IP Routing: Protocol-Independent Configuration Guide, Cisco IOS Release 15SY

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

CHAPTER 1

Configuring IP Routing Protocol-Independent Features

This module describes how to configure IP routing protocol-independent features. Some of the features discussed in this module include the Default Passive Interface, Fast-Switched Policy Routing, and Policy-Based Routing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 Basic IP Routing

Variable-Length Subnet Masks

Dynamic routing protocols, such as the Enhanced Interior Gateway Routing Protocol (EIGRP), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), and Routing Information Protocol

(RIP) Version 2, and static routes support variable-length subnet masks (VLSMs). VLSM enables an organization to use more than one subnet mask within the same network address space. VLSM allows you to conserve IP addresses and efficiently use the available address space. Implementing VLSM is often referred to as "subnetting a subnet."

Note You may want to carefully consider the use of VLSMs. It is easy to make mistakes during address assignments and difficult to monitor networks that use VLSMs. 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.

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 subnet 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 (172.16.255.0–72.16.255.252) with 2 host addresses on each subnet.

Note 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)# exit
Router(config)# interface Serial 0/0
Router(config-if)# ip address 172.16.255.5 255.255.255.252
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 172.16.0.0
```

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 in specifying a gateway of last resort to which all unroutable packets will be sent.

To configure a static route, use the **ip route** command in global configuration mode.

Static routes remain in the router configuration until you remove them (by using the **no** form of the **ip route** command). However, you can override static routes with dynamic routing information through the assignment of administrative distance values. An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers.

Each dynamic routing protocol has a default administrative distance, as listed in the table below. For a configured static route to be overridden, the administrative distance of the static route should be higher than that of the dynamic routing protocol.

Table 1: Default Administrative Distances

Route Source	Default Administrative Distance
Connected interface	0

Route Source	Default Administrative Distance
Static route	1
EIGRP summary route	5
External Border Gateway Protocol (BGP)	20
Internal EIGRP	90
Interior Gateway Routing Protocol (IGRP)	100
OSPF	110
IS-IS	115
RIP	120
Exterior Gateway Protocol (EGP)	140
On-Demand Routing (ODR)	160
External EIGRP	170
Internal BGP	200
Unknown	255

Static routes that point to an interface are advertised through dynamic routing protocols, regardless of whether **redistribute static** router configuration commands were specified for those routing protocols. Static routes that point to an interface are advertised because the routing table considers these routes as connected routes and hence, these routes lose their static nature. However, if you define a static route to an interface that is not connected to one of the networks defined by a **network** command, no dynamic routing protocol will advertise the route unless a **redistribute static** command is specified for the protocols.

When an interface goes down, all static routes associated with 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

Default routes, also known as gateways of last resort, are used to route packets that are addressed to networks not explicitly listed in the routing table. A device might not be able to determine routes to all networks. To provide complete routing capability, network administrators use some devices as smart devices and give the remaining devices default routes to the smart device. (Smart devices have routing table information for the entire internetwork.) Default routes can be either passed along dynamically or configured manually into individual devices.

Most dynamic interior routing protocols include a mechanism for causing a smart device to generate dynamic default information, which is then passed along to other devices.

You can configure a default route by using the following commands:

- **ip default-gateway**
- **ip default-network**
- **ip route 0.0.0.0 0.0.0.0**

You can use the **ip default-gateway** global configuration command to define a default gateway when IP routing is disabled on a device. For instance, if a device is a host, you can use this command to define a default gateway for the device. You can also use this command to transfer a Cisco software image to a device when the device is in boot mode. In boot mode, IP routing is not enabled on the device.

Unlike the **ip default-gateway** command, the **ip default-network** command can be used when IP routing is enabled on a device. When you specify a network by using the **ip default-network** command, the device considers routes to that network for installation as the gateway of last resort on the device.

Gateways of last resort configured by using the **ip default-network** command are propagated differently depending on which routing protocol is propagating the default route. For Interior Gateway Routing Protocol (IGRP) and Enhanced Interior Gateway Routing Protocol (EIGRP) to propagate the default route, the network specified by the **ip default-network** command must be known to IGRP or EIGRP. The network must be an IGRP- or EIGRP-derived network in the routing table, or the static route used to generate the route to the network must be redistributed into IGRP or EIGRP or advertised into these protocols by using the **network** command. The Routing Information Protocol (RIP) advertises a route to network 0.0.0.0 if a gateway of last resort is configured by using the **ip default-network** command. The network specified in the **ip default-network** command need not be explicitly advertised under RIP.

Creating a static route to network 0.0.0.0 0.0.0.0 by using the **ip route 0.0.0.0 0.0.0.0** command is another way to set the gateway of last resort on a device. As with the **ip default-network** command, using the static route to 0.0.0.0 is not dependent on any routing protocols. However, IP routing must be enabled on the device. IGRP does not recognize a route to network 0.0.0.0. Therefore, it cannot propagate default routes created by using the **ip route 0.0.0.0 0.0.0.0** command. Use the **ip default-network** command to have IGRP propagate a default route.

EIGRP propagates a route to network 0.0.0.0, but the static route must be redistributed into the routing protocol.

Depending on your release of the Cisco software, the default route created by using the **ip route 0.0.0.0 0.0.0.0** command is automatically advertised by RIP devices. In some releases, RIP does not advertise the default route if the route is not learned via RIP. You might have to redistribute the route into RIP by using the **redistribute** command.

Default routes created using the **ip route 0.0.0.0 0.0.0.0** command are not propagated by Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS). Additionally, these default routes cannot be redistributed into OSPF or IS-IS by using the **redistribute** command. Use the **default-information originate** command to generate a default route into an OSPF or IS-IS routing domain.

Default Network

Default networks are used to route packets to destinations not established in the routing table. You can use the **ip default-network** *network-number* global configuration command to configure a default network when IP routing is enabled on the device. When you configure a default network, the device considers routes to that network for installation as the gateway of last resort on the device.

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 the 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** 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), the network 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 the default network, the router considers this network as a candidate default path. The route candidates are examined and the best one is chosen based on the administrative distance and metric information. 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 paths in a routing table. Static routes always install six paths. The exception is BGP, which by default allows only one path (the best path) to the 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 in the *BGP Configuration Guide* for more information.

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

Routing Information Redistribution

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

You can also conditionally control the redistribution of routes between routing domains by configuring route maps between two domains. A route map is a route filter that is configured with permit and deny statements, match and set clauses, and sequence numbers. To define a route map for redistribution, use the **route-map** command in global configuration mode.

The metrics of one routing protocol do not necessarily translate into the metrics of another. For example, the RIP metric is 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 the creation of routing loops.

The following examples illustrate the use of redistribution with and without route maps. The following example shows how to redistribute all OSPF routes into EIGRP:

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

The following example shows how to redistribute 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-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 shows how to redistribute 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 shows how to redistribute OSPF intra-area and inter-area routes with next-hop routers on serial interface 0/0 into BGP with a 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 are 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
```

