding Static Routes with Dynamic Protocols Example
- Administrative Distance Examples
- Static Routing Redistribution Example
- IGRP Redistribution Example
- RIP and IGRP Redistribution Example
- EIGRP Redistribution Examples
- RIP and EIGRP Redistribution Examples
- OSPF Routing and Route Redistribution Examples
- Default Metric Values Redistribution Example
- Policy Routing (Route Map) Examples
- Passive Interface Examples
- Policy Routing Example
- Key Management Examples

Variable-Length Subnet Mask Example

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.

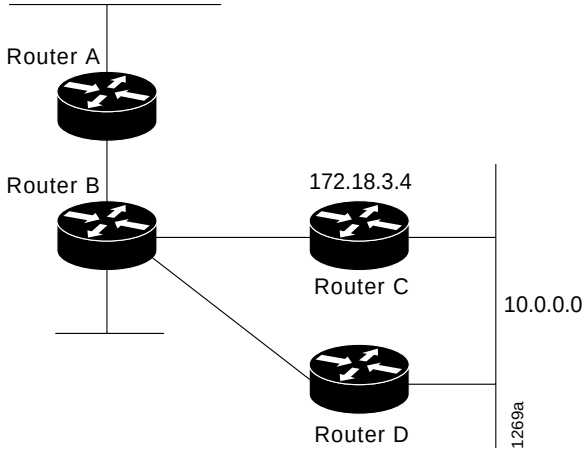
```
interface ethernet 0
 ip address 172.17.1.1 255.255.255.0
! 8 bits of host address space reserved for ethernets

interface serial 0
 ip address 172.17.254.1 255.255.255.252
! 2 bits of address space reserved for serial lines

! Router is configured for OSPF and assigned AS 1
router ospf 1
! Specifies the network directly connected to the router
 network 172.17.0.0 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 172.18.3.4 if a route with an administrative distance less than 110 is not available. Figure 62 illustrates this example. 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 10.0.0.0 255.0.0.0 172.18.3.4 110
```

Figure 62 Overriding Static Routes



Administrative Distance Examples

In the following example, the **router igrp** global configuration command sets up IGRP routing in autonomous system 1. The **network** router configuration commands specify IGRP routing on networks 192.168.7.0 and 172.16.0.0. 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 192.168.7.0. The third **distance** command sets the administrative distance to 120 for the router with the address 172.16.1.3.

```
router igrp 1
 network 192.168.7.0
 network 172.16.0.0
 distance 255
 distance 90 192.168.7.0 0.0.0.255
 distance 120 172.16.1.3 0.0.0.0
```

The following example assigns the router with the address 192.168.7.18 an administrative distance of 100 and all other routers on subnet 192.168.7.0 an administrative distance of 200:

```
 distance 100 192.168.7.18 0.0.0.0
 distance 200 192.168.7.0 0.0.0.255
```

However, if you reverse the order of these two commands, all routers on subnet 192.168.7.0 are assigned an administrative distance of 200, including the router at address 192.168.7.18:

```
 distance 200 192.168.7.0 0.0.0.255
 distance 100 192.168.7.18 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
```

Static Routing Redistribution Example

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

```
ip route 192.168.2.0 255.255.255.0 192.168.7.65
ip route 192.168.5.0 255.255.255.0 192.168.7.65
ip route 172.16.0.0 255.255.255.0 192.168.7.65
access-list 3 permit 192.168.2.0
access-list 3 permit 192.168.5.0
!
router igrp 1
 network 192.168.7.0
 default-metric 10000 100 255 1 1500
 redistribute static
 distribute-list 3 out static
```

IGRP Redistribution Example

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

Suppose that the router has one IGRP routing process for network 10.0.0.0 in autonomous system 71 and another IGRP routing process for network 192.168.7.0 in autonomous system 1, as the following commands specify:

```
router igrp 71
 network 10.0.0.0
router igrp 1
 network 192.168.7.0
```

To transfer a route to 192.168.7.0 into autonomous system 71 (without passing any other information about autonomous system 1), use the command in the following example:

```
router igrp 71
 redistribute igrp 1
 distribute-list 3 out igrp 1
access-list 3 permit 192.168.7.0
```

