IP Routing: Protocol-Independent Configuration Guide, Cisco IOS Release 12.2SR

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Configuring IP Routing Protocol-Independent Features

The Configuring IP Routing Protocol-Independent Features module describes how to configure IP routing protocol-independent features. For a complete description of the IP routing protocol-independent commands in this chapter, see the *Cisco IOS IP Routing: Protocol-Independent Command Reference*. To locate documentation of other commands in this chapter, use the command reference master index or search online.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Configuring IP Routing Protocol-Independent Features

To configure optional protocol-independent features, perform any of the tasks described in the following sections.

Variable-Length Subnet Masks

Enhanced Interior Gateway Routing Protocol (EIGRP), Intermediate System-to-Intermediate System (IS-IS), 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 addressing plan in place and gradually migrate some networks to VLSMs to recover address space. See the Example Variable-Length Subnet Mask, page 25 for an example of using VLSMs.

Static Routes

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

To configure a static route, use the **ip route** *prefix mask*{*ip-address*| *interface-type interface-number*[*ip-address*]}[*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*] command in global configuration mode.

See the Controlling the Advertising of Routes in Routing Updates, page 16 for an example of configuring static routes.

Static routes remains in the router configuration 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 the table below. 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 1 Dynamic Routing Protocol Default Administrative Distances

Route Source	Default Distance
Connected interface	0

Route Source	Default Distance
Static route	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
ODR	160
External EIGRP	170
Internal BGP	200
Unknown	255

Static routes that point to an interface will be advertised via RIP, EIGRP, 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.

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.

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 EIGRP, the network itself is advertised and flagged as an external 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 **ip default-network** *network-number*command in global configuration mode.

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 EIGRP, there might be several networks that can be candidates for the system default. 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.

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 (the best path) to a destination. However, BGP can be configured to use equal and unequal cost multipath load sharing. See the "BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN" feature of the *Cisco IOS IP Routing: BGP Configuration Guide* for more information.

The number of parallel routes that you can configure to be installed in the routing table is dependent on the installed version of Cisco IOS software. To change the maximum number of parallel paths allowed, use the **maximum-paths** *number-paths* command in router configuration mode.

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 **traffic-share min across-interfaces** command in router configuration mode.

Routing Information Redistribution

In addition to running multiple routing protocols simultaneously, Cisco IOS software can be configured to redistribute information from one routing protocol to another. For example, you can configure a router to readvertise EIGRP-derived routes using RIP, or to readvertise static routes using the EIGRP protocol. Redistribution from one routing protocol to another can be configured in all of the IP-based routing protocols.

You also can conditionally control the redistribution of routes between routing domains by configuring route maps between the two domains. A route map is a route/ packet filter that is configured with permit and deny statements, match and set clauses, and sequence numbers.

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 **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] command in global configuration mode.

One or more **match** commands and one or more **set** commands are configured in route map configuration mode. If there are no **match** commands, then everything matches. If there are no **set** commands, then no set action is performed.

To define conditions for redistributing routes from one routing protocol into another, see the Configuring Redistribution Routing Information, page 11

See the "Connecting to a Service Provider Using External BGP" module of the *Cisco IOS IP Routing: BGP Configuration Guide* for examples of BGP route map configuration tasks and configuration examples. See the "Configuring BGP Cost Community" feature of the *Cisco IOS IP Routing: BGP Configuration Guide* for examples of BGP communities and route maps.

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 EIGRP metric is a combination of five metric values. In such situations, a dynamic metric is assigned to the redistributed route. Redistribution in these cases should be applied consistently and carefully in conjunction with inbound filtering to avoid routing loops.

Removing options that you have configured for the **redistribute** command requires careful use of the **no** form of the **redistribute** command to ensure that you obtain the result that you are expecting.

- Supported Metric Translations, page 5

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.
- EIGRP can automatically redistribute static routes from other EIGRP-routed autonomous systems as long as the static route and any associated interfaces are covered by an EIGRP network statement.

EIGRP assigns static routes a metric that identifies them as directly connected. EIGRP does not change the metrics of routes derived from EIGRP updates from other autonomous systems.

Note Any protocol can redistribute routes from other routing protocols as long as a default metric is configured.

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 operators 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 must have adjacencies set, see the Configuring Default Passive Interfaces, page 15.

Sources of Routing Information Filtering

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, see the Filtering Sources of Routing Information, page 16.

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. Sources of Routing Information Filtering, page 6 shows the default administrative distance for various routing information sources.

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

For an example of filtering on sources of routing information, see the section Example Administrative Distances, page 26.

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

Policy-Based Routing

Policy-based 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-based routing if you want certain packets to be routed some way other than the obvious shortest path. Possible applications for policy-based 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-based routing, you must identify which route map to use for policy-based 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-based 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-based routing except when its destination IP address is the same as the IP address of the router's interface. To disable fast switching of all packets arriving on this interface use the **ip policy route-map** *map-tag*command in interface configuration mode.

To define the route map to be used for policy-based routing, use the **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] command in global configuration mode.

To define the criteria by which packets are examined to learn if they will be policy-based routed, use either **match length** *minimum-length maximum-length*commandor **match ip address**{*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*] command or both in route map configuration mode. No match clause in the route map indicates all packets.

To set the precedence and specify where the packets that pass the match criteria are output, see Configuring precedence for policy-based routed packets, page 16.

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 that it had.

The precedence bits in the IP header can be set in the router when policy-based 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. The table below lists the possible numbers and their corresponding name, from least important to most important.

Table 2 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.

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

For information about setting the precedence and specifying where the packets that pass the match the criteria for policy-based routing are output, see the Configuring precedence for policy-based routed packets, page 16.

For information about configure QoS Policy Propagation via BGP, see the Configuring QoS Policy Propagation via BGP, page 17.

See the Example Policy-Based Routing, page 38 for an example of policy routing.

- Fast-Switched Policy Routing, page 8
- Local Policy Routing, page 9
- NetFlow Policy Routing, page 9

Fast-Switched Policy Routing

IP policy routing can also 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 **ip route-cache policy**command in interface configuration mode.

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. To identify the route map to use for local policy routing, use the **ip local policy route-map** *map-tag*command in global configuration mode.

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

NetFlow Policy Routing

NetFlow policy routing (NPR) integrates policy routing, which enables traffic engineering and traffic classification, with NetFlow services, which provide billing, capacity planning, and monitoring information on real-time traffic flows. IP policy routing now works with Cisco Express Forwarding (CEF), distributed CEF (dCEF), and NetFlow.

As quality of service (QoS) and traffic engineering become more popular, so does interest in the ability of policy routing to selectively set IP Precedence and type of service (ToS) bits (based on access lists and packet size), thereby routing packets based on predefined policy. It is important that policy routing work well in large, dynamic routing environments. Hence, distributed support allows customers to leverage their investment in distributed architecture.

NetFlow policy routing leverages the following technologies:

- CEF, which looks at a Forwarding Information Base (FIB) instead of a routing table when switching packets, to address maintenance problems of a demand caching scheme.
- dCEF, which addresses the scalability and maintenance problems of a demand caching scheme.
- NetFlow, which provides accounting, capacity planning, and traffic monitoring capabilities.

Following are NPR benefits:

- NPR takes advantage of the new switching services. CEF, dCEF, and NetFlow can now use policy routing.
- Now that policy routing is integrated into CEF, policy routing can be deployed on a wide scale and on high-speed interfaces.

Following are NPR restrictions:

- NPR is only available on Cisco IOS platforms that support CEF.
- Distributed FIB-based policy routing is only available on platforms that support dCEF.
- The **set ip next-hop verify-availability** command is not supported in dCEF because dCEF does not support the Cisco Discovery Protocol (CDP) database.

In order for NetFlow policy routing to work, the following features must already be configured:

- CEF, dCEF, or NetFlow
- Policy routing

To configure CEF, or dCEF, refer to the "Cisco Express Forwarding Overview" chapter of the *Cisco IOS IP Switching Configuration Guide*. To configure NetFlow, refer to the "Cisco IOS NetFlow Overview" chapter of the *Cisco IOS NetFlow Configuration Guide*.

NPR is the default policy routing mode. No additional configuration tasks are required to enable policy routing in conjunction with CEF, dCEF, or NetFlow. As soon as one of these features is turned on, packets are automatically subject to policy routing in the appropriate switching path.

There is one new, optional configuration command (**set ip next-hop verify-availability**). This command 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.
- The command will not work with dCEF configurations, due to the dependency of the CDP neighbor database.

It is assumed that policy routing itself is already configured.

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 can continue indefinitely.

To prevent this situation from occurring, you can configure the router to first verify that the next hop, using a 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 router to verify that the next hop is a CDP neighbor before the router tries to policy-route to it, use the **set ip next-hop verify-availability** command in route map configuration mode.

If the command shown is set and the next hop is not a CDP neighbor, the router looks to the subsequent next hop, if there is one. If there is none, the packets are simply not policy-routed.

If the command shown is not set, the packets are either policy-routed or remain forever unrouted.

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.

Typically, you would use existing policy routing and NetFlow **show** commands to monitor these features. For more information on these **show** commands, refer to the *Cisco IOS IP Routing: Protocol Independent Command Reference* for policy routing commands and the appropriate chapter of the *Cisco IOS IP NetFlow Command Reference* for NetFlow commands.

To display the route-map Inter Processor Communication (IPC) message statistics in the Route Processor (RP) or Versatile Interface Processor (VIP), use the **show route-map ipc** command in EXEC mode.

Authentication Keys Management

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, 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, see "Managing Authentication Keys".

How to Configure IP Routing Protocol-Independent Features

Configuring Redistribution Routing Information

Defining conditions for redistributing routes

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 or Action	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** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}	Matches a BGP community.
`Router(config-route-map)#` **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...*| *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}	Matches any routes that have a destination network number address that is permitted by a standard access list, an extended access list, or a prefix list, or to perform policy routing on packets
`Router(config-route-map)#` **match metric** *metric-value*	Matches routes with 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.

Command or Action	Purpose
`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.

To define conditions for redistributing routes from one routing protocol into another, use at least one of the following **set** commands in route map configuration mode as needed.

Command or Action	Purpose
`Router(config-route-map)#` **set community** {*community-number* [**additive**] [**well-known**]| **none**}	Sets the BGP communities attribute.
`Router(config-route-map)#` **set dampening** *halflife reuse suppress max-suppress-time*	Sets BGP route dampening parameters.
`Router(config-route-map)#` **set local-preference** *number-value*	Assigns a BGP local-preference value to a 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 route 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 computation 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.

Command or Action	Purpose
`Router(config-route-map)#` **set metric** *metric-value*	Sets the metric value for redistributed routes (for any protocol except EIGRP.
`Router(config-route-map)#` **set metric** *bandwidth delay reliability load mtu*	Sets the metric value to give the redistributed routes (for EIGRP only).
`Router(config-route-map)#` **set metric-type** {**internal** | **external** | **type-1** | **type-2**}	Sets the metric type assigned to redistributed routes.
`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 a tag value to apply to redistributed routes.

Redistributing routes from one routing domain to another

To distribute routes from one routing domain into another routing domain and to control route redistribution, perform the following task.

SUMMARY STEPS

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**]
2. Router(config-router)# **default-metric** *number*
3. Router(config-router)# **default-metric** *bandwidth delay reliability loading mtu*
4. Router(config-router)# **no default-information** {**in** | **out**}

DETAILED STEPS

	Command or Action	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 into 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 EIGRP routing protocol to use the same metric value for all non-EIGRP redistributed routes.
Step 4	Router(config-router)# **no default-information** {**in** | **out**}	Disables the redistribution of default information between EIGRP processes, which is enabled by default.

Removing options for redistributing routes

Removing options that you have configured for the **redistribute** command requires careful use of the **no** form of the **redistribute** command to ensure that you obtain the result that you are expecting.