In the following example, 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
```

The following example shows how a route map is referenced by using the **default-information** router configuration command. Such referencing is called conditional default origination. OSPF will generate the default route (network 0.0.0.0) with a type 2 metric of 5 if 172.16.0.0 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 1 172.16.0.0 0.0.255.255
Router(config)# router ospf 101
Router(config-router)# default-information originate route-map ospf-default
```

Supported Automatic Metric Translations

This section describes supported automatic metric translations between routing protocols. The following points are based on the assumption 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 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.

Protocol Differences in Implementing the no redistribute Command

Caution Removing options that you have configured for the **redistribute** command requires careful use of the **no redistribute** command to ensure that you obtain the result that you are expecting. In most cases, changing or disabling any keyword will not affect the state of other keywords.

Different protocols implement the **no redistribute** command differently as follows:

- In Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP) configurations, the **no redistribute** command removes only the specified keywords from the **redistribute** commands in the running configuration. They use the *subtractive keyword* method when redistributing from other protocols. For example, in the case of BGP, if you configure **no redistribute static route-map interior**, only the route map is removed from the redistribution, leaving **redistribute static** in place with no filter.
- The **no redistribute isis** command removes the Intermediate System to Intermediate System (IS-IS) redistribution from the running configuration. IS-IS removes the entire command, regardless of whether IS-IS is the redistributed or redistributing protocol.
- The Enhanced Interior Gateway Routing Protocol (EIGRP) used the subtractive keyword method prior to EIGRP component version rel5. Starting with EIGRP component version rel5, the **no redistribute** command removes the entire **redistribute** command when redistributing from any other protocol.

Default Passive Interfaces

The Default Passive Interfaces feature simplifies the configuration of distribution devices by allowing all interfaces to be set as passive by default. In ISPs and large enterprise networks, many distribution devices have more than 200 interfaces. Obtaining routing information from these interfaces requires configuration of the routing protocol on all interfaces and manual configuration of the **passive-interface** command on interfaces where adjacencies were not desired.

Sources of Routing Information Filtering

Filtering sources of routing information prioritizes routing information gathered 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. Numerically, an administrative distance is an integer from 0 to 255. In general, the higher the value the lower the trust rating. An administrative distance of 255 means that the routing information source cannot be trusted at all and should be ignored.

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 on the same router for IP, the same route may be advertised by more than one routing process. By specifying administrative distance values, you enable a router to intelligently discriminate between sources of routing information. The router will always pick the route whose routing protocol has the lowest administrative distance.

There are no guidelines for assigning administrative distances because each network has its own requirements. You must determine a reasonable matrix of administrative distances for a network as a whole.

> **Note** You can use the administrative distance to rate the routing information from routers that are running the same routing protocol. However, using the administrative distance for this purpose can result in inconsistent routing information and forwarding loops.

In the following example, the **router eigrp** global configuration command configures EIGRP routing in autonomous system 1. The **network** command specifies EIGRP routing on networks 192.0.2.16 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 192.0.2.16
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 192.0.2.1 an administrative distance of 100 and all other routers on subnet 192.0.2.0 an administrative distance of 200:

```
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
```

However, if you reverse the order of these two commands, all routers on subnet 192.0.2.0 are assigned an administrative distance of 200, including the router at address 192.0.2.1:

```
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
```

> **Note** Administrative distances should be applied carefully and consistently to avoid the creation of routing loops or other network failures.

In the following example, the administrative distance value for learned IP routes 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
```

Policy-Based Routing

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

To enable PBR, you must identify the route map to be used for PBR and create the route map. The route map specifies the match criteria and the resulting action if all match clauses are met.

A packet arriving on a specified interface will be subject to PBR, 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** command in interface configuration mode.

To define the route map to be used for PBR, use the **route-map** command in global configuration mode.

To define the criteria by which packets are examined to learn if they will follow PBR, use either the **match length** command or the **match ip address** command or both in route map configuration mode. The **match length** command allows you to configure policy routing based on the Level 3 length of the packet, and the **match ip address** command allows you to policy route packets based on the criteria that can be matched with an extended access list.

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 packets. 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 packets. 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
```

You can set IP header precedence bits in the router when PBR is enabled. The precedence setting in the IP header determines how packets are treated during times of high traffic. When packets containing these headers arrive at another router, the packets are ordered for transmission according to the precedence set if the queuing feature is enabled. The router does not honor the precedence bits if queuing is not enabled, and the packets are sent in FIFO order. You can change the precedence setting by using either a number or a name.