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, 172.16.0.0, which uses IGRP as the routing protocol. The goal in this case is to advertise the networks in the university network to the routers on the regional network. The commands for the interconnecting router are listed in the example that follows:

```
router igrp 1
 network 172.16.0.0
 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 172.16.0.0 (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 number of entries in each outgoing update. The access list prevents unauthorized advertising of university routes to the regional network.

EIGRP Redistribution Examples

Each Enhanced IGRP (EIGRP) routing process provides routing information to only one autonomous system. The Cisco IOS software must run a separate EIGRP process and maintain a separate routing database for each autonomous system that it services. However, you can transfer routing information between these routing databases.

Suppose that the software has one EIGRP routing process for network 10.0.0.0 in autonomous system 71 and another EIGRP routing process for network 192.168.7.0 in autonomous system 1, as the following commands specify:

```
router eigrp 71
 network 10.0.0.0
router eigrp 1
 network 192.168.7.0
```

To transfer a route from 192.168.7.0 into autonomous system 71 (without passing any other information about autonomous system 1), use the command in the following example:

```
router eigrp 71
 redistribute eigrp 1 route-map 1-to-71
 route-map 1-to-71 permit
 match ip address 3
 set metric 10000 100 1 255 1500
access-list 3 permit 192.168.7.0
```

The following example is an alternative way to transfer a route to 192.168.7.0 into autonomous system 71. Unlike the previous configuration, this one does not allow you to arbitrarily set the metric.

```
router eigrp 71
 redistribute eigrp 1
 distribute-list 3 out eigrp 1
access-list 3 permit 192.168.7.0
```

RIP and EIGRP Redistribution Examples

This section provides a simple RIP redistribution example and a complex redistribution example between Enhanced IGRP (EIGRP) and BGP.

Simple 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, 172.16.0.0, which uses Enhanced IGRP (EIGRP) 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 1
 network 172.16.0.0
 redistribute rip
 default-metric 10000 100 255 1 1500
 distribute-list 10 out rip
```

In this example, the **router** global configuration command starts an EIGRP routing process. The **network** router configuration command specifies that network 172.16.0.0 (the regional network) is to send and receive EIGRP 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 EIGRP 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.

Complex Redistribution Example

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

Suppose that BGP is running on a router somewhere else in autonomous system 50000 and that the BGP routes are injected into EIGRP 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 EIGRP.

```
! Configuration for router R1:
router bgp 50000
 network 172.18.0.0
 neighbor 192.168.10.1 remote-as 2
 neighbor 192.168.10.15 remote-as 1
 neighbor 192.168.10.24 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:
!
access-list 1 permit 172.18.0.0
access-list 1 permit 172.16.0.0
access-list 1 permit 172.17.0.0
!
router eigrp 1
 network 172.18.0.0
 network 192.168.10.0
```

```
 redistribute bgp 50000
```

OSPF Routing and Route Redistribution Examples

OSPF typically requires coordination among many internal routers, ABRs, and Autonomous System Boundary Routers (ASBRs). At a minimum, OSPF-based routers can be configured with all default parameter values, with no authentication, and with interfaces assigned to areas. Three types of examples follow:

- The first examples are simple configurations 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.

Basic OSPF Configuration Examples

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

```
interface ethernet 0
 ip address 172.16.1.1 255.255.255.0
 ip ospf cost 1
!
interface ethernet 1
 ip address 172.17.1.1 255.255.255.0
!
router ospf 9000
 network 172.16.0.0 0.0.255.255 area 0.0.0.0
 redistribute rip metric 1 subnets
!
router rip
 network 172.17.0.0
 redistribute ospf 9000
 default-metric 1
```