SUMMARY STEPS

1. Router(config-router)# **no redistribute connected metric** *1000* **subnets**
2. Router(config-router)# **no redistribute connected metric** *1000*
3. Router(config-router)# **no redistribute connected subnets**
4. Router(config-router)# **no redistribute connected**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config-router)# **no redistribute connected metric** *1000* **subnets**	Removes the configured metric and the metric-value of 1000 and the configured subnets and retains the **redistribute connected** command in the configuration.
Step 2	Router(config-router)# **no redistribute connected metric** *1000*	Removes the configured metric and the metric-value of 1000 and retains the **redistribute connected subnets** command in the configuration.
Step 3	Router(config-router)# **no redistribute connected subnets**	Removes the configured subnets and retains the **redistribute connected metric** *metric-value* command in the configuration.
Step 4	Router(config-router)# **no redistribute connected**	Removes the **redistribute connected** command and any of the options that were configured for the command.

Configuring Routing Information Filtering

To filter routing protocol information, perform the tasks in the following sections. The tasks in the first section are required; the remaining sections are 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 that 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 **passive-interface** *interface-type interface-number* command in router configuration mode. See the Example Passive Interface, page 37 for examples of configuring passive interfaces.

Configuring Default Passive Interfaces

To set all interfaces as passive by default and then activate only those interfaces that must have adjacencies set, perform the following task.

SUMMARY STEPS

1. Router> **enable**
2. Router# **configure terminal**
3. Router(config)# **router** *protocol*
4. Router(config-router)# **passive-interface default**
5. Router(config-router)# **no passive-interface** *interface-type*
6. Router(config-router)# **network** *network-address*[*options*]
7. Router(config-router)# **end**
8. Router# **show ip ospf interface**
9. Router# **show ip interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **router** *protocol*	Configures the routing protocol on the network.
Step 4	Router(config-router)# **passive-interface default**	Sets all interfaces as passive by default.
Step 5	Router(config-router)# **no passive-interface** *interface-type*	Activates only those interfaces that must have adjacencies set.
Step 6	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.
Step 7	Router(config-router)# **end**	Exits router configuration mode and returns to privileged EXEC mode.
Step 8	Router# **show ip ospf interface**	Displays interface information related to Open Shortest Path First (OSPF).
Step 9	Router# **show ip interface**	Displays the usability status of interfaces configured for IP.

See the section "Example Default Passive Interface" 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** command, or you could verify the interfaces that you enabled as active using a command such as the **show ip interface** 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 **distribute-list** {*access-list-number* | *access-list-name*} **out**[*interface-name* | *routing-process* | *as-number*] command in router configuration mode.

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 **distribute-list** {*access-list-number* | *access-list-name*} **in**[*interface-type interface-number*] command in router configuration mode.

Filtering Sources of Routing Information

To filter sources of routing information, use the **distance** *ip-address wildcard- mask*[*ip-standard-acl* | *ip-extended-acl* | *access-list-name*] command in router configuration mode.

Configuring precedence for policy-based routed packets

To configure the precedence and specify where the packets that pass the match criteria are output, perform the following task.

Note The **set ip next-hop** and **set ip default next-hop** commands are similar but have a different order of operation. 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.

SUMMARY STEPS

1. Router(config-route-map)# **set ip precedence**{*number* | *name*}
2. Router(config-route-map)# **set ip next-hop** *ip-address* [*ip-address*]
3. Router(config-route-map)# **set interface** *interface-type interface-number*[*... interface-type interface-number*]
4. Router(config-route-map)# **set ip default next-hop** *ip-address* [*ip-address*]
5. Router(config-route-map)# **set default interface** *interface-type interface-number*[*... interface-type interface-number*]

DETAILED STEPS

	Command or Action	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. **Note** The next hop 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 must 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 the destination.

Configuring QoS Policy Propagation via BGP

The QoS Policy Propagation via BGP feature allows you to classify packets by IP precedence based on BGP community lists, BGP autonomous system paths, and access lists. After a packet has been classified, you can use other QoS features such as committed access rate (CAR) and Weighted Random Early Detection (WRED) to specify and enforce policies to fit your business model.

To configure Policy Propagation via BGP, perform the following basic tasks:

- Configure BGP and Cisco Express Forwarding (CEF) or distributed CEF (dCEF). To configure BGP, refer to the *Cisco IOS IP Routing: BGP Configuration Guide* . To configure CEF and dCEF, refer to the *Cisco IOS IP Switching Configuration Guide* .
- Define the policy.
- Apply the policy through BGP.
- Configure the BGP community list, BGP autonomous system path, or access list and enable the policy on an interface. For information about these tasks, see the tasks below.
- Enable CAR or WRED to use the policy. To enable CAR, see the chapter "Configuring Committed Access Rate" in the *Cisco IOS Quality of Service Solutions Configuration Guide*. To configure WRED, see the chapter "Configuring Weighted Random Early Detection" in the *Cisco IOS Quality of Service Solutions Configuration Guide*.

This section describes how to configure QoS Policy Propagation based on BGP community list, BGP autonomous system path, or access list. It assumes you have already configured BGP and CEF or dCEF.

To configure QoS Policy Propagation via BGP, perform the tasks described in the following sections. The tasks in the first three sections are required; the task in the remaining section is optional.

Note For the QoS Policy Propagation via BGP feature to work, you must enable BGP and CEF/dCEF on the router. Subinterfaces on an ATM interface that have the **bgp-policy**command enabled must use CEF mode because dCEF is not supported. dCEF uses the Versatile Interface Processor (VIP) rather than the Route Switch Processor (RSP) to perform forwarding functions.

For configuration examples, see "Examples QoS Policy Propagation via BGP Configuration."

- Configuring QoS Policy Propagation Based on Community Lists, page 18
- Configuring QoS Policy Propagation Based on the Autonomous System Path Attribute, page 19
- Configuring QoS Policy Propagation Based on an Access List, page 21
- Monitoring QoS Policy Propagation via BGP, page 22

Configuring QoS Policy Propagation Based on Community Lists

This section describes how to configure Policy Propagation via BGP using community lists. The tasks listed in this section are required unless noted as optional. This section assumes that you have already configured CEF/dCEF and BGP on your router.

Perform the following task to configure the router to propagate the IP precedence based on the community lists.

SUMMARY STEPS

1. Router> **enable**
2. Router# **configure terminal**
3. Router(config)# **route-map** *route-map-name* [**permit** | **deny**[*sequence-number*]]
4. Router(config-route-map)# **match community-list** *community-list-number* [**exact**]
5. Router(config-route-map)# **set ip precedence**[*number* | *name*]
6. Router(config-route-map)# **exit**
7. Router(config)# **router bgp** *autonomous-system*
8. Router(config-router)# **table-map** *route-map-name*
9. Router(config-router)# **exit**
10. Router(config)# **ip community-list** *community-list-number*{**permit** | **deny**} *community-number*
11. Router(config)# **interface** *interface-type interface-number*
12. Router(config-if)# **bgp-policy** {**source** | **destination**} **ip-prec-map**
13. Router(config-if)# **ip bgp-community new-format**
14. Router(config-if)# **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Router# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 3	Router(config)# **route-map** *route-map-name* [**permit** | **deny**[*sequence-number*]]	Defines a route map to control redistribution and enters route map configuration mode.
Step 4	Router(config-route-map)# **match community-list** *community-list-number* [**exact**]	Matches a BGP community list.
Step 5	Router(config-route-map)# **set ip precedence**[*number* | *name*]	Sets the IP Precedence field when the community list matches. You can specify either a precedence number or name.
Step 6	Router(config-route-map)# **exit**	Exits route map configuration mode and returns the router to global configuration mode.
Step 7	Router(config)# **router bgp** *autonomous-system*	Enters router configuration mode.
Step 8	Router(config-router)# **table-map** *route-map-name*	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 9	Router(config-router)# **exit**	Exits router configuration mode and returns the router to global configuration mode.
Step 10	Router(config)# **ip community-list** *community-list-number*{**permit** | **deny**} *community-number*	Creates a community list for BGP and controls access to it.
Step 11	Router(config)# **interface** *interface-type interface-number*	Specifies the interfaces (or subinterface) and enters interface configuration mode.
Step 12	Router(config-if)# **bgp-policy** {**source** | **destination**} **ip-prec-map**	Classifies packets using IP Precedence.
Step 13	Router(config-if)# **ip bgp-community new-format**	(Optional) Configures a new community format so that the community number is displayed in the short form.
Step 14	Router(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring QoS Policy Propagation Based on the Autonomous System Path Attribute

This section describes how to configure QoS Policy Propagation via BGP based on the autonomous system path. This section assumes that you have already configured CEF/dCEF and BGP on your router.

Perform the following task to configure the router to propagate the IP precedence based on the autonomous system path attribute.

SUMMARY STEPS

1. Router> **enable**
2. Router# **configure terminal**
3. Router(config)# **route-map** *route-map-name* [**permit** | **deny**[*sequence-number*]]
4. Router(config-route-map)# **match as-path** *path-list-number*
5. Router(config-route-map)# **set ip precedence**[*number* | *name*]
6. Router(config-route-map)# **exit**
7. Router(config)# **router bgp** *autonomous-system*
8. Router(config-router)# **table-map** *route-map-name*
9. Router(config-router)# **exit**
10. Router(config)# **ip as-path access-list** *access-list-number*{**permit** | **deny**} *as-regular-expression*
11. Router(config)# **interface** *interface-type interface-number*
12. Router(config-if)# **bgp-policy** {**source** | **destination**} **ip-prec-map**
13. Router(config-if)# **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **route-map** *route-map-name* [**permit** | **deny**[*sequence-number*]]	Defines a route map to control redistribution and enters route-map configuration mode.
Step 4	Router(config-route-map)# **match as-path** *path-list-number*	Matches a BGP autonomous system path access list.
Step 5	Router(config-route-map)# **set ip precedence**[*number* | *name*]	Sets the IP Precedence field when the autonomous system path matches. Specifies either a precedence number or name.
Step 6	Router(config-route-map)# **exit**	Exits route map configuration mode and returns the router to global configuration mode.
Step 7	Router(config)# **router bgp** *autonomous-system*	Enters router configuration mode.
Step 8	Router(config-router)# **table-map** *route-map-name*	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 9	Router(config-router)# **exit**	Exits router configuration mode and returns the router to global configuration mode.
Step 10	Router(config)# **ip as-path access-list** *access-list-number*{**permit** | **deny**} *as-regular-expression*	Defines an autonomous system path access list.
Step 11	Router(config)# **interface** *interface-type interface-number*	Specifies the interfaces (or subinterface) and enters interface configuration mode.

	Command or Action	Purpose
Step 12	Router(config-if)# **bgp-policy** {**source** | **destination**} **ip-prec-map**	Classifies packets using IP Precedence.
Step 13	Router(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring QoS Policy Propagation Based on an Access List

This section describes how to configure QoS Policy Propagation via BGP based on an access list. This section assumes you have already configured CEF/dCEF and BGP on your router.

To configure the router to propagate the IP Precedence based on an access list, perform the following task.s

SUMMARY STEPS

1. Router> **enable**
2. Router# **configure terminal**
3. Router(config)# **route-map** *route-map-name* [**permit** | **deny**[*sequence-number*]]
4. Router(config-route-map)# **match ip address** *access-list-number*
5. Router(config-route-map)# **set ip precedence**[*number* | *name*]
6. Router(config-route-map)# **exit**
7. Router(config)# **router bgp** *autonomous-system*
8. Router(config-router)# **table-map** *route-map-name*
9. Router(config-router)# **exit**
10. Router(config)# **access-list** *access-list-number* {**permit** | **deny**} *source*
11. Router(config)# **interface** *interface-type interface-number*
12. Router(config-if)# **bgp-policy** {**source** | **destination**} **ip-prec-map**
13. Router(config-if)# **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **route-map** *route-map-name* [**permit** | **deny**[*sequence-number*]]	Defines a route map to control redistribution and enters route map configuration mode.
Step 4	Router(config-route-map)# **match ip address** *access-list-number*	Matches an access list.
Step 5	Router(config-route-map)# **set ip precedence**[*number* | *name*]	Sets the IP Precedence field when the autonomous system path matches.