The table below lists the possible IP Precedence values (numbers and their corresponding names), from the least important to the most important.

Table 2: IP Precedence Values

Number	Name
0	routine
1	priority
2	immediate

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

Fast-Switched Policy Routing

IP policy routing can 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. With fast-switched policy routing, users who need policy routing to occur at faster speeds can implement policy routing without slowing down the device.

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

- **set ip default**
- **set interface**

The **set interface** command is supported only over point-to-point links, unless there is a route cache entry that uses the same interface that is specified in the command in the route map.

To configure fast-switched policy routing, use the **ip route-cache policy** interface configuration command.

Local Policy Routing

Packets that are generated by the router are not normally policy-routed. To enable local policy routing for such packets, you must indicate which route map the router should use. All packets originating on the router will then be subject to local policy routing. To identify the route map to be used for local policy routing, use the **ip local policy route-map** 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 information monitoring on real-time traffic flows. IP policy routing works with Cisco Express Forwarding (formerly known as CEF), distributed Cisco Express Forwarding (formerly known as dCEF), and NetFlow.

NetFlow policy routing leverages the following technologies:

- Cisco Express Forwarding, 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.

- Distributed Cisco Express Forwarding, which addresses the scalability and maintenance problems of a demand caching scheme.
- NetFlow, which provides accounting, capacity planning, and traffic monitoring capabilities.

The following are the benefits of NPR:

- NPR takes advantage of new switching services. Cisco Express Forwarding, distributed Cisco Express Forwarding, and NetFlow can now use policy routing.
- Policy routing can be deployed on a wide scale and on high-speed interfaces.

NPR is the default policy routing mode. No additional configuration tasks are required to enable policy routing with Cisco Express Forwarding, distributed Cisco Express Forwarding, or NetFlow. As soon as one of these features is turned on, packets are automatically subjected to policy routing in the appropriate switching path.

The following example shows how to configure policy routing with Cisco Express Forwarding. The route is configured to verify that the next hop 10.0.0.8 of the route map named test is a Cisco Discovery Protocol neighbor before the device tries to policy-route to it.

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

Authentication Key Management and Supported Protocols

Key management is a method of controlling the authentication keys used by routing protocols. Not all protocols support key management. Authentication keys are available for Director Response Protocol (DRP) Agent, Enhanced Interior Gateway Routing Protocol (EIGRP), and Routing Information Protocol (RIP) Version 2.

You can manage authentication keys by defining key chains, identifying the keys that belong to the key chain, and specifying how long each key is valid. Each key has its own key identifier (specified using the **key chain** configuration command), which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and the message digest algorithm 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 ascending order and uses the first valid key it encounters. The lifetimes allow for overlap during key changes.

How to Configure Basic IP Routing

Redistributing Routing Information

You can redistribute routes from one routing domain into another, with or without controlling the redistribution with a route map. To control which routes are redistributed, configure a route map and reference the route map from the **redistribute** command.

The tasks in this section describe how to define the conditions for redistributing routes (a route map), how to redistribute routes, and how to remove options for redistributing routes, depending on the protocol being used.

Defining Conditions for Redistributing Routes

Route maps can be used to control route redistribution (or to implement policy-based routing). To define conditions for redistributing routes from one routing protocol into another, configure the **route-map** command. Then use at least one **match** command in route map configuration mode, as needed. At least one **match** command is used in this task because the purpose of the task is to illustrate how to define one or more conditions on which to base redistribution.

Note A route map is not required to have **match** commands; it can have only **set** commands. If there are no **match** commands, everything matches the route map.

Note There are many more **match** commands not shown in this table. For additional **match** commands, see the *Cisco IOS Master Command List*.

Command or Action	Purpose
match as-path *path-list-number*	Matches a BGP autonomous system path access list.
match community {*standard-list-number* | *expanded-list-number* | *community-list-name* **match community**[**exact**]}	Matches a BGP community.
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 routes that have a destination network address that is permitted to policy route packets or is permitted by a standard access list, an extended access list, or a prefix list.
match metric *metric-value*	Matches routes with the specified metric.
match ip next-hop {*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*]	Matches a next-hop device address passed by one of the specified access lists.
match tag *tag-value* [*tag-value*]	Matches the specified tag value.
match interface *type number* [*type number*]	Matches routes that use the specified interface as the next hop.
match ip route-source {*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*]	Matches the address specified by the advertised access lists.
match route-type {**local** | **internal** | **external** [**type-1** | **type-2**] | **level-1** | **level-2**}	Matches the specified route type.

To optionally specify the routing actions for the system to perform if the match criteria are met (for routes that are being redistributed by the route map), use one or more **set** commands in route map configuration mode, as needed.

Note A route map is not required to have **set** commands; it can have only **match** commands.

Note There are more **set** commands not shown in this table. For additional **set** commands, see the *Cisco IOS Master Command List*.

Command or Action	Purpose
set community {*community-number* [**additive**] [**well-known**] | **none**}	Sets the community attribute (for BGP).
set dampening *halflife reuse suppress max-suppress-time*	Sets route dampening parameters (for BGP).
set local-preference *number-value*	Assigns a local preference value to a path (for BGP).
set origin {**igp** | **egp** *as-number* | **incomplete**}	Sets the route origin code.
set as-path{**tag** | **prepend** *as-path-string* }	Modifies the autonomous system path (for BGP).
set next-hop *next-hop*	Specifies the address of the next hop.
set automatic-tag	Enables automatic computation of the tag table.
set level {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**}	Specifies the areas to import routes.
set metric *metric-value*	Sets the metric value for redistributed routes (for any protocol, except EIGRP).
set metric *bandwidth delay reliability load mtu*	Sets the metric value for redistributed routes (for EIGRP only).
set metric-type {**internal** | **external** | **type-1** | **type-2**}	Sets the metric type for redistributed routes.
set metric-type internal	Sets the Multi Exit Discriminator (MED) value on prefixes advertised to the external BGP neighbor to match the Interior Gateway Protocol (IGP) metric of the next hop.
set tag *tag-value*	Sets a tag value to be applied to redistributed routes.