The following example illustrates the assignment of four area IDs to four IP address ranges. In the example, OSPF routing process 1 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, whereas area 0 enables OSPF for *all other* networks.

```
router ospf 1
 network 172.16.20.0 0.0.0.255 area 10.9.50.0
 network 172.16.0.0 0.0.255.255 area 2
 network 172.17.10.0 0.0.0.255 area 3
 network 0.0.0.0 255.255.255.255 area 0
!
! Ethernet interface 0 is in area 10.9.50.0:
interface ethernet 0
 ip address 172.16.20.5 255.255.255.0
!
! Ethernet interface 1 is in area 2:
interface ethernet 1
 ip address 172.16.1.5 255.255.255.0
!
! Ethernet interface 2 is in area 2:
interface ethernet 2
 ip address 172.17.2.5 255.255.255.0
```

```
!
! Ethernet interface 3 is in area 3:
interface ethernet 3
 ip address 172.18.10.5 255.255.255.0
!
! Ethernet interface 4 is in area 0:
interface ethernet 4
 ip address 172.19.1.1 255.255.255.0
!
! Ethernet interface 5 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 *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols* publication for more information.

Consider the first **network** command. Area ID 10.9.50.0 is configured for the interface on which subnet 172.18.20.0 is located. Assume that a match is determined for Ethernet interface 0. Ethernet interface 0 is attached to Area 10.9.50.0 only.

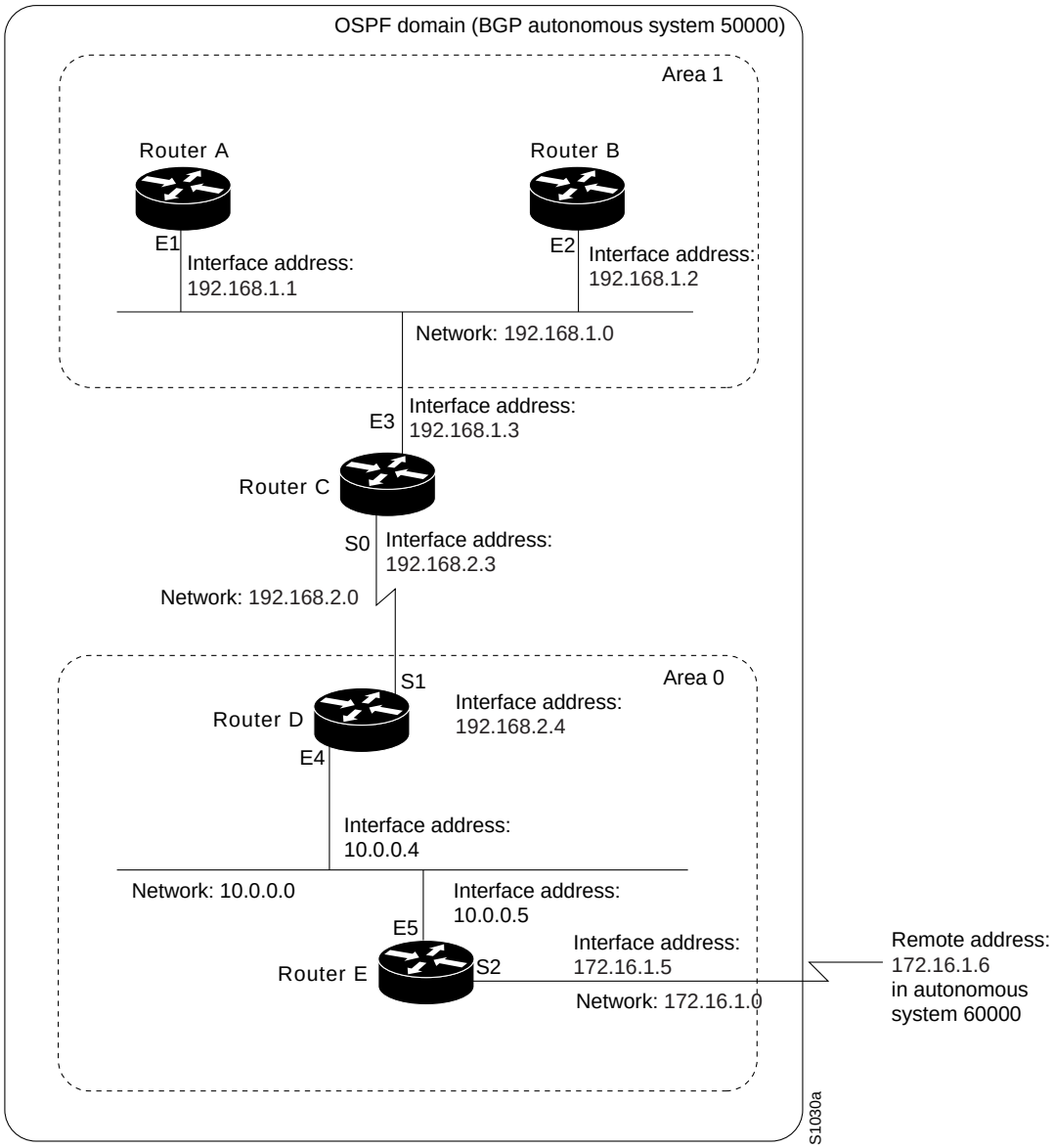
The second **network** command is evaluated next. For Area 2, the same process is then applied to all interfaces (except Ethernet interface 0). Assume that a match is determined for Ethernet interface 1. OSPF is then enabled for that interface and Ethernet 1 is attached to Area 2.

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

Internal Router, ABR, and ASBRs Configuration Example

Figure 63 provides a general network map that illustrates a sample configuration for several routers within a single OSPF autonomous system.

Figure 63 Example OSPF Autonomous System Network Map



In this configuration, five routers are configured in OSPF autonomous system 1:

- Router A and Router B are both internal routers within area 1.
- Router C is an OSPF ABR. 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 ASBR. 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 define only 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 ABR (Router C) injects summary LSAs into area 1.

Autonomous system 60000 is connected to the outside world via the BGP link to the external peer at IP address 172.16.1.6.

Following is the example configuration for the general network map shown in Figure 63.

Router A Configuration—Internal Router