	Command or Action	Purpose
Step 6	Router(config-route-map)# **exit**	Exits route map configuration mode and returns the router to global configuration mode.
Step 7	Router(config)# **router bgp** *autonomous-system*	Enters router configuration mode.
Step 8	Router(config-router)# **table-map** *route-map-name*	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 9	Router(config-router)# **exit**	Exits router configuration mode and returns the router to global configuration mode.
Step 10	Router(config)# **access-list** *access-list-number* {**permit** | **deny**} *source*	Defines an access list.
Step 11	Router(config)# **interface** *interface-type interface-number*	Specifies the interfaces (or subinterface) and enters interface configuration mode.
Step 12	Router(config-if)# **bgp-policy** {**source** | **destination**} **ip-prec-map**	Classifies packets using IP Precedence.
Step 13	Router(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

Monitoring QoS Policy Propagation via BGP

To monitor the QoS Policy Propagation via BGP configuration, use the following commands in EXEC mode, as needed. The commands listed in this section are optional.

Command or Action	Purpose
`Router#` **show ip bgp**	Displays entries in the BGP routing table, to verify that the correct community is set on the prefixes.
`Router#` **show ip bgp community-list** *community-list-number*	Displays routes permitted by the BGP community list, to verify that the correct prefixes are selected.
`Router#` **show ip cef** *network*	Displays entries in the Forwarding Information Base (FIB) table based on the IP address, to verify that CEF has the correct precedence value for the prefix.
`Router#` **show ip interface**	Displays information about the interface.
`Router#` **show ip route** *prefix*	Displays the current status of the routing table, to verify that the correct precedence values are set on the prefixes.

Managing Authentication Keys

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**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 that 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 *Network Management Configuration Guide*.

To manage authentication keys, perform the following task.

SUMMARY STEPS

1. Router> **enable**
2. Router# **configure terminal**
3. Router(config)# **key chain** *name-of-chain*
4. Router(config-keychain)# **key** *number*
5. Router(config-keychain-key)# **key-string** *text*
6. Router(config-keychain-key)# **accept-lifetime** *start-time*{**infinite** | *end-time*| **duration** *seconds*}
7. Router(config-keychain-key)# **send-lifetime** *start-time*{**infinite** | *end-time* | **duration** *seconds*}
8. Router(config-keychain-key)# **end**
9. Router# **show key chain**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **key chain** *name-of-chain*	Identifies a key chain.
Step 4	Router(config-keychain)# **key** *number*	Identifies the key number in key chain configuration mode.
Step 5	Router(config-keychain-key)# **key-string** *text*	Identifies the key string in key chain configuration mode.
Step 6	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 7	Router(config-keychain-key)# **send-lifetime** *start-time*{**infinite** | *end-time* | **duration** *seconds*}	Specifies the time period during which the key can be sent.
Step 8	Router(config-keychain-key)# **end**	Exits key chain key configuration mode and returns to privileged EXEC mode.
Step 9	Router# **show key chain**	Displays authentication key information.

For examples of key management, see the Examples Key Management, page 41.

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 **clear ip route** {*network* [*mask*] | *****} command in EXEC mode.

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 that packets leaving your device are taking through the network.

Command or Action	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** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | **list** {*access-list-number* | *access-list-name*} | **static download**]	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.
`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.

Configuration Examples for Configuring IP Routing Protocol-Independent Features

Example Variable-Length Subnet Mask

The following example uses two different subnet masks for the class B network address of 172.16.0.0. A subnet mask of /24 is used for LAN interfaces. The /24 mask allows 256 subnets with 254 host IP addresses on each subnet. The final subnet of the range of possible subnets using a /24 mask (172.16.255.0) is reserved for use on point-to-point interfaces and assigned a longer mask of /30. The use of a /30 mask on 172.16.255.0 creates 64 subnets (172.16.255.0 - 172.16.255.252) with 2 host addresses on each subnet.

Danger To ensure unambiguous routing, you must not assign 172.16.255.0/24 to a LAN interface in your network.

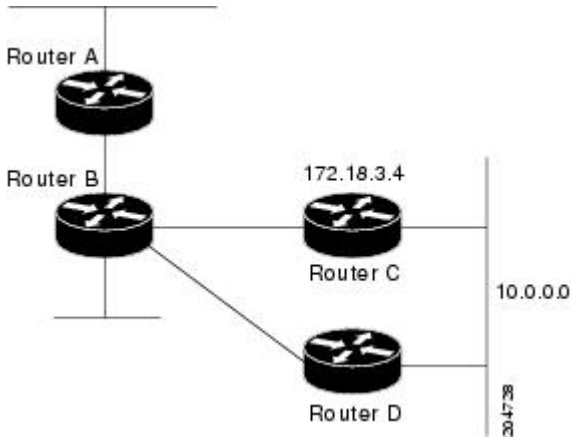
```
Router(config)# interface Ethernet 0/0

Router(config-if)# ip address 172.16.1.1 255.255.255.0

Router(config-if)# ! 8 bits of host address space reserved for Ethernet interfaces
Router(config-if)# exit
Router(config)# interface Serial 0/0

Router(config-if)# ip address 172.16.255.5 255.255.255.252

Router(config-if)# ! 2 bits of address space reserved for point-to-point serial
interfaces
Router(config-if)# exit

Router(config)# router rip
Router(config-router)# network 172.16.0.0
Router(config-router)# ! Specifies the network directly connected to the router
```

Example Overriding Static Routes with Dynamic Protocols

In the following example, packets for network 10.0.0.0 from Router B (where the static route is installed) will be routed through 172.18.3.4 if a route with an administrative distance less than 110 is not available. The figure below 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.

```
Router(config)# ip route 10.0.0.0 255.0.0.0 172.18.3.4 110
```

Figure 1 Overriding Static Routes



Example Administrative Distances

In the following example, the **router eigrp** global configuration command configures EIGRP routing in autonomous system 1. The **network** command configuration specifies EIGRP 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 80 for internal EIGRP routes and to 100 for external EIGRP routes. The third **distance** command sets the administrative distance to 120 for the router with the address 172.16.1.3.

```
Router(config)# router eigrp 1
Router(config-router)# network 192.168.7.0
Router(config-router)# network 172.16.0.0
Router(config-router)# distance 255
Router(config-router)# distance eigrp 80 100
Router(config-router)# distance 120 172.16.1.3 0.0.0.0
```

Note The **distance eigrp** command must be used to set the administrative distance for EIGRP-derived routes.

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:

```
Router(config-router)# distance 100 192.168.7.18 0.0.0.0
Router(config-router)# 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:

```
Router(config-router)# distance 200 192.168.7.0 0.0.0.255
Router(config-router)# distance 100 192.168.7.18 0.0.0.0
```

Note Assigning administrative distances can be used to solve unique problems. However, administrative distances should be applied carefully and consistently to avoid the creation of routing loops or other network failures.

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(config)# router isis

Router(config-router)# distance 90 ip
```

Example Static Routing Redistribution

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 EIGRP process. Any redistributed static routes should be sourced by a single router to minimize the likelihood of creating a routing loop.

```
Router(config)# ip route 192.168.2.0 255.255.255.0 192.168.7.65
Router(config)# ip route 192.168.5.0 255.255.255.0 192.168.7.65
Router(config)# ip route 10.10.10.0 255.255.255.0 10.20.1.2
Router(config)# !
Router(config)# access-list 3 permit 192.168.2.0 0.0.255.255
Router(config)# access-list 3 permit 192.168.5.0 0.0.255.255
Router(config)# access-list 3 permit 10.10.10.0 0.0.0.255
Router(config)# !
Router(config)# router eigrp 1
Router(config-router)# network 192.168.0.0
Router(config-router)# network 10.10.10.0
Router(config-router)# redistribute static metric 10000 100 255 1 1500
Router(config-router)# distribute-list 3 out static
```

Example EIGRP Redistribution

Each 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.

In the following configuration, network 10.0.0.0 is configured under EIGRP autonomous system 1 and network 192.168.7.0 is configured under EIGRP autonomous system 101:

```
Router(config)# router eigrp 1
Router(config-router)# network 10.0.0.0
Router(config-router)# exit
Router(config)# router eigrp 101
Router(config-router)# network 192.168.7.0
```

In the following example, routes from the 192.168.7.0 network are redistributed into autonomous system 1 (without passing any other routing information from autonomous system 101):

```
Router(config)# access-list 3 permit 192.168.7.0
```

```
Router(config)# !
Router(config)# route-map 101-to-1 permit 10
Router(config-route-map)# match ip address 3
Router(config-route-map)# set metric 10000 100 1 255 1500
Router(config-route-map)# exit
Router(config)# router eigrp 1
Router(config-router)# redistribute eigrp 101 route-map 101-to-1
Router(config-router)#!
```

The following example is an alternative way to redistribute routes from the 192.168.7.0 network into autonomous system 1. Unlike the previous configuration, this method does not allow you to set the metric for redistributed routes.

```
Router(config)# access-list 3 permit 192.168.7.0
Router(config)# !
Router(config)# router eigrp 1
Router(config-router)# redistribute eigrp 101
Router(config-router)# distribute-list 3 out eigrp 101
Router(config-router)# !
```

Example 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, 172.16.0.0, which uses 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.

In the following example, EIGRP-to-RIP redistribution is configured:

```
Router(config)# access-list 10 permit 172.16.0.0
Router(config)# !
Router(config)# router eigrp 1
Router(config-router)# network 172.16.0.0
Router(config-router)# redistribute rip metric 10000 100 255 1 1500
Router(config-router)# distribute-list 10 out rip
Router(config-router)# exit
Router(config)# router rip
Router(config-router)# redistribute eigrp 1
Router(config-router)# !
```

In this example, an EIGRP routing process is started. 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.

Example Complex Redistribution

In the following example, *mutual* redistribution is configured between EIGRP and BGP.

Routes from BGP autonomous system 50000 are injected into EIGRP routing process101. A filter is configured to ensure that the correct routes are advertised.

```
Router(config)# ! All networks that should be advertised from R1 are controlled with
ACLs:
Router(config)# access-list 1 permit 172.18.0.0 0.0.255.255
Router(config)# access-list 1 permit 172.16.0.0 0.0.255.255
Router(config)# access-list 1 permit 172.25.0.0 0.0.255.255
Router(config)# ! Configuration for router R1:
Router(config)# router bgp 50000
Router(config-router)# network 172.18.0.0
```

```
Router(config-router)# network 172.16.0.0
Router(config-router)# neighbor 192.168.10.1 remote-as 2
Router(config-router)# neighbor 192.168.10.15 remote-as 1
Router(config-router)# neighbor 192.168.10.24 remote-as 3
Router(config-router)# redistribute eigrp 101
Router(config-router)# distribute-list 1 out eigrp 101
Router(config-router)# exit
Router(config)# router eigrp 101
Router(config-router)# network 172.25.0.0
Router(config-router)# redistribute bgp 50000
Router(config-router)# distribute-list 1 out bgp 50000
Router(config-router)# !
```

Caution BGP should be redistributed into an IGP when there are no other suitable options. Redistribution from BGP into any IGP should be applied with proper filtering using distribute-lists, IP prefix-list, and route map statements to limit the number of prefixes. BGP routing tables can be very large. Redistributing all BGP prefixes into an IGP can have a detrimental effect on IGP network operations.

