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 *Network Protocols Command Reference, Part 1*. To locate documentation of other commands in this chapter, use the command reference master index or search online.

Previous chapters addressed configurations of specific routing protocols. The following sections describe optional features that are protocol-independent:

- Use Variable-Length Subnet Masks
- Configure Static Routes
- Specify Default Routes
- Change the Maximum Number of Paths
- Redistribute Routing Information
- Filter Routing Information
- Enable Policy Routing
- Manage Authentication Keys
- Monitor and Maintain the IP Network

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

Use Variable-Length Subnet Masks

Enhanced IGRP, IS-IS, OSPF, 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.

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.

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

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

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 3. 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 3 Dynamic Routing Protocol Default Administrative Distances

Route Source	Default Distance
Connected interface	0
Static route	1
Enhanced IGRP summary route	5
External BGP	20
Internal Enhanced IGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EIGRP external route	170
Internal 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** 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.

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:

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

Understand 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.

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:

Task	Command
Configure the maximum number of parallel paths allowed in a routing table.	**maximum-paths** *maximum*

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 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:

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

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:

Task	Command
Match a BGP autonomous system path access list.	**match as-path** *path-list-number*
Match a BGP community list.	**match community-list** *community-list-number* [**exact**]
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*}

Task	Command
Match the specified route type.	**match route-type** {**local** | **internal** | **external** [**type-1** | **type-2**] | **level-1** | **level-2**}

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

Task	Command
Set the COMMUNITIES attribute.	**set community** {*community-number* [**additive**]} | **none**
Set BGP route dampening factors.	**set dampening** *halflife reuse suppress max-suppress-time*
Assign a value to a local BGP path.	**set local-preference** *value*
Specify the BGP weight for the routing table.	**set weight** *weight*
Set the BGP origin code.	**set origin** {**igp** | **egp** *as* | **incomplete**}
Modify the BGP autonomous system path.	**set as-path** {**tag** | **prepend** *as-path-string*}
Specify the address of the next hop.	**set next-hop** *next-hop*
Enable automatic computing of tag table.	**set automatic-tag**
For routes that are advertised into the specified area of the routing domain.	**set level** {**level-1** | **level-2** | level-1-2 | **stub-area** | **backbone**}
Set the metric value to give the redistributed routes (for any protocol except IGRP or IP Enhanced IGRP).	**set metric** *metric-value*
Set the metric value to give the redistributed routes (for IGRP or IP Enhanced IGRP only).	**set metric** *bandwidth delay reliability loading mtu*
Set the metric type to give redistributed routes.	**set metric-type** {**internal** | **external** | **type-1** | **type-2**}
Set the MED value on prefixes advertised to EBGP neighbor to match the IGP metric of the next hop.	**set metric-type internal**
Set the tag value to associate with the redistributed routes.	**set tag** *tag-value*

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, perform the following tasks in router configuration mode:

Task	Command
Redistribute routes from one routing protocol to another routing protocol.	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**match internal** | **external** *type-value*] [**tag** *tag-value*] [**route-map** *map-tag*] [**weight** *weight*] [**subnets**]
Cause the current routing protocol to use the same metric value for all redistributed routes (BGP, OSPF, RIP).	**default-metric** *number*

Task	Command
Cause the IGRP or Enhanced IGRP routing protocol to use the same metric value for all non-IGRP redistributed routes.	**default-metric** *bandwidth delay reliability loading mtu*
Disable the redistribution of default information between IGRP processes. This is enabled by default.	**no default-information** {**in** | **out**}

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

Understand 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.

Filter Routing Information

You can filter routing protocol information by performing the following tasks, each of which is described in this section:

- Prevent Routing Updates through an Interface
- Control the Advertising of Routes in Routing Updates
- Control the Processing of Routing Updates
- Filter Sources of Routing Information

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

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 a router interface. This is done to prevent other systems on an 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, perform the following task in router configuration mode:

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

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

Control the Advertising of Routes in Routing Updates

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

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

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

Control the Processing of Routing Updates

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

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

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

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

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

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 3 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" at the end of 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.

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.