```
interface ethernet 1
 ip address 192.168.1.1 255.255.255.0

router ospf 1
 network 192.168.1.0 0.0.0.255 area 1
```

Router B Configuration—Internal Router

```
interface ethernet 2
 ip address 192.168.1.2 255.255.255.0

router ospf 1
 network 192.168.1.0 0.0.0.255 area 1
```

Router C Configuration—ABR

```
interface ethernet 3
 ip address 192.168.1.3 255.255.255.0

interface serial 0
 ip address 192.168.2.3 255.255.255.0

router ospf 1
 network 192.168.1.0 0.0.0.255 area 1
 network 192.168.2.0 0.0.0.255 area 0
```

Router D Configuration—Internal Router

```
interface ethernet 4
 ip address 10.0.0.4 255.0.0.0

interface serial 1
 ip address 192.168.2.4 255.255.255.0

router ospf 1
 network 192.168.2.0 0.0.0.255 area 0
 network 10.0.0.0 0.255.255.255 area 0
```

Router E Configuration—ASBR

```
interface ethernet 5
 ip address 10.0.0.5 255.0.0.0

interface serial 2
 ip address 172.16.1.5 255.255.0.0

router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 redistribute bgp 50000 metric 1 metric-type 1
```

```
router bgp 50000
 network 192.168.0.0
 network 10.0.0.0
 neighbor 172.16.1.6 remote-as 60000
```

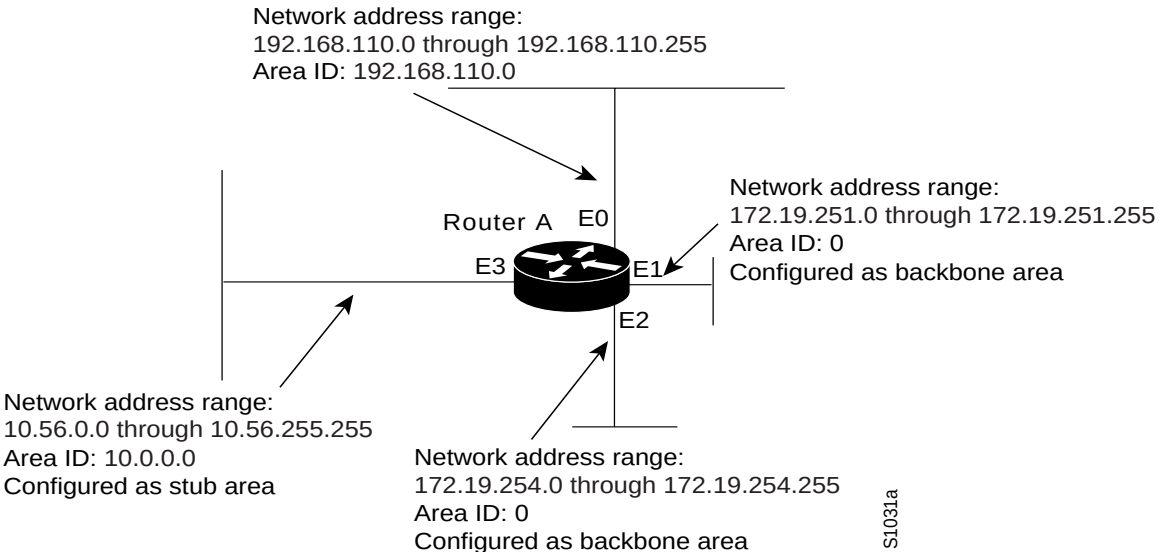
Complex OSPF Configuration Example

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 64 illustrates the network address ranges and area assignments for the interfaces.

Figure 64 Interface and Area Specifications for OSPF Configuration Example



The basic configuration tasks in this example are as follows:

- Configure address ranges for Ethernet interface 0 through Ethernet interface 3.
- 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 10.0.0.0. (Note that the **authentication** and **stub** options of the **area** router configuration command are specified with separate **area** command entries, but they can be merged into a single **area** command.)
- Specify the backbone area (area 0).

Configuration tasks associated with redistribution are as follows:

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

The following is an example OSPF configuration:

```
interface ethernet 0
 ip address 192.168.110.201 255.255.255.0
 ip ospf authentication-key abcdefgh
 ip ospf cost 10
!
interface ethernet 1
 ip address 172.19.251.201 255.255.255.0
 ip ospf authentication-key ijklmnop
 ip ospf cost 20
 ip ospf retransmit-interval 10
 ip ospf transmit-delay 2
 ip ospf priority 4
!
interface ethernet 2
 ip address 172.19.254.201 255.255.255.0
 ip ospf authentication-key abcdefgh
 ip ospf cost 10
!
interface ethernet 3
 ip address 10.0.0.201 255.255.0.0
 ip ospf authentication-key ijklmnop
 ip ospf cost 20
 ip ospf dead-interval 80
```

In the following configuration, OSPF is on network 172.19.0.0:

```
router ospf 1
 network 10.0.0.0 0.255.255.255 area 10.0.0.0
 network 192.168.110.0 0.0.0.255 area 192.168.110.0
 network 172.19.0.0 0.0.255.255 area 0
 area 0 authentication
 area 10.0.0.0 stub
 area 10.0.0.0 authentication
 area 10.0.0.0 default-cost 20
 area 192.168.110.0 authentication
 area 10.0.0.0 range 10.0.0.0 255.0.0.0
 area 192.168.110.0 range 192.168.110.0 255.255.255.0
 area 0 range 172.19.251.0 255.255.255.0
 area 0 range 172.19.254.0 255.255.255.0

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

In the following configuration IGRP autonomous system 1 is on 172.19.0.0:

```
router igrp 1
 network 172.19.0.0
!
! RIP for 192.168.110.0
!
router rip
 network 192.168.110.0
 redistribute igrp 1 metric 1
 redistribute ospf 201 metric 1
```

Default Metric Values Redistribution Example

The following example shows a router in autonomous system 1 using both RIP and IGRP. The example advertises IGRP-derived routes using RIP and assigns the IGRP-derived routes a RIP metric of 10.

```
router rip
 default-metric 10
 redistribute igrp 1
```

Policy Routing (Route Map) Examples

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

The following example redistributes all OSPF routes into IGRP:

```
router igrp 1
 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 LSAs with a metric of 5, metric a type of type 1, and a tag equal to 1.

```
router ospf 1
 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 1 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 50000
 redistribute ospf 1 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 type is OSPF external IP routes with tag 5; these routes are inserted into Level 2 IS-IS link-state packets (LSPs) with a metric of 5. The second type is ISO-IGRP derived CLNS prefix routes that match CLNS access list 2000; these routes will be redistributed into IS-IS as Level 2 LSPs with a metric of 30.

```
router isis
 redistribute ospf 1 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 1 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
```

Given the following configuration, a RIP learned route for network 172.18.0.0 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 172.18.0.0 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 type of reference is called *conditional default origination*. OSPF will originate the default route (network 0.0.0.0) with a type 2 metric of 5 if 172.20.0.0 is in the routing table.

```
route-map ospf-default permit
 match ip address 1
 set metric 5
 set metric-type type-2
!
access-list 1 172.20.0.0 0.0.255.255
!
router ospf 1
 default-information originate route-map ospf-default
```

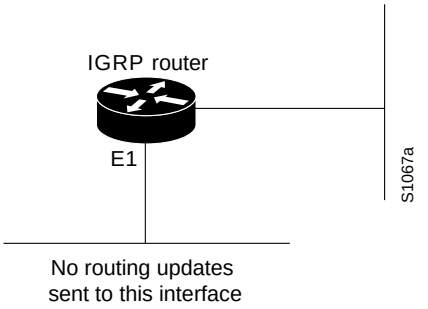
See more route map examples in the "BGP Route Map Examples" and "BGP Community with Route Maps Examples" sections of the 12.4 BGP documentation.