Redistributing Routes from One Routing Domain to Another

Perform this task to redistribute routes from one routing domain into another and to control route redistribution. This task shows how to redistribute OSPF routes into a BGP domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system*
4. **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**]
5. **default-metric** *number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system* **Example:** `Device(config)# router bgp 109`	Enables a BGP routing process and enters router configuration mode.
Step 4	**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**] **Example:** `Device(config-router)# redistribute ospf 2 level-1`	Redistributes routes from the specified routing domain into another routing domain.
Step 5	**default-metric** *number* **Example:** `Device(config-router)# default-metric 10`	Sets the default metric value for redistributed routes. **Note** The metric value specified in the **redistribute** command supersedes the metric value specified using the **default-metric** command.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Removing Options for Redistribution Routes

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

Different protocols implement the **no redistribute** command differently as follows:

- In BGP, OSPF, and RIP configurations, the **no redistribute** command removes only the specified keywords from the **redistribute** commands in the running configuration. They use the *subtractive keyword* method when redistributing from other protocols. For example, in the case of BGP, if you configure **no redistribute static route-map interior**, only the route map is removed from the redistribution, leaving **redistribute static** in place with no filter.
- The **no redistribute isis** command removes the IS-IS redistribution from the running configuration. IS-IS removes the entire command, regardless of whether IS-IS is the redistributed or redistributing protocol.
- EIGRP used the subtractive keyword method prior to EIGRP component version rel5. Starting with EIGRP component version rel5, the **no redistribute** command removes the entire **redistribute** command when redistributing from any other protocol.
- For the **no redistribute connected** command, the behavior is subtractive if the **redistribute** command is configured under the **router bgp** or the **router ospf** command. The behavior is complete removal of the command if it is configured under the **router isis** or the **router eigrp** command.

The following OSPF commands illustrate how various options are removed from the redistribution in router configuration mode.

Command or Action	Purpose
no redistribute connected metric 1000 subnets	Removes the configured metric value of 1000 and the configured subnets and retains the **redistribute connected** command in the configuration.
no redistribute connected metric 1000	Removes the configured metric value of 1000 and retains the **redistribute connected subnets** command in the configuration.
no redistribute connected subnets	Removes the configured subnets and retains the **redistribute connected metric** *metric-value* command in the configuration.
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 this section.

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

Preventing Routing Updates Through an Interface

To prevent other routers on a local network from dynamically learning routes, you can keep routing update messages from being sent through a router interface. To prevent routing updates through a specified interface, use the **passive-interface** command in router configuration mode. This command is supported in all IP-based routing protocols, except BGP.

OSPF and IS-IS behave 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.

Configuring Default Passive Interfaces

Perform this task to set all interfaces on a device, in an Enhanced Interior Gateway Routing Protocol (EIGRP) environment, as passive by default, and then activate only those interfaces where adjacencies are desired.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** {*autonomous-system-number* | *virtual-instance-number*}
4. **passive-interface** [**default**] [*type number*]
5. **no passive-interface** [**default**] [*type number*]
6. **network** *network-address* [*options*]
7. **end**
8. **show ip eigrp interfaces**
9. **show ip interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** {*autonomous-system-number* | *virtual-instance-number*} **Example:** `Device(config)# router eigrp 1`	Configures an EIGRP process and enters router configuration mode. • *autonomous-system-number*—Autonomous system number that identifies the services to the other EIGRP address-family devices. It is also used to tag routing information. The range is 1 to 65535. • *virtual-instance-number*—EIGRP virtual instance name. This name must be unique among all address-family router processes on a single device, but need not be unique among devices

	Command or Action	Purpose
Step 4	**passive-interface** [**default**] [*type number*] **Example:** `Device(config-router)# passive-interface default`	Sets all interfaces as passive by default.
Step 5	**no passive-interface** [**default**] [*type number*] **Example:** `Device(config-router)# no passive-interface gigabitethernet 0/0/0`	Activates only those interfaces that need adjacencies.
Step 6	**network** *network-address* [*options*] **Example:** `Device(config-router)# network 192.0.2.0`	Specifies the list of networks to be advertised by routing protocols.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 8	**show ip eigrp interfaces** **Example:** `Device# show ip eigrp interfaces`	Verifies whether interfaces on your network have been set to passive.
Step 9	**show ip interface** **Example:** `Device# show ip interface`	Verifies whether interfaces you enabled are active.

Controlling the Advertising of Routes in Routing Updates

To prevent other devices from learning one or more routes, you can suppress routes from being advertised in routing updates. 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.

You cannot specify an interface name in Open Shortest Path First (OSPF). When used for OSPF, this feature applies only to external routes.

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 Routing Default Next-Hop Routes

Perform this task to configure the precedence of packets and specify where packets that pass the match criteria are output.