Task	Command
Identify the route map to use for policy routing.	**ip policy route-map** *map-tag*

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

Task	Command
Define a route map to control where packets are output.	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]

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:

Task	Command
Match the Level 3 length of the packet.	**match length** *min max*
Match the destination IP address that is permitted by one or more standard or extended access lists.	**match ip address** {*access-list-number* | *name*} [...*access-list-number* | *name*]

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

Task	Command
Set the precedence value in the IP header.	**set ip precedence** *value*
Specify the next hop to which to route the packet. The next hop must be an adjacent router.	**set ip next-hop** *ip-address* [... *ip-address*]
Specify the output interface for the packet.	**set interface** *type number* [... *type number*]
Specify the next hop to which to route the packet, if there is no explicit route for this destination. The next hop must be an adjacent router.	**set ip default next-hop** *ip-address* [... *ip-address*]
Specify the output interface for the packet, if there is no explicit route for this destination.	**set default interface** *type number* [... *type number*]

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 queuing feature is enabled. The router does not honor the precedence bits if queuing is not enabled; the packets are sent in first in, first out 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** command to determine precedence. Table 4 lists the possible numbers and their corresponding name, from least important to most important.

Table 4 IP Precedence Values

Number	Name
0	**routine**
1	**priority**
2	**immediate**

Table 4 IP Precedence Values (Continued)

Number	Name
3	**flash**
4	**flash-override**
5	**critical**
6	**internet**
7	**network**

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.

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

If you want policy routing to be fast-switched, see the section “Enable Fast-Switched Policy Routing,” which follows.

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

Enable Fast-Switched Policy Routing

IP policy routing can now be fast-switched. Prior to this feature, policy routing could only be process switched, which meant that on most platforms, the switching rate was approximately 1,000 to 10,000 packets per second. This was not fast enough for many applications. Users who 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, perform the following task in interface configuration mode:

Task	Command
Enable fast switching of policy routing.	**ip route-cache policy**

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.

Task	Command
Identify the route map to use for local policy routing.	**ip local policy route-map** *map-tag*

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

Manage 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, IP Enhanced IGRP, and RIP Version 2.

Before you manage authentication keys, authentication must be enabled. See the appropriate protocol chapter to see 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** *number* command), which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and 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 NTP and calendar commands in the "Performing Basic System Management" chapter of the *Configuration Fundamentals Configuration Guide*.

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

Task	Command
Identify a key chain.	**key chain** *name-of-chain*
In key chain configuration mode, identify the key number.	**key** *number*
In key chain key configuration mode, identify the key string.	**key-string** *text*
Specify the time period during which the key can be received.	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
Specify the time period during which the key can be sent.	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}

Use the **show key chain** command to display key chain information. For examples of key management, see the "Manage Authentication Keys" 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 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, perform the following task in EXEC mode:

Task	Command
Clear one or more routes from the IP routing table.	**clear ip route** {*network* [*mask*] | *****}

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:

Task	Command
Display the cache entries in the policy route-cache.	**show ip cache policy**
Display the local policy route map, if any.	**show ip local policy**
Display policy route maps.	**show ip policy**
Display the parameters and current state of the active routing protocol process.	**show ip protocols**
Display the current state of the routing table.	**show ip route** [*address* [*mask*] [**longer-prefixes**]] | [*protocol* [*process-id*]]
Display the current state of the routing table in summary form.	**show ip route summary**
Display supernets.	**show ip route supernets-only**
Display authentication key information.	**show key chain** [*name*]
Display all route maps configured or only the one specified.	**show route-map** [*map-name*]

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
- IP Enhanced IGRP Redistribution Examples
- RIP and IP Enhanced IGRP Redistribution Examples
- OSPF Routing and Route Redistribution Examples
- Default Metric Values Redistribution Example
- 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.

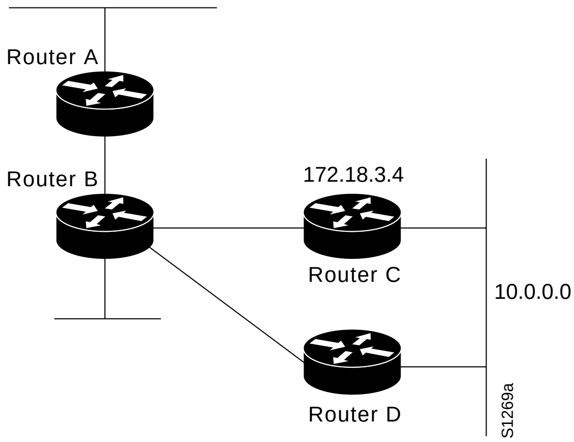
```
interface ethernet 0
 ip address 131.107.1.1 255.255.255.0
! 8 bits of host address space reserved for ethernets

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



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
```