Passive Interface Examples

The following example configures Ethernet interface 1 as a passive interface under IGRP. Figure 65 shows the router topology. Routing updates are sent out all interfaces in the 192.168/16 network except for Ethernet interface 1.

```
 interface Ethernet 1
 ip address 192.168.0.1 255.255.0.0
router igrp 1
 network 192.168.0.0
 passive-interface Ethernet 1
```

Figure 65 Filtering IGRP Updates



In the following example, as in the first example, IGRP updates are sent out all interfaces in the 192.168/16 network except for Ethernet interface 1. However, in this configuration a neighbor statement is configured explicitly for the 192.168.0.2 neighbor. This neighbor statement will override the passive-interface configuration, and all interfaces in the 192.168/16 network, including Ethernet interface 1, will send routing advertisements to the 192.168.0.2 neighbor.

```
router igrp 1
 network 192.168.0.0
 passive-interface ethernet 1
 neighbor 192.18.0.2
```

The **passive-interface** command disables the transmission and receipt of EIGRP hello packets on an interface. Unlike IGRP or RIP, EIGRP sends hello packets in order to form and sustain neighbor adjacencies. Without a neighbor adjacency, EIGRP cannot exchange routes with a neighbor. Therefore, the **passive-interface** command prevents the exchange of routes on the interface. Although EIGRP does not send or receive routing updates on an interface configured with the **passive-interface** command, it still includes the address of the interface in routing updates sent out of other nonpassive interfaces.

Note For more information about configuring passive interfaces in EIGRP, see the *How Does the Passive Interface Feature Work in EIGRP?* document on cisco.com.

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 configuration 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 is typically 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 172.18.0.0:

```
interface ethernet 0
 ip address 172.18.1.1 255.255.255.0
interface ethernet 1
 ip address 172.18.2.1 255.255.255.0
interface ethernet 2
 ip address 172.18.3.1 255.255.255.0
!
router ospf 1
 network 172.18.0.0 0.0.255.255 area 0
```

If you do not want OSPF to run on 172.18.3.0, enter the following commands:

```
router ospf 1
 network 172.18.0.0 0.0.255.255 area 0
 passive-interface ethernet 2
```

Default Passive Interface Example

The following example configures the network interfaces, sets all interfaces that are running OSPF as passive, and then enables serial interface 0:

```
interface Ethernet 0
 ip address 172.19.64.38 255.255.255.0 secondary
 ip address 172.19.232.70 255.255.255.240
 no ip directed-broadcast
!
interface Serial 0
 ip address 172.24.101.14 255.255.255.252
 no ip directed-broadcast
 no ip mroute-cache
!
interface TokenRing0
 ip address 172.20.10.4 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 ring-speed 16
!
router ospf 1
 passive-interface default
 no passive-interface Serial0
 network 172.16.10.0 0.0.0.255 area 0
 network 172.19.232.0 0.0.0.255 area 4
 network 172.24.101.0 0.0.0.255 area 4
```

Policy Routing Example

The following example provides two sources with equal access to two different service providers. Packets that arrive on asynchronous interface 1 from the source 10.1.1.1 are sent to the router at 172.16.6.6 if the router has no explicit route for the destination of the packet. Packets that arrive from the source 172.17.2.2 are sent to the router at 192.168.7.7 if the router has no explicit route for the destination of the packet. All other packets for which the router has no explicit route to the destination are discarded.

```
access-list 1 permit ip 10.1.1.1
access-list 2 permit ip 172.17.2.2
!
```

```
interface async 1
 ip policy route-map equal-access
!
route-map equal-access permit 10
 match ip address 1
 set ip default next-hop 172.16.6.6
route-map equal-access permit 20
 match ip address 2
 set ip default next-hop 192.168.7.7
route-map equal-access permit 30
 set default interface null0
```

Key Management Examples

The following example configures a key chain named 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 discrepancy in the set time of the router. Likewise, the key birch immediately follows chestnut, and there is a 30-minute leeway on each side to handle time-of-day differences.

```
interface ethernet 0
 ip rip authentication key-chain trees
 ip rip authentication mode md5
!
router 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 named trees:

```
key chain trees
 key 1
 key-string willow
 key 2
 key-string chesnut
 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 172.16.232.147 255.255.255.240
 ip rip authentication key-chain trees
 media-type 10BaseT
!
interface Ethernet1
 no ip address
 shutdown
 media-type 10BaseT
```

```
interface Fddi0
 ip address 10.1.1.1 255.255.255.0
 no keepalive
!
interface Fddi1
 ip address 172.16.1.1 255.255.255.0
 ip rip send version 1
 ip rip receive version 1
 no keepalive
!
router rip
 version 2
 network 172.19.0.0
 network 10.0.0.0
 network 172.16.0.0
```