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 first use policy routing and then use the routing table. Configuring the **set ip default next-hop** command causes the system to first use the routing table and then the policy-route-specified next hop.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *route-map-name* [**permit** | **deny**] [*sequence-number*]]
4. **set ip precedence** {*number* | *name*}
5. **set ip next-hop** *ip-address* [*ip-address*]
6. **set interface** *type number* [*...type number*]
7. **set ip default next-hop** *ip-address* [*ip-address*]
8. **set default interface** *type number* [*...type number*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *route-map-name* [**permit** | **deny**] [*sequence-number*]] **Example:** `Device(config)# route-map rm1`	Defines a route map to control redistribution and enters route-map configuration mode.
Step 4	**set ip precedence** {*number* | *name*} **Example:** `Device(config-route-map)# set ip precedence 5`	Sets the precedence value in the IP header. **Note** You can specify either a precedence number or a precedence name.
Step 5	**set ip next-hop** *ip-address* [*ip-address*] **Example:** `Device(config-route-map)# set ip next-hop 192.0.2.1`	Specifies the next hop for routing packets. **Note** The next hop must be an adjacent device.
Step 6	**set interface** *type number* [...*type number*] **Example:** `Device(config-route-map)# set interface gigabitethernet 0/0/0`	Specifies the output interface for the packet.
Step 7	**set ip default next-hop** *ip-address* [*ip-address*] **Example:** `Device(config-route-map)# set ip default next-hop 172.16.6.6`	Specifies the next hop for routing packets 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 device.
Step 8	**set default interface** *type number* [...*type number*] **Example:** `Device(config-route-map)# set default interface serial 0/0/0`	Specifies the output interface for the packet if there is no explicit route for the destination.
Step 9	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Configuring QoS Policy Propagation via BGP

Configuring QoS Policy Propagation via BGP Based on Community Lists

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**route-map** *route-map-name* [**permit** | **deny** [*sequence-number*]] **Example:** `Device(config)# route-map rm1`	Defines a route map to control redistribution and enters route-map configuration mode.
Step 4	**match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]} **Example:** `Device(config-route-map)# match community 1`	Matches a Border Gateway Protocol (BGP) community list.
Step 5	**set ip precedence** [*number* | *name*] **Example:** `Device(config-route-map)# set ip precedence 5`	Sets the IP Precedence field when the community list matches. **Note** You can specify either a precedence number or a precedence name.
Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 7	**router bgp** *autonomous-system* **Example:** `Device(config)# router bgp 45000`	Enables a BGP process and enters router configuration mode.
Step 8	**table-map** *route-map-name* **Example:** `Device(config-router)# table-map rm1`	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 9	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 10	**ip community-list** *standard-list-number* {**permit** | **deny**} [*community-number*] **Example:** `Device(config)# ip community-list 1 permit 2`	Creates a community list for BGP and controls access to it.

	Command or Action	Purpose
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the interface (or subinterface) and enters interface configuration mode.
Step 12	**bgp-policy** {**source** | **destination**} **ip-prec-map** **Example:** `Device(config-if)# bgp-policy source ip-prec-map`	Classifies packets using IP precedence.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 14	**ip bgp-community new-format** **Example:** `Device(config)# ip bgp-community new-format`	(Optional) Displays the BGP community number in AA:NN (autonomous system:community number/4-byte number) format.
Step 15	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring QoS Policy Propagation via BGP Based on the Autonomous System Path Attribute

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *route-map-name* [**permit** | **deny** [*sequence-number*]] **Example:** `Device(config)# route-map rm1`	Defines a route map to control redistribution and enters route-map configuration mode.
Step 4	**match as-path** *path-list-number* **Example:** `Device(config-route-map)# match as-path 2`	Matches a Border Gateway Protocol (BGP) autonomous system path access list.

	Command or Action	Purpose
Step 5	**set ip precedence** [*number* | *name*] **Example:** `Device(config-route-map)# set ip precedence 5`	Sets the IP Precedence field when the autonomous-system path matches. **Note** You can specify either a precedence number or a precedence name.
Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 7	**router bgp** *autonomous-system* **Example:** `Device(config)# router bgp 45000`	Enables a BGP process and enters router configuration mode.
Step 8	**table-map** *route-map-name* **Example:** `Device(config-router)# table-map rm1`	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 9	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 10	**ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expression* **Example:** `Device(config)# ip as-path access-list 500 permit 45000`	Defines an autonomous system path access list.
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the interface (or subinterface) and enters interface configuration mode.
Step 12	**bgp-policy** {**source** | **destination**} **ip-prec-map** **Example:** `Device(config-if)# bgp-policy source ip-prec-map`	Classifies packets using IP precedence.
Step 13	**end** **Example:** `Device(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 the QoS Policy Propagation via BGP feature based on an access list. This section assumes that you have already configured Cisco Express Forwarding or distributed Cisco Express Forwarding and BGP on your router.

Perform this task to configure the router to propagate the IP precedence based on an access list:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *route-map-name* [**permit** | **deny** [*sequence-number*]]
4. **match ip address** *access-list-number*
5. **set ip precedence** [*number* | *name*]
6. **exit**
7. **router bgp** *autonomous-system*
8. **table-map** *route-map-name*
9. **exit**
10. **access-list** *access-list-number* {**permit** | **deny**} *source*
11. **interface** *type number*
12. **bgp-policy** {**source** | **destination**} **ip-prec-map**
13. **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	**route-map** *route-map-name* [**permit** | **deny** [*sequence-number*]] **Example:** `Router(config)# route-map rm1`	Defines a route map to control redistribution and enters route-map configuration mode.