Examples OSPF Routing and Route Redistribution

OSPF typically requires coordination among many internal routers, Area Border 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.

- Examples Basic OSPF Configuration, page 29
- Example Internal Router ABR and ASBRs Configuration, page 30
- Example Complex OSPF Configuration, page 33

Examples Basic OSPF Configuration

The following example illustrates a simple OSPF configuration that enables OSPF routing process 1, attaches Ethernet interface 0 to area 0.0.0.0, and redistributes RIP into OSPF and OSPF into RIP:

```
Router(config)# interface Ethernet 0
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# ip ospf cost 1
Router(config-if)# exit
Router(config)# interface Ethernet 1
Router(config-if)# ip address 172.17.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 172.18.0.0 0.0.255.255 area 0.0.0.0
Router(config-router)# redistribute rip metric 1 subnets
Router(config-router)# exit
Router(config)# router rip
Router(config-router)# network 172.17.0.0
Router(config-router)# redistribute ospf 1
Router(config-router)# default-metric 1
Router(config-router)# !
```

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(config)# router ospf 1
Router(config-router)# network 172.18.20.0 0.0.0.255 area 10.9.50.0
Router(config-router)# network 172.18.0.0 0.0.255.255 area 2
Router(config-router)# network 172.19.10.0 0.0.0.255 area 3
Router(config-router)# network 0.0.0.0 255.255.255.255 area 0
Router(config-router)# exit
Router(config)# ! Ethernet interface 0 is in area 10.9.50.0:
Router(config)# interface Ethernet 0
Router(config-if)# ip address 172.18.20.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 1 is in area 2:
Router(config)# interface Ethernet 1
Router(config-if)# ip address 172.18.1.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 2 is in area 2:
Router(config)# interface Ethernet 2

Router(config-if)# ip address 172.18.2.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 3 is in area 3:
Router(config)# interface Ethernet 3
Router(config-if)# ip address 172.19.10.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 4 is in area 0:
Router(config)# interface Ethernet 4
Router(config-if)# ip address 172.19.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 5 is in area 0:
Router(config)# interface Ethernet 5
Router(config-if)# ip address 10.1.0.1 255.255.0.0
Router(config-if)# !
```

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 *Cisco IOS IP Routing: Protocol-Independent Command Reference* 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.

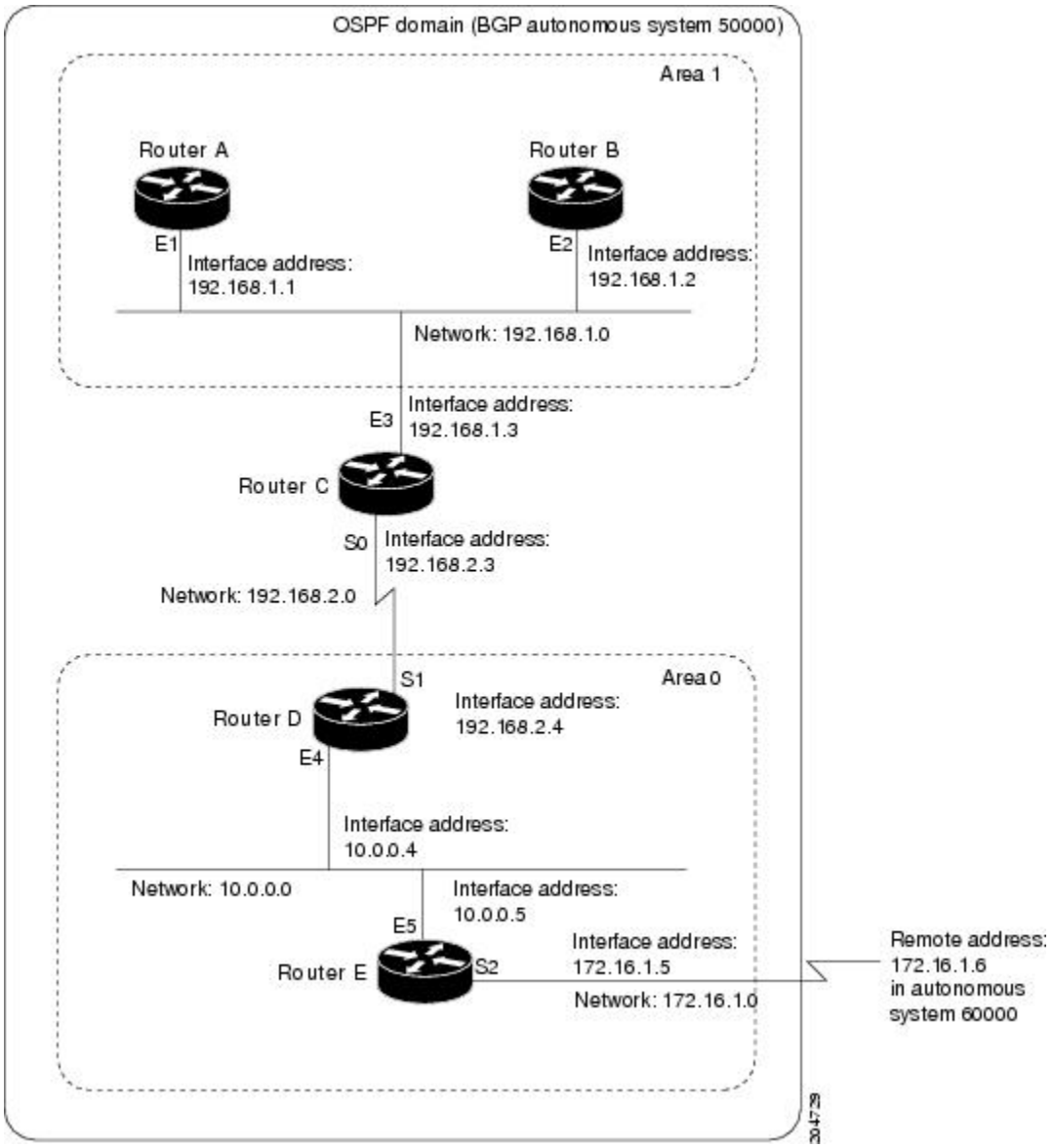
The second **network**command is evaluated next. For Area 2, the same process is then applied to all interfaces (except Ethernet interface 0). Assume that a match is determined for Ethernet interface 1. OSPF is then enabled for that interface, and Ethernet 1 is attached to Area 2.

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.

Example Internal Router ABR and ASBRs Configuration

The figure below provides a general network map that illustrates a sample configuration for several routers within a single OSPF autonomous system.

Figure 2 Example OSPF Autonomous System Network Map



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

- Router A and Router Bare 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 the figure above.

Router A Configuration--Internal Router

```
Router(config)# interface Ethernet 1
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 1
Router(config-router)# exit
```

Router B Configuration--Internal Router

```
Router(config)# interface Ethernet 2
Router(config-if)# ip address 192.168.1.2 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 1
Router(config-router)# exit
```

Router C Configuration--ABR

```
Router(config)# interface Ethernet 3
Router(config-if)# ip address 192.168.1.3 255.255.255.0
Router(config-if)# exit
Router(config)# interface Serial 0
Router(config-if)# ip address 192.168.2.3 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 1
Router(config-router)# network 192.168.2.0 0.0.0.255 area 0
Router(config-router)# exit
```

Router D Configuration--Internal Router

```
Router(config)# interface Ethernet 4
Router(config-if)# ip address 10.0.0.4 255.0.0.0
Router(config-if)# exit
Router(config)# interface Serial 1
Router(config-if)# ip address 192.168.2.4 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.2.0 0.0.0.255 area 0
Router(config-router)# network 10.0.0.0 0.255.255.255 area 0
Router(config-router)# exit
```

Router E Configuration--ASBR

```
Router(config)# interface Ethernet 5
```

```
Router(config-if)# ip address 10.0.0.5 255.0.0.0
Router(config-if)# exit
Router(config)# interface Serial 2
Router(config-if)# ip address 172.16.1.5 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 10.0.0.0 0.255.255.255 area 0
Router(config-router)# redistribute bgp 50000 metric 1 metric-type 1
Router(config-router)# exit
Router(config)# router bgp 50000
Router(config-router)# network 192.168.0.0
Router(config-router)# network 10.0.0.0
Router(config-router)# neighbor 172.16.1.6 remote-as 60000
```

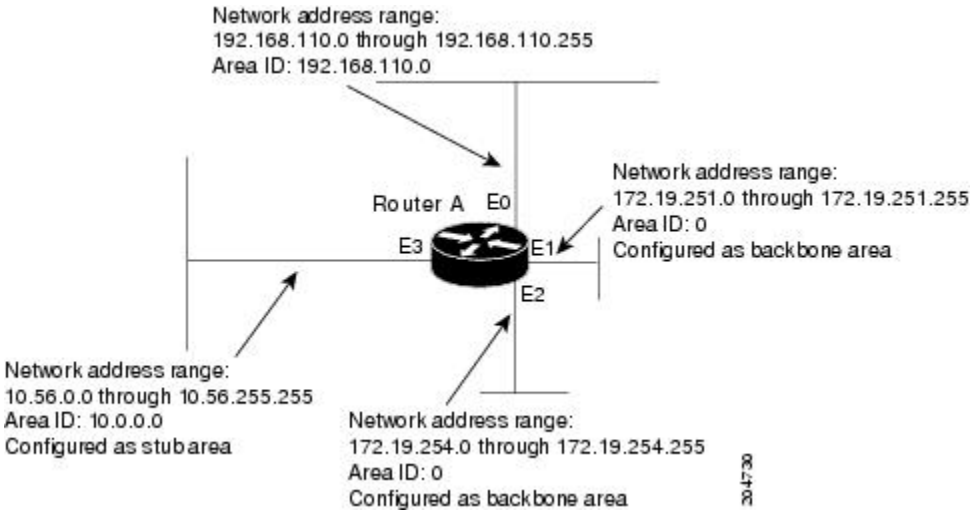
Example Complex OSPF Configuration

The following example configuration accomplishes several tasks in setting up an ABR. These tasks can be split into two general categories:

- Basic OSPF configuration
- Route redistribution

The specific tasks outlined in this configuration are detailed briefly in the following descriptions. The figure below illustrates the network address ranges and area assignments for the interfaces.

Figure 3 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 EIGRP and RIP into OSPF with various options set (including **metric-type**, **metric**, **tag**, and **subnet**).
- Redistribute EIGRP and OSPF into RIP.

The following is an example OSPF configuration:

```
Router(config)# interface Ethernet 0
Router(config-if)# ip address 192.168.110.201 255.255.255.0
Router(config-if)# ip ospf authentication-key abcdefgh
Router(config-if)# ip ospf cost 10
Router(config-if)# exit
Router(config)# interface Ethernet 1
Router(config-if)# ip address 172.19.251.201 255.255.255.0
Router(config-if)# ip ospf authentication-key ijklmnop
Router(config-if)# ip ospf cost 20
Router(config-if)# ip ospf retransmit-interval 10
Router(config-if)# ip ospf transmit-delay 2
Router(config-if)# ip ospf priority 4
Router(config-if)# exit
Router(config)# interface Ethernet 2
Router(config-if)# ip address 172.19.254.201 255.255.255.0
Router(config-if)# ip ospf authentication-key abcdefgh
Router(config-if)# ip ospf cost 10
Router(config-if)# exit
Router(config)# interface Ethernet 3
Router(config-if)# ip address 10.56.0.201 255.255.0.0
Router(config-if)# ip ospf authentication-key ijklmnop
Router(config-if)# ip ospf cost 20
Router(config-if)# ip ospf dead-interval 80
Router(config-if)# exit
```

In the following configuration, OSPF is on network 172.19.0.0:

```
Router(config)# router ospf 1
Router(config-router)# network 10.0.0.0 0.255.255.255 area 10.0.0.0
Router(config-router)# network 192.168.110.0 0.0.0.255 area 192.68.110.0
Router(config-router)# network 172.19.0.0 0.0.255.255 area 0
Router(config-router)# area 0 authentication
Router(config-router)# area 10.0.0.0 stub
Router(config-router)# area 10.0.0.0 authentication
Router(config-router)# area 10.0.0.0 default-cost 20
Router(config-router)# area 192.168.110.0 authentication
Router(config-router)# area 10.0.0.0 range 10.0.0.0 255.0.0.0
Router(config-router)# area 192.168.110.0 range 192.168.110.0 255.255.255.0
Router(config-router)# area 0 range 172.19.251.0 255.255.255.0
Router(config-router)# area 0 range 172.19.254.0 255.255.255.0
Router(config-router)# redistribute eigrp 200 metric-type 2 metric 1 tag 200 subnets
Router(config-router)# redistribute rip metric-type 2 metric 1 tag 200
Router(config-router)# exit
```

In the following configuration, EIGRP autonomous system 1 is on 172.19.0.0:

```
Router(config)# router eigrp 1
Router(config-router)# network 172.19.0.0
Router(config-router)# exit
Router(config)# ! RIP for 192.168.110.0:
Router(config)# router rip
Router(config-router)# network 192.168.110.0
Router(config-router)# redistribute eigrp 1 metric 1
Router(config-router)# redistribute ospf 201 metric 1
Router(config-router)# exit
```

Example Default Metric Values Redistribution

The following example shows a router in autonomous system 1 that is configured to run both RIP and EIGRP. The example advertises EIGRP-derived routes using RIP and assigns the EIGRP-derived routes a RIP metric of 10.