Static Routing Redistribution Example

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

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

IGRP Redistribution Example

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

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

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

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

RIP and IGRP Redistribution Example

Consider a WAN at a university that uses RIP as an interior routing protocol. Assume that the university wants to connect its 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:

```
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:

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

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

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

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

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

RIP and IP Enhanced IGRP Redistribution Examples

This section provides a simple RIP redistribution example and a complex redistribution example between IP Enhanced IGRP and BGP.

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:
!
access-list 1 permit 131.108.0.0
access-list 1 permit 150.136.0.0
access-list 1 permit 128.125.0.0
!
router eigrp 1
 network 131.108.0.0
 network 192.5.10.0
 redistribute bgp 1
```

OSPF Routing and Route Redistribution Examples

OSPF typically requires coordination among many internal 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 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 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
```

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.

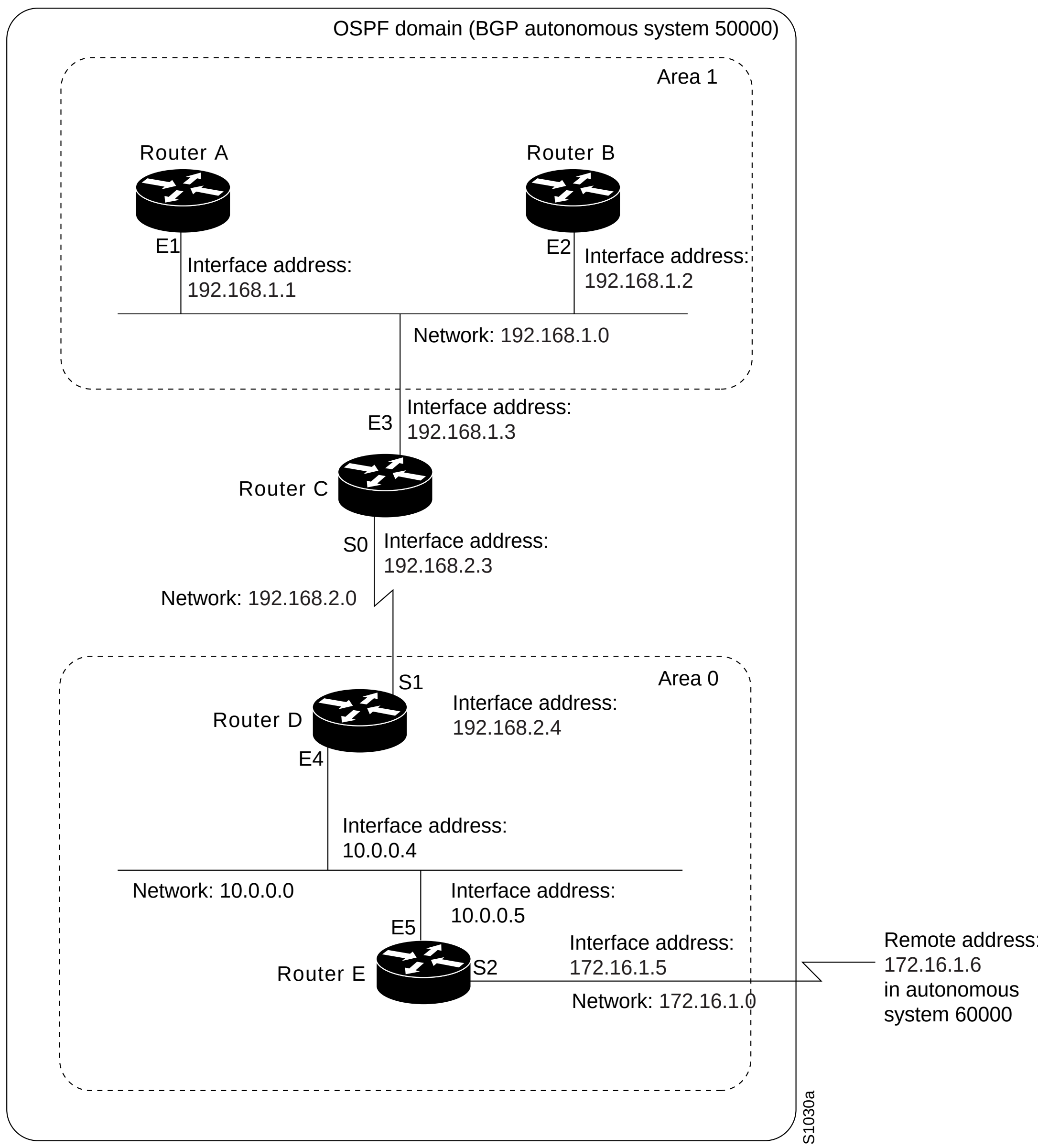
The second **network** command is evaluated next. For Area 2, the same process is then applied to all interfaces (except interface Ethernet 0). Assume that a match is determined for interface Ethernet 1. OSPF is then enabled for that interface and Ethernet 1 is attached to Area 2.

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.

Internal Router, ABR, and ASBRs Configuration Example

The following example outlines a configuration for several routers within a single OSPF autonomous system. Figure 31 provides a general network map that illustrates this example configuration.

Figure 31 Sample OSPF Autonomous System Network Map



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

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

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

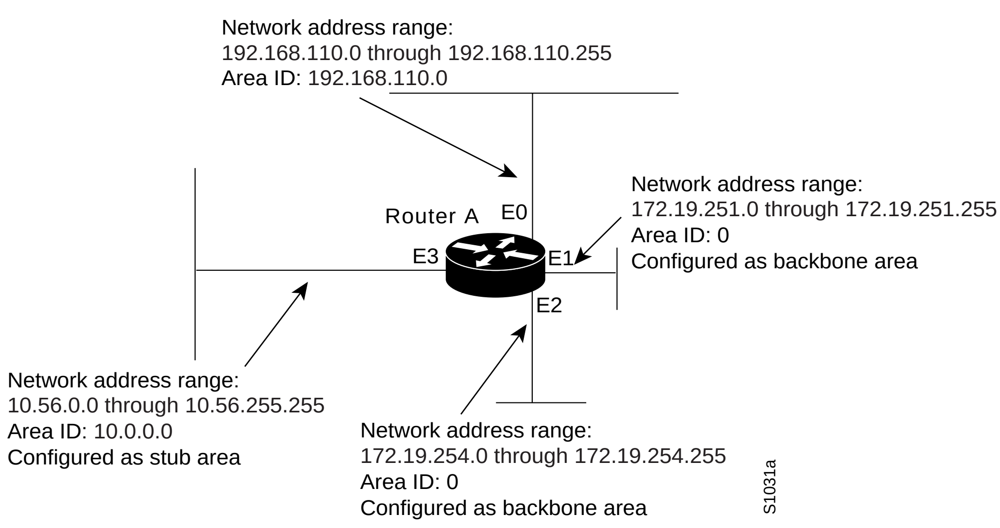