	Command or Action	Purpose
Step 4	**match ip address** *access-list-number* **Example:** `Router(config-route-map)# match ip address 3`	Matches routes that have a destination network address that is permitted by a standard or extended access list.
Step 5	**set ip precedence** [*number* | *name*] **Example:** `Router(config-route-map)# set ip precedence 5`	Sets the IP Precedence field when the autonomous system path matches. **Note** You can specify either a precedence number or a precedence name.
Step 6	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 7	**router bgp** *autonomous-system* **Example:** `Router(config)# router bgp 45000`	Enables a BGP routing process and enters router configuration mode.
Step 8	**table-map** *route-map-name* **Example:** `Router(config-router)# table-map rm1`	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 9	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 10	**access-list** *access-list-number* {**permit** | **deny**} *source* **Example:** `Router(config)# access-list 2 permit 172.16.0.2`	Defines an access list.
Step 11	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/0`	Specifies the interface (or subinterface) and enters interface configuration mode.
Step 12	**bgp-policy** {**source** | **destination**} **ip-prec-map** **Example:** `Router(config-if)# bgp-policy source ip-prec-map`	Classifies packets using IP precedence.
Step 13	**end** **Example:** `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 the BGP feature configuration, use the following optional commands.

Command or Action	Purpose
show ip bgp	Displays entries in the Border Gateway Protocol (BGP) routing table to verify whether the correct community is set on the prefixes.
show ip bgp community-list *community-list-number*	Displays routes permitted by the BGP community to verify whether correct prefixes are selected.
show ip cef *network*	Displays entries in the forwarding information base (FIB) table based on the specified IP address to verify whether Cisco Express Forwarding has the correct precedence value for the prefix.
show ip interface	Displays information about the interface.
show ip route *prefix*	Displays the current status of the routing table to verify whether correct precedence values are set on the prefixes.

Managing Authentication Keys

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** 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 ascending order and uses the first valid key it encounters. The lifetimes allow for overlap during key changes. `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain chain1`	Defines a key chain and enters key-chain configuration mode.
Step 4	**key** *number* **Example:** `Device(config-keychain)# key key1`	Identifies the key number in key-chain configuration mode and enters key-chain key configuration mode.

	Command or Action	Purpose
Step 5	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string string1`	Identifies the key string.
Step 6	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# accept-lifetime 13:30:00 Dec 22 2011 duration 7200`	Specifies the time period during which the key can be received.
Step 7	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# send-lifetime 14:30:00 Dec 22 2011 duration 3600`	Specifies the time period during which the key can be sent.
Step 8	**end** **Example:** `Device(config-keychain-key)# end`	Exits key-chain key configuration mode and returns to privileged EXEC mode.
Step 9	**show key chain** **Example:** `Device# show key chain`	(Optional) Displays authentication key information.

Monitoring and Maintaining the IP Network

Clearing Routes from the IP Routing Table

You can remove all contents of a particular table. Clearing a table may 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 privileged EXEC mode.

Displaying System and Network Statistics

You can use the following **show** commands to display system and network statistics. You can display specific statistics such as contents of IP routing tables, caches, and databases. You can also display information about node reachability and discover the routing path that packets leaving your device are taking through the network. This information can an be used to determine resource utilization and solve network problems.

Command or Action	Purpose
show ip cache policy	Displays cache entries in the policy route cache.
show ip local policy	Displays the local policy route map if one exists.
show ip policy	Displays policy route maps.
show ip protocols	Displays the parameters and current state of the active routing protocols.
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.
show ip route summary	Displays the current state of the routing table in summary form.
show ip route supernets-only	Displays supernets.
show key chain [*name-of-chain*]	Displays authentication key information.
show route-map [*map-name*]	Displays all route maps configured or only the one specified.

Configuration Examples for Basic IP Routing

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
```

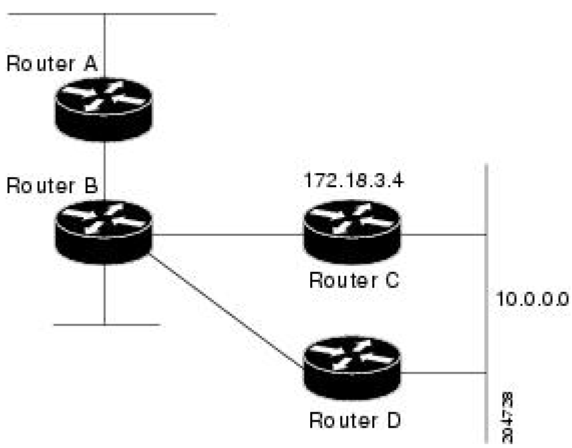
```
Router(config-if)# ip address 172.16.1.1 255.255.255.0

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 specifies EIGRP routing on networks 192.0.2.16 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 192.0.2.16
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 192.0.2.1 an administrative distance of 100 and all other routers on subnet 192.0.2.0 an administrative distance of 200:

```
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
```

However, if you reverse the order of these two commands, all routers on subnet 192.0.2.0 are assigned an administrative distance of 200, including the router at address 192.0.2.1:

```
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
Router(config-router)# distance 100 192.0.2.1 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 learned IP routes 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 the software services. However, you can transfer routing information among routing databases.

In the following example, 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.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: Mutual Redistribution Between EIGRP and RIP

Consider a WAN at a university that uses the Routing Information Protocol (RIP) as an interior routing protocol. Assume that the university wants to connect its WAN to regional network 172.16.0.0, which uses the Enhanced Interior Gateway Routing Protocol (EIGRP) as the routing protocol. The goal in this case is to advertise the networks in the university network to devices in the regional network.

Mutual redistribution is configured between EIGRP and RIP in the following example:

```
Device(config)# access-list 10 permit 172.16.0.0
Device(config)# !
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.0
Device(config-router)# redistribute rip metric 10000 100 255 1 1500
Device(config-router)# default-metric 10
Device(config-router)# distribute-list 10 out rip
Device(config-router)# exit
Device(config)# router rip
Device(config-router)# redistribute eigrp 1
Device(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 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 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: Mutual Redistribution Between EIGRP and BGP

In the following example, mutual redistribution is configured between the Enhanced Interior Gateway Routing Protocol (EIGRP) and the Border Gateway Protocol (BGP).