```
Router(config)# router rip
Router(config-router)# default-metric 10
Router(config-router)# redistribute eigrp 1
Router(config-router)# exit
```

Example Route Map

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 EIGRP:

```
Router(config)# router eigrp 1
Router(config-router)# redistribute ospf 101
Router(config-router)# exit
```

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(config)# router ospf 1
Router(config-router)# redistribute rip route-map rip-to-ospf
Router(config-router)# exit
Router(config)# route-map rip-to-ospf permit
Router(config-route-map)# match metric 1
Router(config-route-map)# set metric 5
Router(config-route-map)# set metric-type type 1
Router(config-route-map)# set tag 1
Router(config-route-map)# exit
```

The following example redistributes OSPF learned routes with tag 7 as a RIP metric of 15:

```
Router(config)# router rip
Router(config-router)# redistribute ospf 1 route-map 5
Router(config-router)# exit
Router(config)# route-map 5 permit
Router(config-route-map)# match tag 7
Router(config-route-map)# 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(config)# router bgp 50000
Router(config-router)# redistribute ospf 1 route-map 10
Router(config-router)# exit
Router(config)# route-map 10 permit
Router(config-route-map)# match route-type internal
Router(config-route-map)# match interface serial 0
Router(config-route-map)# 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(config)# router isis
```

```
Router(config-router)# redistribute ospf 1 route-map 2
Router(config-router)# redistribute iso-igrp nsfnet route-map 3

Router(config-router)# exit
Router(config)# route-map 2 permit
Router(config-route-map)# match route-type external
Router(config-route-map)# match tag 5
Router(config-route-map)# set metric 5
Router(config-route-map)# set level level-2
Router(config-route-map)# exit
Router(config)# route-map 3 permit
Router(config-route-map)# match address 2000
Router(config-route-map)# set metric 30
Router(config-route-map)# exit
```

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(config)# router rip
Router(config-router)# redistribute ospf 101 route-map 1
Router(config-router)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match tag 1 2
Router(config-route-map)# set metric 1
Router(config-route-map)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match tag 3
Router(config-route-map)# set metric 5
Router(config-route-map)# exit
Router(config)# route-map 1 deny
Router(config-route-map)# match tag 4
Router(config-route-map)# exit
Router(config)# route map 1 permit
Router(config-route-map)# match tag 5
Router(config-route-map)# set metric 5
Router(config-route-map)# exit
```

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(config)# router isis
Router(config-router)# redistribute rip route-map 1
Router(config-router)# redistribute iso-igrp remote route-map 1
Router(config-router)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match ip address 1
Router(config-route-map)# match clns address 2
Router(config-route-map)# set metric 5
Router(config-route-map)# set level level-2
Router(config-route-map)# exit
Router(config)# access-list 1 permit 172.18.0.0 0.0.255.255
Router(config)# 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 calledconditional 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.

```
Router(config)# route-map ospf-default permit
Router(config-route-map)# match ip address 1
Router(config-route-map)# set metric 5
Router(config-route-map)# set metric-type type-2
Router(config-route-map)# exit
Router(config)# access-list 1 172.20.0.0 0.0.255.255
Router(config)# router ospf 101
Router(config-router)# default-information originate route-map ospf-default
```

See the "Connecting to a Service Provider Using External BGP" module for more examples of BGP route-map configuration tasks and configuration examples.

Example Passive Interface

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:

```
Router(config)# interface Ethernet 0
Router(config-if)# ip address 172.18.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Ethernet 1
Router(config-if)# ip address 172.18.2.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Ethernet 2
Router(config-if)# ip address 172.18.3.1 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 172.18.0.0 0.0.255.255 area 0

Router(config-router)# exit
```

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

```
Router(config)# router ospf 1
Router(config-router)# network 172.18.0.0 0.0.255.255 area 0
Router(config-router)# passive-interface Ethernet 2
Router(config-router)# exit
```

- Example Default Passive Interface, page 37

Example Default Passive Interface

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

```
Router(config)# interface Ethernet 0
Router(config-if)# ip address 172.19.64.38 255.255.255.0 secondary
Router(config-if)# ip address 172.19.232.70 255.255.255.240
Router(config-if)# no ip directed-broadcast
Router(config-if)# exit
Router(config)# interface Serial 0
Router(config-if)# ip address 172.24.101.14 255.255.255.252
Router(config-if)# no ip directed-broadcast
Router(config-if)# no ip mroute-cache
Router(config-if)# exit
Router(config)# interface TokenRing 0
Router(config-if)# ip address 172.20.10.4 255.255.255.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# no ip mroute-cache
Router(config-if)# ring-speed 16
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# passive-interface default
Router(config-router)# no passive-interface Serial 0
Router(config-router)# network 172.16.10.0 0.0.0.255 area 0
Router(config-router)# network 172.19.232.0 0.0.0.255 area 4
Router(config-router)# network 172.24.101.0 0.0.0.255 area 4
Router(config-router)# exit
```

Example Policy-Based Routing

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.

```
Router(config)# access-list 1 permit ip 10.1.1.1
Router(config)# access-list 2 permit ip 172.17.2.2
Router(config)# interface async 1
Router(config-if)# ip policy route-map equal-access
Router(config-if)# exit
Router(config)# route-map equal-access permit 10
Router(config-route-map)# match ip address 1
Router(config-route-map)# set ip default next-hop 172.16.6.6
Router(config-route-map)# exit
Router(config)# route-map equal-access permit 20
Router(config-route-map)# match ip address 2
Router(config-route-map)# set ip default next-hop 192.168.7.7
Router(config-route-map)# exit
Router(config)# route-map equal-access permit 30
Router(config-route-map)# set default interface null 0
Router(config-route-map)# exit
```

Example Policy Routing with CEF

The following example configures policy routing with CEF. The route is configured to verify that next hop 10.0.0.8 of the route map named test1 is a CDP neighbor before the router tries to policy-route to it.

```
Router(config)# ip cef
Router(config)# interface Ethernet 0/0/1
Router(config-if)# ip route-cache flow
Router(config-if)# ip policy route-map test
Router(config-if)# exit
Router(config)# route-map test permit 10
Router(config-route-map)# match ip address 1
Router(config-route-map)# set ip precedence priority
Router(config-route-map)# set ip next-hop 10.0.0.8
Router(config-route-map)# set ip next-hop verify-availability
Router(config-route-map)# exit
Router(config)# route-map test permit 20
Router(config-route-map)# match ip address 101
Router(config-route-map)# set interface Ethernet 0/0/3
Router(config-route-map)# set ip tos max-throughput
Router(config-route-map)# exit
```

Examples QoS Policy Propagation via BGP Configuration

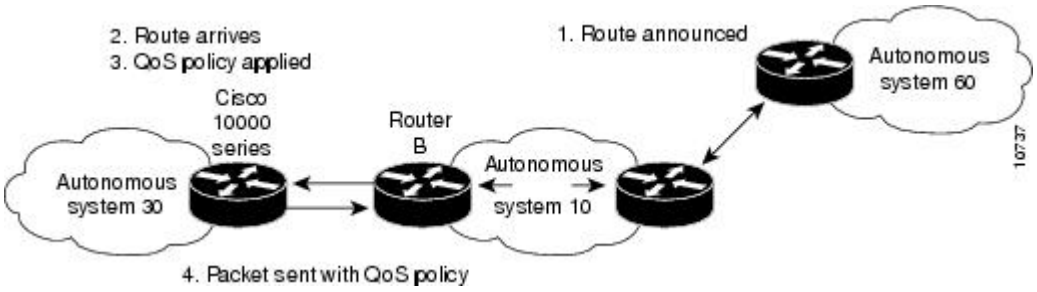
The following example shows how to create route maps to match access lists, BGP community lists, and BGP autonomous system paths, and apply IP precedence to routes learned from neighbors.

For information on how to configure QoS Policy Propagation via BGP, see the section Configuring QoS Policy Propagation via BGP, page 17 in this document.

In the figure below, Router A (Cisco 10000 Series) learns routes from autonomous system 10 and autonomous system 60. QoS policy is applied to all packets that match the defined route maps. Any packets

from Router A (Cisco 10000 Series) to autonomous system 10 or autonomous system 60 are sent the appropriate QoS policy, as the numbered steps indicate.

Figure 4 Router Learning Routes and Applying QoS Policy



Router A (Cisco 10000 Series) Configuration

```
interface serial 5/0/0/1:0
ip address 10.28.38.2 255.255.255.0
bgp-policy destination ip-prec-map
no ip mroute-cache
no cdp enable
frame-relay interface-dlci 20 IETF
router bgp 30
 table-map precedence-map
 neighbor 10.20.20.1 remote-as 10
 neighbor 10.20.20.1 send-community
!
ip bgp-community new-format
!
! Match community 1 and set the IP Precedence to priority
route-map precedence-map permit 10
 match community 1
 set ip precedence priority
!
! Match community 2 and set the IP Precedence to immediate
route-map precedence-map permit 20
 match community 2
 set ip precedence immediate
!
! Match community 3 and set the IP Precedence to flash
route-map precedence-map permit 30
 match community 3
 set ip precedence flash
!
! Match community 4 and set the IP Precedence to flash-override
route-map precedence-map permit 40
 match community 4
 set ip precedence flash-override
!
! Match community 5 and set the IP Precedence to critical
route-map precedence-map permit 50
 match community 5
 set ip precedence critical
!
! Match community 6 and set the IP Precedence to internet
route-map precedence-map permit 60
 match community 6
 set ip precedence internet
!
! Match community 7 and set the IP Precedence to network
route-map precedence-map permit 70
 match community 7
 set ip precedence network
!
! Match ip address access list 69 or match AS path 1
```

```
! and set the IP Precedence to critical
route-map precedence-map permit 75
 match ip address 69
 match as-path 1
 set ip precedence critical
!
! For everything else, set the IP Precedence to routine
route-map precedence-map permit 80
 set ip precedence routine
!
! Define the community lists
ip community-list 1 permit 60:1
ip community-list 2 permit 60:2
ip community-list 3 permit 60:3
ip community-list 4 permit 60:4
ip community-list 5 permit 60:5
ip community-list 6 permit 60:6
ip community-list 7 permit 60:7
!
! Define the AS path
ip as-path access-list 1 permit ^10_60
!
! Define the access list
access-list 69 permit 10.69.0.0
```

Router B Configuration

```
router bgp 10
 neighbor 10.30.30.1 remote-as 30
 neighbor 10.30.30.1 send-community
 neighbor 10.30.30.1 route-map send_community out
!
ip bgp-community new-format
!
! Match prefix 10 and set community to 60:1
route-map send_community permit 10
 match ip address 10
 set community 60:1
!
! Match prefix 20 and set community to 60:2
route-map send_community permit 20
 match ip address 20
 set community 60:2
!
! Match prefix 30 and set community to 60:3
route-map send_community permit 30
 match ip address 30
 set community 60:3
!
! Match prefix 40 and set community to 60:4
route-map send_community permit 40
 match ip address 40
 set community 60:4
!
! Match prefix 50 and set community to 60:5
route-map send_community permit 50
 match ip address 50
 set community 60:5
!
! Match prefix 60 and set community to 60:6
route-map send_community permit 60
 match ip address 60
 set community 60:6
!
! Match prefix 70 and set community to 60:7
route-map send_community permit 70
 match ip address 70
 set community 60:7
!
! For all others, set community to 60:8
route-map send_community permit 80
 set community 60:8
```

```
!
! Define the access lists
access-list 10 permit 10.61.0.0
access-list 20 permit 10.62.0.0
access-list 30 permit 10.63.0.0
access-list 40 permit 10.64.0.0
access-list 50 permit 10.65.0.0
access-list 60 permit 10.66.0.0
access-list 70 permit 10.67.0.0
```

Examples Key Management

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.