Complex OSPF Configuration Example

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

Figure 32 Interface and Area Specifications for OSPF Example Configuration



The basic configuration tasks in this example are as follows:

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

Configuration tasks associated with redistribution are as follows:

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

The following is an example OSPF configuration:

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

OSPF is on network 131.119.0.0:

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

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

IGRP autonomous system 200 is on 131.119.0.0:

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

Default Metric Values Redistribution Example

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

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

Route Map Examples

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

The following example redistributes all OSPF routes into IGRP:

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

The following example redistributes 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
```

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 is in the routing table. Extended access-lists cannot be used in a route map for *conditional default origination*.

```
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
```

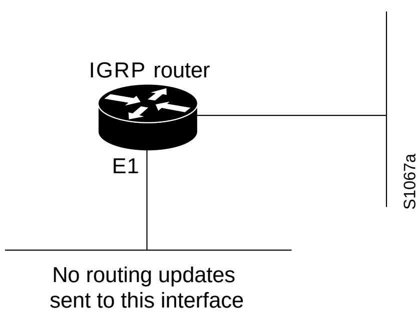
See more route map examples in the sections "BGP Route Map Examples" and "BGP Community with Route Maps Examples" in the "Configuring BGP" chapter.

Passive Interface Examples

The following example sends IGRP updates to all interfaces on network 131.108.0.0 except interface Ethernet 1. Figure 33 shows this configuration.

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

Figure 33 Filtering IGRP Updates



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

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

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

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

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

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

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

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
```

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
!
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 called *flintstone*:

```
key chain flintstone
 key 1
 key-string fred
 key 2
 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
```