Routes from EIGRP routing process 101 are injected into BGP autonomous system 50000. A filter is configured to ensure that the correct routes are advertised, in this case, three networks. Routes from BGP autonomous system 50000 are injected into EIGRP routing process 101. The same filter is used.

```
Device(config)# ! All networks that should be advertised from R1 are controlled with ACLs:

Device(config)# access-list 1 permit 172.18.0.0 0.0.255.255
Device(config)# access-list 1 permit 172.16.0.0 0.0.255.255
Device(config)# access-list 1 permit 172.25.0.0 0.0.255.255
Device(config)# ! Configuration for router R1:
Device(config)# router bgp 50000
Device(config-router)# network 172.18.0.0
Device(config-router)# network 172.16.0.0
Device(config-router)# neighbor 192.168.10.1 remote-as 2
Device(config-router)# neighbor 192.168.10.15 remote-as 1
Device(config-router)# neighbor 192.168.10.24 remote-as 3
Device(config-router)# redistribute eigrp 101
Device(config-router)# distribute-list 1 out eigrp 101
Device(config-router)# exit
Device(config)# router eigrp 101
Device(config-router)# network 172.25.0.0
Device(config-router)# redistribute bgp 50000
Device(config-router)# distribute-list 1 out bgp 50000
Device(config-router)# !
```

Caution BGP should be redistributed into an Interior Gateway Protocol (IGP) when there are no other suitable options. Redistribution from BGP into any IGP should be applied with proper filtering by using distribute lists, IP prefix lists, and route map statements to limit the number of prefixes.

Examples: OSPF Routing and Route Redistribution

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

This section provides the following configuration examples:

- The first example shows simple configurations illustrating basic OSPF commands.
- The second example shows configurations for an internal device, ABR, and ASBR within a single, arbitrarily assigned OSPF autonomous system.
- The third example illustrates a more complex configuration and the application of various tools available for controlling OSPF-based routing environments.

Example: Basic OSPF Configurations

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

```
Router(config)# interface Ethernet 0/0
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 1/0
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 shows 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 0/0 is in area 10.9.50.0:
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.18.20.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 1/0 is in area 2:
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 172.18.1.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 2/0 is in area 2:
Router(config)# interface Ethernet 2/0
Router(config-if)# ip address 172.18.2.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 3/0 is in area 3:
Router(config)# interface Ethernet 3/0
Router(config-if)# ip address 172.19.10.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 4/0 is in area 0:
Router(config)# interface Ethernet 4/0
Router(config-if)# ip address 172.19.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 5/0 is in area 0:
Router(config)# interface Ethernet 5/0
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 *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 0/0. Ethernet interface 0/0 is attached to Area 10.9.50.0 only.

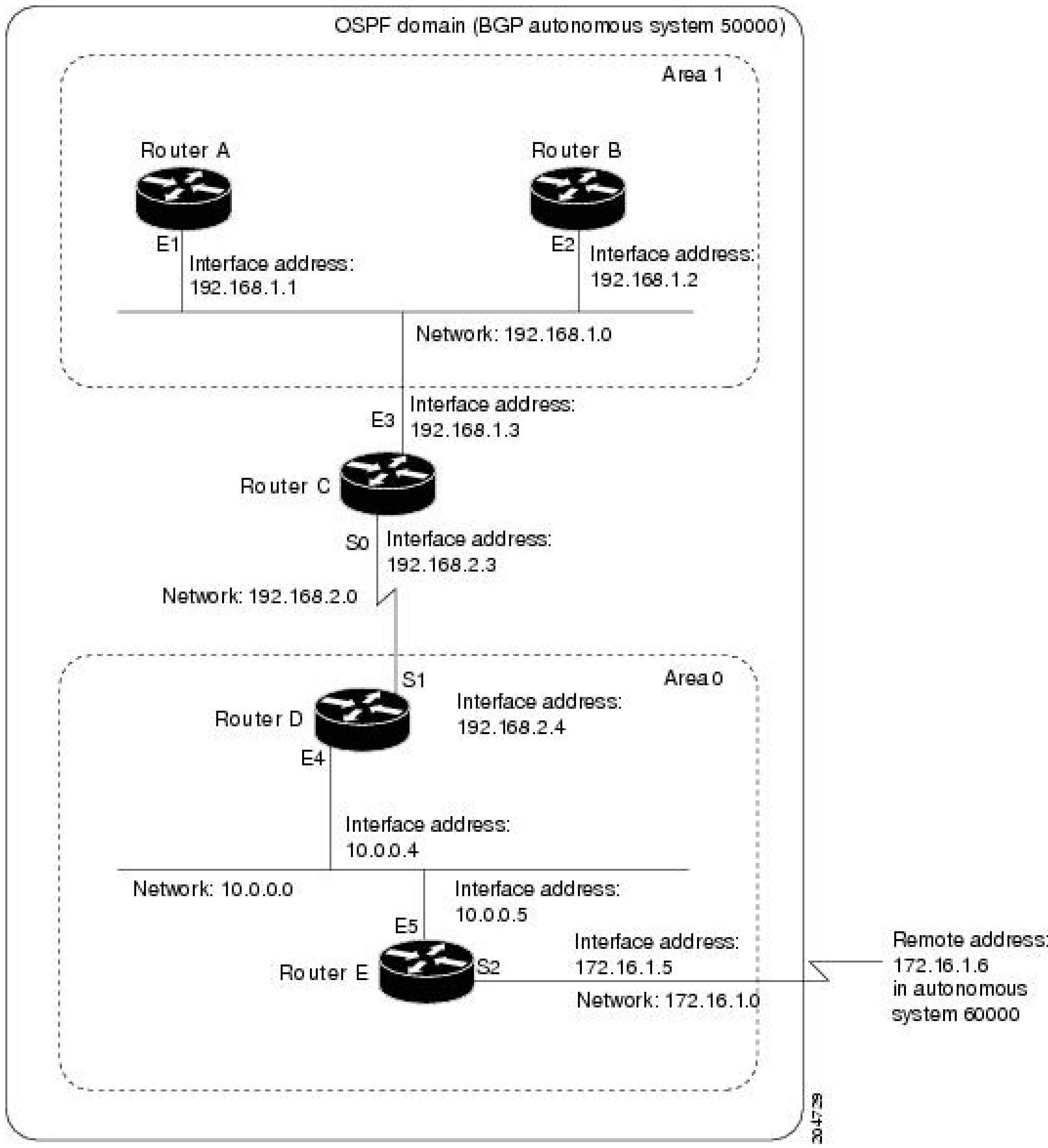
The second **network** command is evaluated next. For Area 2, all interfaces (except Ethernet interface 0/0) are evaluated. Assume that a match is determined for Ethernet interface 1/0. OSPF is then enabled for that interface, and Ethernet 1/0 is attached to Area 2.

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 ASBR Configurations

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, the following five routers are configured in OSPF autonomous system 1:

- Router A and Router B are both internal routers within area 1.
- Router C is an OSPF ABR. Note that for Router C, area 1 is assigned to E3 and Area 0 is assigned to S0.

- Router D is an internal router in Area 0 (backbone area). In this case, both **network** router configuration commands specify the same area (Area 0 or the backbone area).
- Router E is an OSPF ASBR. Note that BGP routes are redistributed into OSPF and that these routes are advertised by OSPF.

Note You don't have 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 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.

Here is an example configuration for the general network map shown in the figure above.

Router A Configuration—Internal Router