```
Router(config)# interface Ethernet 0
Router(config-if)# ip rip authentication key-chain trees
Router(config-if)# ip rip authentication mode md5
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 172.19.0.0
Router(config-router)# version 2
Router(config-router)# exit
Router(config)# key chain trees
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string willow
Router(config-keychain-key)# key 2
Router(config-keychain-key)# key-string chestnut
Router(config-keychain-key)# accept-lifetime 13:30:00 Jan 25 2005 duration 7200
Router(config-keychain-key)# send-lifetime 14:00:00 Jan 25 2005 duration 3600
Router(config-keychain-key)# key 3
Router(config-keychain-key)# key-string birch
Router(config-keychain-key)# accept-lifetime 14:30:00 Jan 25 2005 duration 7200
Router(config-keychain-key)# send-lifetime 15:00:00 Jan 25 2005 duration 3600
Router(config-keychain-key)# exit
```

The following example configures a key chain named trees:

```
Router(config)# key chain trees
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string willow
Router(config-keychain-key)# key 2
Router(config-keychain-key)# key-string chestnut
Router(config-keychain-key)# accept-lifetime 00:00:00 Dec 5 2004 23:59:59 Dec 5 2005
Router(config-keychain-key)# send-lifetime 06:00:00 Dec 5 2004 18:00:00 Dec 5 2005
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# interface Ethernet 0
Router(config-if)# ip address 172.19.104.75 255.255.255.0 secondary 172.19.232.147
255.255.255.240
Router(config-if)# ip rip authentication key-chain trees
Router(config-if)# media-type 10BaseT
Router(config-if)# exit
Router(config)# interface Ethernet 1
Router(config-if)# no ip address
Router(config-if)# shutdown
Router(config-if)# media-type 10BaseT
Router(config-if)# exit
Router(config)# interface Fddi 0
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# no keepalive
Router(config-if)# exit
Router(config)# interface Fddi 1
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# ip rip send version 1
Router(config-if)# ip rip receive version 1
```

```
Router(config-if)# no keepalive
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# version 2
Router(config-router)# network 172.19.0.0
Router(config-router)# network 10.0.0.0
Router(config-router)# network 172.16.0.0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP Routing Protocol Independent Commands	

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring IP Routing Protocol-Independent Features

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3 Feature Information for Configuring IP Routing Protocol-Independent Features

Feature Name	Releases	Feature Information
Default Passive Interface	12.0	In Internet service provider (ISP) and large enterprise networks, many of the distribution routers have more than 200 interfaces. Obtaining routing information from these interfaces required configuration of the routing protocol on all interfaces and manual configuration of the **passive-interface** command on the interfaces where adjacency was not desired. The Default Passive Interface feature simplifies the configuration of distribution routers by allowing all interfaces to be set as passive by default using a single **passive-interface default** command, and then by configuring individual interfaces where adjacencies are desired using the **no passive-interface** command.
Fast-Switched Policy Routing	11.3	IP policy routing can also 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.
IP Routing	11.0 Cisco IOS XE Release 3.1.0SG	The IP Routing feature introduced basic IP routing features that are documented throughout this document and also in other IP Routing Protocol documents.

Feature Name	Releases	Feature Information
NetFlow Policy Routing (NPR)	12.0(3)T	NetFlow policy routing (NPR) integrates policy routing, which enables traffic engineering and traffic classification, with NetFlow services, which provide billing, capacity planning, and monitoring information on real-time traffic flows. IP policy routing works with Cisco Express Forwarding (CEF), distributed CEF (dCEF), and NetFlow.
Policy-Based Routing	11.0	The Policy-Based Routing feature introduced a more flexible mechanism for routing packets than destination routing. Policy-based routing is a process where a router puts packets through a route map before routing the packets. The route map determines which packets are routed to which router next. The following command was introduced by this feature: **ip policy route-map**.
Policy-Based Routing (PBR) Default Next-Hop Route	12.1(11)E	The Policy-Based Routing (PBR) Default Next-Hop Route feature introduces the ability for packets that are forwarded as a result of the **set ip default next-hop** command to be switched at the hardware level. In prior releases, the router packets to be forwarded that are generated from the route map for PBR are switched at the software level. The following command was modified by this feature: **set ip default next-hop**.

Feature Name	Releases	Feature Information
Policy Routing Infrastructure	12.2(15)T	The Policy Routing Infrastructure feature provides full support of IP policy-based routing in conjunction with Cisco Express Forwarding (CEF) and NetFlow. As CEF gradually obsoletes fast switching, policy routing is integrated with CEF to increase customer performance requirements. When both policy routing and NetFlow are enabled, redundant processing is avoided.
QoS Policy Propagation via BGP	12.0	The QoS Policy Propagation via BGP feature allows you to classify packets by IP precedence based on BGP community lists, BGP autonomous system paths, and access lists. After a packet has been classified, you can use other QoS features such as committed access rate (CAR) and Weighted Random Early Detection (WRED) to specify and enforce policies to fit your business model.

IP Event Dampening

The IP Event Dampening feature introduces a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network operator to configure a router to automatically identify and selectively dampen a local interface that is flapping.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for IP Event Dampening

Subinterface Restrictions

Only primary interfaces can be configured with this feature. The primary interface configuration is applied to all subinterfaces by default. IP Event Dampening does not track the flapping of individual subinterfaces on an interface.

Virtual Templates Not Supported

Copying a dampening configuration from virtual templates to virtual access interfaces is not supported because dampening has limited usefulness to existing applications that use virtual templates. Virtual access interfaces are released when an interface flaps, and new connections and virtual access interfaces are

acquired when the interface comes up and is made available to the network. Since dampening states are attached to the interface, the dampening states would not survive an interface flap.

IPX Routing Protocols Not Supported

Internetwork Packet Exchange (IPX) protocols are not supported by the IP Event Dampening feature. However, IPX variants of these protocols will still receive up and down state event information when this feature is enabled. This should not create any problems or routing issues.

Information About IP Event Dampening

IP Event Dampening Overview

Interface state changes occur when interfaces are administratively brought up or down or if an interface changes state. When an interface changes state or flaps, routing protocols are notified of the status of the routes that are affected by the change in state. Every interface state change requires all affected devices in the network to recalculate best paths, install or remove routes from the routing tables, and then advertise valid routes to peer routers. An unstable interface that flaps excessively can cause other devices in the network to consume substantial amounts of system processing resources and cause routing protocols to lose synchronization with the state of the flapping interface.

The IP Event Dampening feature introduces a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network operator to configure a router to automatically identify and selectively dampen a local interface that is flapping. Dampening an interface removes the interface from the network until the interface stops flapping and becomes stable. Configuring the IP Event Dampening feature improves convergence times and stability throughout the network by isolating failures so that disturbances are not propagated. This, in turn, reduces the utilization of system processing resources by other devices in the network and improves overall network stability.

Interface State Change Events

This section describes the interface state change events of the IP Event Dampening feature. This feature employs a configurable exponential decay mechanism that is used to suppress the effects of excessive interface flapping or state changes. When the IP Event Dampening feature is enabled, flapping interfaces are dampened from the perspective of the routing protocol by filtering excessive route updates. Flapping interfaces are identified, assigned penalties, suppressed if necessary, and made available to the network when the interface stabilizes. Figure 1 displays interface state events as they are perceived by routing protocols.

Suppress Threshold

The suppress threshold is the value of the accumulated penalty that triggers the router to dampen a flapping interface. The flapping interface is identified by the router and assigned a penalty for each up and down state change, but the interface is not automatically dampened. The router tracks the penalties that a flapping interface accumulates. When the accumulated penalty reaches the default or preconfigured suppress threshold, the interface is placed in a dampened state.

Half-Life Period

The half-life period determines how fast the accumulated penalty can decay exponentially. When an interface is placed in a dampened state, the router monitors the interface for additional up and down state changes. If the interface continues to accumulate penalties and the interface remains in the suppress threshold range, the interface will remain dampened. If the interface stabilizes and stops flapping, the penalty is reduced by half after each half-life period expires. The accumulated penalty will be reduced until the penalty drops to the reuse threshold. The configurable range of the half-life period timer is from 1 to 30 seconds. The default half-life period timer is 5 seconds.

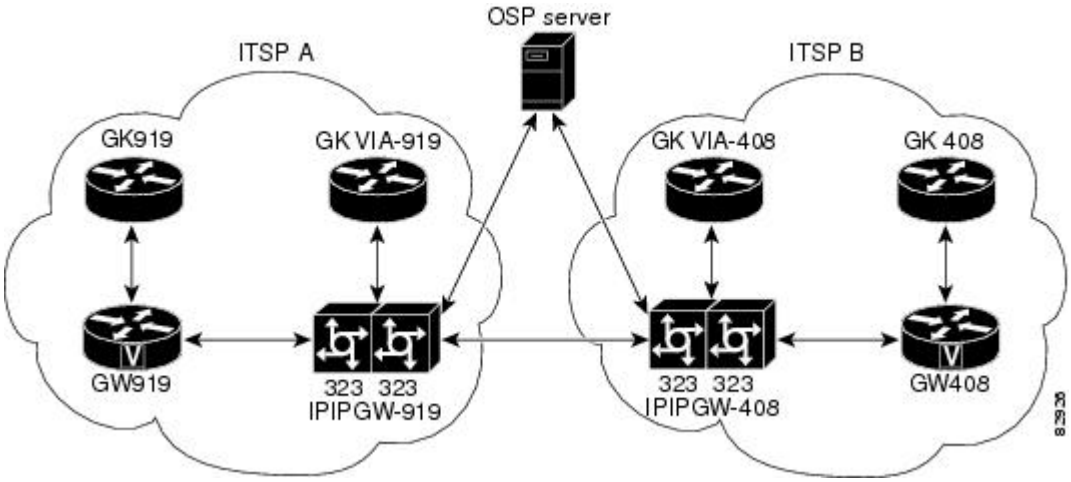
Reuse Threshold

When the accumulated penalty decreases until the penalty drops to the reuse threshold, the route is unsuppressed and made available to other devices in the network. The range of the reuse value is from 1 to 20000 penalties. The default value is 1000 penalties.

Maximum Suppress Time

The maximum suppress time represents the maximum time an interface can remain dampened when a penalty is assigned to an interface. The maximum suppress time can be configured from 1 to 20000 seconds. The default maximum penalty timer is 20 seconds or four times the default half-life period (5 seconds). The maximum value of the accumulated penalty is calculated based on the maximum suppress time, reuse threshold, and half-life period.

Figure 5 Interface State Change Events Perceived by the Routing Protocols



Affected Components

When an interface is not configured with dampening, or when an interface is configured with dampening but is not suppressed, the routing protocol behavior as a result of interface state transitions is not changed by the IP Event Dampening feature. However, if an interface is suppressed, the routing protocols and routing tables are immune to any further state transitions of the interface until it is unsuppressed.

Route Types

The following interfaces are affected by the configuration of this feature:

- Connected routes:
  - The connected routes of dampened interfaces are not installed into the routing table.
  - When a dampened interface is unsuppressed, the connected routes will be installed into the routing table if the interface is up.
- Static routes:
  - Static routes assigned to a dampened interface are not installed into the routing table.
  - When a dampened interface is unsuppressed, the static route will be installed into the routing table if the interface is up.

Note Only the primary interface can be configured with this feature, and all subinterfaces are subject to the same dampening configuration as the primary interface. IP Event Dampening does not track the flapping of individual subinterfaces on an interface.

Supported Protocols

The IP Event Dampening feature supports Border Gateway Protocol (BGP), Connectionless Network Services (CLNS), Enhanced Interior Gateway Routing Protocol (EIGRP), Intermediate System-to-Intermediate System (IS-IS), Hot Standby Routing Protocol (HSRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP). The following list provides some general information about the operation of this feature with these protocols.

- BGP, EIGRP, IS-IS, RIP, and OSPF:
  - When an interface is dampened, the interface is considered to be down by the routing protocol. The routing protocol will not hold any adjacencies with this peer router over the dampened interface or generate advertisements of any routes related to this interface to other peer routers.
  - When the interface is unsuppressed and made available to the network, the interface will be considered by the routing protocols to be up. The routing protocols will be notified that the interface is in an up state and routing conditions will return to normal.
- HSRP:
  - When an interface is dampened, it is considered to be down by HSRP. HSRP will not generate HSRP messages out of the dampened interface or respond to any message received by the dampened interface. When the interface is unsuppressed and made available to the network, HSRP will be notified of the up state and will return to normal operations.

- CLNS:
  - When an interface is dampened, the interface is dampened to both IP and CLNS routing equally. The interface is dampened to both IP and CLNS because integrated routing protocols like IS-IS, IP, and CLNS routing are closely interconnected, so it is impossible to apply dampening separately.

Note The IP Event Dampening feature has no effect on any routing protocols if it is not enabled or an interface is not dampened.

Network Deployments

In real network deployments, some routers may not be configured with interface dampening, and all routers may not even support this feature. No major routing issues are expected, even if the router at the other end of a point-to-point interface or routers of the same multicast LAN do not have interface dampening turned on or do not have this feature implemented. On the router, where the interface is dampened, routes associated with the interface will not be used. No packets will be sent out of this interface, and no routing protocol activity will be initiated with routers on the other side of the interface. However, routers on the other side can still install some routes, in their routing tables, that are associated with this subnet because the routers recognize that their own interfaces are up and can start forwarding packets to the dampened interface. In such situations, the router with the dampened interface will start forwarding these packets, depending on the routes in its routing table.

The IP Event Dampening feature does not introduce new information into the network. In fact, the effect of dampening is to subtract a subset of routing information from the network. Therefore, looping should not occur as a result of dampening.

Benefits of IP Event Dampening

Reduced Processing Load

The IP Event Dampening Feature employs a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols. Excessive interface up and down state changes that are received in a short period of time are not processed and do not consume system resources. Other routers in the network need not waste system resources because of a flapping route.

Faster Convergence

The IP Event Dampening feature improves convergence times and stability throughout the network by isolating failures so that disturbances are not propagated. Routers that are not experiencing link flap reach convergence sooner, because routing tables are not rebuilt each time the offending router leaves and enters the service

Improved Network Stability

The IP Event Dampening feature provides increased network stability. A router with a flapping interface removes the flapping interface from the network until the interface stabilizes, so other routers simply redirect traffic around the affected router until the interface becomes stable, which ensures that the router loses no data packets.

How to Configure IP Event Dampening

Enabling IP Event Dampening

The **dampening** command is entered in interface configuration mode to enable the IP Event Dampening feature. If this command is applied to an interface that already has dampening configured, all dampening states are reset and the accumulated penalty will be set to 0. If the interface has been dampened, the accumulated penalty will fall into the reuse threshold range, and the dampened interface will be made available to the network. The flap counts, however, are retained.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **dampening** [*half-life-period reuse-threshold*] [*suppress-threshold max-suppress* [*restart-penalty*]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Enters interface configuration mode and configures the specified interface.

	Command or Action	Purpose
Step 4	**dampening** [*half-life-period reuse-threshold*] [*suppress-threshold max-suppress* [*restart-penalty*]] **Example:** `Router(config-if)# dampening`	Enables interface dampening. • Entering the **dampening** command without any arguments enables interface dampening with default configuration parameters. • When manually configuring the timer for the *restart-penalty* argument, the values must be manually entered for all arguments.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying IP Event Dampening

Use the **show dampening interface** or **show interface dampening** commands to verify the configuration of the IP Event Dampening feature.

Note The **clear counters** command can be used to clear the flap count and reset it to zero. All other parameters and status, including dampening states and accumulated penalties, are not affected by this command.

SUMMARY STEPS

1. **enable**
2. **show dampening interface**
3. **show interface dampening**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show dampening interface** **Example:** `Router# show dampening interface`	Displays dampened interfaces.

	Command or Action	Purpose
Step 3	**show interface dampening** **Example:** `Router# show interface dampening`	Displays dampened interfaces on the local router.

Configuration Examples for IP Event Dampening

Example: Enabling IP Event Dampening

The following example shows how to enable interface dampening on Ethernet interface 0/0 and sets the half life to 30 seconds, the reuse threshold to 1500, the suppress threshold to 10000, and the maximum suppress time to 120 seconds:

```
interface Ethernet 0/0
 dampening 30 1500 10000 120
```

The following example shows how to enable interface dampening on ATM interface 6/0 and uses the default interface dampening values:

```
interface atm 6/0
 dampening
```

The following example shows how to configure the router to apply a penalty of 500 on Ethernet interface 0/0 when the interface comes up for the first time after the router is reloaded:

```
interface Ethernet 0/0
 dampening 5 500 1000 20 500
```

Example: Verifying IP Event Dampening

The following sample output from the **show dampening interface** command displays a summary of interface dampening:

```
Router# show dampening interface
3 interfaces are configured with dampening.
No interface is being suppressed.
Features that are using interface dampening:
  IP Routing
  CLNS Routing
```

The following sample output from the **show interface dampening** command displays the summary of the dampening parameters and the status of the interfaces on the local router:

```
Router# show interface dampening
FastEthernet0/0
  Flaps Penalty    Supp ReuseTm   HalfL  ReuseV   SuppV  MaxSTm    MaxP Restart
```

```
      0       0   FALSE       0       5    1000    2000      20   16000       0
ATM2/0
  Flaps Penalty    Supp ReuseTm   HalfL  ReuseV   SuppV  MaxSTm    MaxP Restart
      0       0   FALSE       0       5    1000    2000      20   16000       0
POS6/0
  Flaps Penalty    Supp ReuseTm   HalfL  ReuseV   SuppV  MaxSTm    MaxP Restart
      0       0   FALSE       0       5    1000    2000      20   16000       0
```

Additional References

Related Documents

Related Topic	Document Title
IP Routing Protocol-Independent commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Routing: Protocol-Independent Command Reference
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IP Event Dampening

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4 Feature Information for IP Event Dampening

Feature Name	Releases	Feature Information
IP Event Dampening	12.0(22)S 12.2(14)S 12.2(13)T 12.2(18)SXD Cisco IOS XE 3.1.0SG	The IP Event Dampening feature introduces a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network operator to configure a router to automatically identify and selectively dampen a local interface that is flapping. The following commands were introduced or modified: **dampening**, **debug dampening**, **show dampening interface**, **show interface dampening**.

Glossary

event dampening--The process in which a router dampens a flapping interface from the perspective of the routing tables and routing protocols of IP and CLNS by filtering the excessive route adjust message because of the interface state change.

flap--Rapid interface state changes from up to down and down to up within a short period of time.

half life--The rate of the exponential decay of the accumulated penalty is determined by this value.

maximum penalty--The maximum value beyond which the penalty assigned does not increase. It is derived from the maximum suppress time.

maximum suppress time--The maximum amount of time the interface can stay suppressed at the time a penalty is assigned.

penalty--A value assigned to an interface when it flaps. This value increases with each flap and decreases over time. The rate at which it decreases depends on the half life.

reuse threshold --The threshold value after which the interface will be unsuppressed and can be used again.

suppress threshold--Value of the accumulated penalty that triggers the router to dampen a flapping interface. When the accumulated penalty exceeds this value, the interface state is considered to be down from the perspective of the routing protocol.

suppressed--Suppressing an interface removes an interface from the network from the perspective of the routing protocol. An interface enters the suppressed state when it has flapped frequently enough for the penalty assigned to it to cross a threshold limit.

PBR Support for Multiple Tracking Options

The PBR Support for Multiple Tracking Options feature extends the capabilities of object tracking using Cisco Discovery Protocol (CDP) to allow the policy-based routing (PBR) process to verify object availability by using additional methods. The verification method can be an Internet Control Message Protocol (ICMP) ping, a User Datagram Protocol (UDP) ping, or an HTTP GET request.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About PBR Support for Multiple Tracking Options

Object Tracking

Object tracking is an independent process that monitors objects such as the following:

- State of the line protocol of an interface
- Existence of an entry in the routing table
- Results of a Service Assurance Agent (SAA) operation, such as a ping

Clients such as Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), Gateway Load Balancing Protocol (GLBP), and (with this feature) PBR can register their interest in specific, tracked objects and then take action when the state of the objects changes.

PBR Support for Multiple Tracking Options Feature Design

The PBR Support for Multiple Tracking Options feature gives PBR access to all the objects that are available through the tracking process. The tracking process provides the ability to track individual objects--such as ICMP ping reachability, routing adjacency, an application running on a remote device, a route in the Routing Information Base (RIB)--or to track the state of an interface line protocol.

Object tracking functions in the following manner. PBR will inform the tracking process that a certain object should be tracked. The tracking process will in turn notify PBR when the state of that object changes.

How to Configure PBR Support for Multiple Tracking Options

The tasks in this section are divided according to the Cisco IOS release that you are running because Cisco IOS Release 12.3(14)T introduced new syntax for IP Service Level Agreements (SLAs). To use this feature, you must be running Cisco IOS Release 12.3(4)T, 12.2(25)S, or a later release. This section contains the following tasks:

- Cisco IOS Release 12.3(11)T 12.2(25)S and Earlier, page 60
- Cisco IOS Release 12.3(14)T 12.2(33)SXH and Later, page 64

Cisco IOS Release 12.3(11)T 12.2(25)S and Earlier

Perform this task to configure PBR support for multiple tracking options. In this task, a route map is created and configured to verify the reachability of the tracked object.

This task requires the networking device to be running Cisco IOS Release 12.3(11)T, 12.2(25)S, or prior releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **rtr** *operation-number*
4. **type echo protocol** *protocol-type target* [**source-ipaddr** *ip-address*]
5. **exit**
6. **rtr schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*]
7. **track** *object-number* **rtr** *entry-number* [**reachability**]
8. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask* [**secondary**]
12. **ip policy route-map** *map-tag*
13. **exit**
14. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
15. **set ip next-hop verify-availability** [*next-hop-address sequence* **track** *object*]
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**rtr** *operation-number* **Example:** `Router(config)# rtr 1`	Enters SAA RTR configuration mode and configures an SAA operation.

	Command or Action	Purpose
Step 4	**type echo protocol** *protocol-type target* [**source-ipaddr** *ip-address*] **Example:** `Router(config-rtr)# type echo protocol ipicmpecho 10.1.1.10`	Configures an SAA end-to-end echo response time probe operation.
Step 5	**exit** **Example:** `Router(config-rtr)# exit`	Exits SAA RTR configuration mode and returns the router to global configuration mode.
Step 6	**rtr schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] **Example:** `Router(config)# rtr schedule 1 life forever start-time now`	Configures the time parameters for the SAA operation.
Step 7	**track** *object-number* **rtr** *entry-number* [**reachability**] **Example:** `Router(config)# track 123 rtr 1 reachability`	Tracks the reachability of a Response Time Reporter (RTR) object and enters tracking configuration mode.
Step 8	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Router(config-track)# delay up 60 down 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 9	**exit** **Example:** `Router(config-track)# exit`	Exits tracking configuration mode and returns the router to global configuration mode.

	Command or Action	Purpose
Step 10	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface type and number and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.1.1.11 255.0.0.0`	Specifies a primary or secondary IP address for an interface. • See the "Configuring IPv4 Addresses" chapter of the *Cisco IOS IP Addressing Services Configuration Guide* for information on configuring IPv4 addresses.
Step 12	**ip policy route-map** *map-tag* **Example:** `Router(config-if)# ip policy route-map alpha`	Enables policy routing and identifies a route map to be used for policy routing.
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns the router to global configuration mode.
Step 14	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map alpha`	Specifies a route map and enters route-map configuration mode.
Step 15	**set ip next-hop verify-availability** [*next-hop-address sequence* **track** *object*] **Example:** `Router(config-route-map)# set ip next-hop verify-availability 10.1.1.1 10 track 123`	Configures the route map to verify the reachability of the tracked object.
Step 16	**end** **Example:** `Router(config-route-map)# end`	Exits route-map configuration mode and returns the router to privileged EXEC mode.

Cisco IOS Release 12.3(14)T 12.2(33)SXH and Later

Perform this task to configure PBR support for multiple tracking options. In this task, a route map is created and configured to verify the reachability of the tracked object.

This task requires the networking device to be running Cisco IOS Release 12.3(14)T, 12.2(33)SXH, or a later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla monitor** *operation-number*
4. **type echo protocol ipIcmpEcho** {*destination-ip-address*| *destination-hostname*}[**source-ipaddr** {*ip-address*| *hostname*} | **source-interface** *interface-name*]
5. **exit**
6. **ip sla monitor schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] [**recurring**]
7. **track** *object-number* **rtr** *entry-number* [**reachability**| **state**]
8. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask* [**secondary**]
12. **ip policy route-map** *map-tag*
13. **exit**
14. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
15. **set ip next-hop verify-availability** [*next-hop-address sequence* **track** *object*]
16. **end**
17. **show track** *object-number*
18. **show route-map** [*map-name*| **all**| **dynamic**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip sla monitor** *operation-number* **Example:** `Router(config)# ip sla monitor 1`	Starts a Cisco IOS IP Service Level Agreement (SLA) operation configuration and enters IP SLA monitor configuration mode.
Step 4	**type echo protocol ipIcmpEcho** {*destination-ip-address*| *destination-hostname*}[**source-ipaddr** {*ip-address*| *hostname*} | **source-interface** *interface-name*] **Example:** `Router(config-sla-monitor)# type echo protocol ipIcmpEcho 10.1.1.1`	Configures an IP SLA Internet Control Message Protocol (ICMP) echo probe operation.
Step 5	**exit** **Example:** `Router(config-sla-monitor)# exit`	Exits IP SLA monitor configuration mode and returns the router to global configuration mode.
Step 6	**ip sla monitor schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] [**recurring**] **Example:** `Router(config)# ip sla monitor schedule 1 life forever start-time now`	Configures the scheduling parameters for a single Cisco IOS IP SLA operation. • In this example, the time parameters for the IP SLA operation are configured.
Step 7	**track** *object-number* **rtr** *entry-number* [**reachability**| **state**] **Example:** `Router(config)# track 123 rtr 1 reachability`	Tracks the reachability of a Response Time Reporter (RTR) object and enters tracking configuration mode.
Step 8	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Router(config-track)# delay up 60 down 30`	(Optional) Specifies a period of time, in seconds, to delay communicating state changes of a tracked object.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(config-track)# exit`	Exits tracking configuration mode and returns the router to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface serial 2/0`	Specifies an interface type and number and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 192.168.1.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface. • See the "Configuring IPv4 Addresses" chapter of the *Cisco IOS IP Addressing Services Configuration Guide* for information on configuring IPv4 addresses. • In this example, the IP address of the incoming interface is specified. This is the interface on which policy routing is to be enabled.
Step 12	**ip policy route-map** *map-tag* **Example:** `Router(config-if)# ip policy route-map alpha`	Enables policy routing and identifies a route map to be used for policy routing.
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns the router to global configuration mode.
Step 14	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map alpha`	Specifies a route map and enters route-map configuration mode.
Step 15	**set ip next-hop verify-availability** [*next-hop-address sequence* **track** *object*] **Example:** `Router(config-route-map)# set ip next-hop verify-availability 10.1.1.1 10 track 123`	Configures the route map to verify the reachability of the tracked object. • In this example, the policy is configured to forward packets received on serial interface 2/0 to 10.1.1.1 if that device is reachable.

	Command or Action	Purpose
Step 16	**end** **Example:** `Router(config-route-map)# end`	Exits route-map configuration mode and returns the router to privileged EXEC mode.
Step 17	**show track** *object-number* **Example:** `Router# show track 123`	(Optional) Displays tracking information. • Use this command to verify the configuration. See the display output in the "Examples" section of this task.
Step 18	**show route-map** [*map-name*| **all**| **dynamic**] **Example:** `Router# show route-map alpha`	(Optional) Displays route map information. • In this example, information about the route map named alpha is displayed. See the display output in the "Examples" section of this task.

Examples

The following output from the **show track** command shows that the tracked object 123 is reachable.

```
Router# show track 123
Track 123
  Response Time Reporter 1 reachability
  Reachability is Up
    2 changes, last change 00:00:33
  Delay up 60 secs, down 30 secs
  Latest operation return code: OK
  Latest RTT (millisecs) 20
  Tracked by:
    ROUTE-MAP 0
```

The following output from the **show route-map** command shows information about the route map named alpha that was configured in the task.

```
Router# show route-map alpha
route-map alpha, permit, sequence 10
  Match clauses:
  Set clauses:
    ip next-hop verify-availability 10.1.1.1 10 track 123  [up]
  Policy routing matches: 0 packets, 0 bytes
```

Configuration Examples for PBR Support for Multiple Tracking Options

Cisco IOS Release 12.3(11)T 12.2(25)S and Earlier

In the following example, object tracking is configured for PBR on routers that are running Cisco IOS Release 12.3(11)T, 12.2(25)S, or earlier releases.

The configured policy is that packets received on Ethernet interface 0, should be forwarded to 10.1.1.1 only if that device is reachable (responding to pings). If 10.1.1.1 is not up, then the packets should be forwarded to 10.2.2.2. If 10.2.2.2 is also not reachable, then the policy routing fails and the packets are routed according to the routing table.

Two Response Time Reporters (RTRs) are configured to ping the remote devices. The RTRs are then tracked. Policy routing will monitor the state of the tracked RTRs and make forwarding decisions based on their state.

```
! Define and start the RTRs.
rtr 1
 type echo protocol ipicmpecho 10.1.1.1
rtr schedule 1 start-time now life forever
!
rtr 2
 type echo protocol ipicmpecho 10.2.2.2
rtr schedule 2 start-time now life forever
!
! Track the RTRs.
track 123 rtr 1 reachability
track 124 rtr 2 reachability
!
! Enable policy routing on the incoming interface.
interface ethernet 0
 ip address 10.4.4.4 255.255.255.0
 ip policy route-map beta
!
! 10.1.1.1 is via this interface.
interface ethernet 1
 ip address 10.1.1.254 255.255.255.0
!
! 10.2.2.2 is via this interface.
interface ethernet 2
 ip address 10.2.2.254 255.255.255.0
!
! Define a route map to set the next-hop depending on the state of the tracked RTRs.
route-map beta
 set ip next-hop verify-availability 10.1.1.1 10 track 123
 set ip next-hop verify-availability 10.2.2.2 20 track 124
```

Cisco IOS Release 12.3(14)T 12.2(33)SXH and Later

In the following example, object tracking is configured for PBR on routers running Cisco IOS Release 12.3(14)T, 12.2(33)SXH, and later releases.

The configured policy is that packets received on Ethernet interface 0, should be forwarded to 10.1.1.1 only if that device is reachable (responding to pings). If 10.1.1.1 is not up, then the packets should be forwarded to 10.2.2.2. If 10.2.2.2 is also not reachable, then the policy routing fails and the packets are routed according to the routing table.

Two RTRs are configured to ping the remote devices. The RTRs are then tracked. Policy routing will monitor the state of the tracked RTRs and make forwarding decisions based on their state.

```
! Define and start the RTRs.
ip sla monitor 1
 type echo protocol ipicmpecho 10.1.1.1
ip sla monitor schedule 1 start-time now life forever
!
```

```
ip sla monitor 2
 type echo protocol ipicmpecho 10.2.2.2
ip sla monitor schedule 2 start-time now life forever
!
! Track the RTRs.
track 123 rtr 1 reachability
track 124 rtr 2 reachability
!
! Enable policy routing on the incoming interface.
interface ethernet 0
 ip address 10.4.4.4 255.255.255.0
 ip policy route-map beta
!
! 10.1.1.1 is via this interface.
interface ethernet 1
 ip address 10.1.1.254 255.255.255.0
!
! 10.2.2.2 is via this interface.
interface ethernet 2
 ip address 10.2.2.254 255.255.255.0
!
! Define a route map to set the next-hop depending on the state of the tracked RTRs.
route-map beta
 set ip next-hop verify-availability 10.1.1.1 10 track 123
 set ip next-hop verify-availability 10.2.2.2 20 track 124
```

Additional References

The following sections provide references related to the PBR Support for Multiple Tracking Options feature.

Related Documents

Related Topic	Document Title
Object tracking within Cisco IOS software	Configuring Enhanced Object Tracking" chapter of the *Cisco IOS IP Application Services Configuration Guide*
Configuring IP addresses	"Configuring IPv4 Addresses" chapter of the *Cisco IOS IP Addressing Services Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Routing: Protocol-Independent Command Reference*. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases* , at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **set ip next-hop verify-availability**

Feature Information for PBR Support for Multiple Tracking Options

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5 Feature Information for PBR Support for Multiple Tracking Options

Feature Name	Releases	Feature Information
PBR Support for Multiple Tracking Options	12.3(4)T 12.2(25)S 12.2(33)SXH	The PBR Support for Multiple Tracking Options feature extends the capabilities of object tracking using Cisco Discovery Protocol (CDP) to allow the policy-based routing (PBR) process to verify object availability by using additional methods. The verification method can be an Internet Control Message Protocol (ICMP) ping, a User Datagram Protocol (UDP) ping, or an HTTP GET request. Due to syntax changes for IP SLAs, a new task and configuration example were introduced in the Cisco IOS Release 12.2(33)SXH. The following commands were introduced or modified by this feature: **set ip next-hop verify-availability**.

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.