```
Router(config)# interface Ethernet 1/0
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 2/0
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 3/0
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 4/0
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 5/0
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
```

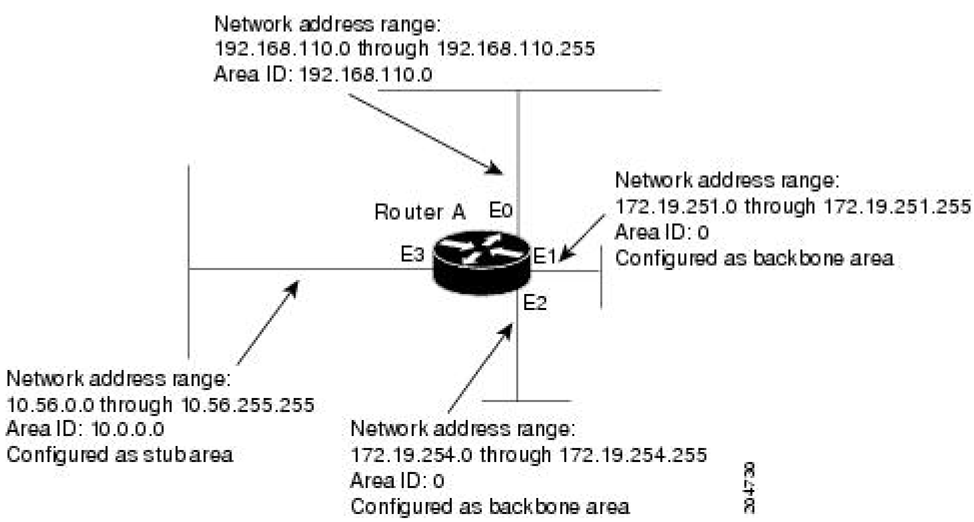
Example: Complex OSPF Configuration

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

- Basic OSPF configuration
- Route redistribution

The figure below illustrates the network address ranges and area assignments for interfaces.

Figure 3: Interface and Area Specifications for the OSPF Configuration



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 route 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 a sample OSPF configuration:

```
Router(config)# interface Ethernet 0/0
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 1/0
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 2/0
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 3/0
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 how a router in autonomous system 1 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 below. The following example shows how to redistribute all OSPF routes into EIGRP:

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

The following example shows how to redistribute 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-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 shows how to redistribute 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 shows how to redistribute OSPF intra-area and inter-area routes with next-hop routers on serial interface 0/0 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 called conditional default origination. OSPF will originate the default route (network 0.0.0.0) with a type 2 metric of 5 if 172.20.0.0 is in the routing table.

```
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
```

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 0/0
Router(config-if)# ip address 172.18.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 172.18.2.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Ethernet 2/0
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: Configuring Default Passive Interfaces

The following example shows how to configure network interfaces, set all interfaces that are running OSPF as passive, and then enable serial interface 0/0:

```
Router(config)# interface Ethernet 0/0
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 0/0
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 0/0
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 1/0/0 from the source 10.1.1.1 are sent to the device at 172.16.6.6 if the device has no explicit route for the destination of the packet. Packets that arrive from the source 172.17.2.2 are sent to the device at 192.168.7.7 if the device has no explicit route for the destination of the packet. All other packets for which the device has no explicit route to the destination are discarded.

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

Example: Policy Routing with Cisco Express Forwarding

The following example shows how to configure policy routing with Cisco Express Forwarding. The route is configured to verify that the next hop 10.0.0.8 of the route map named test is a Cisco Discovery Protocol neighbor before the device tries to policy-route to it.

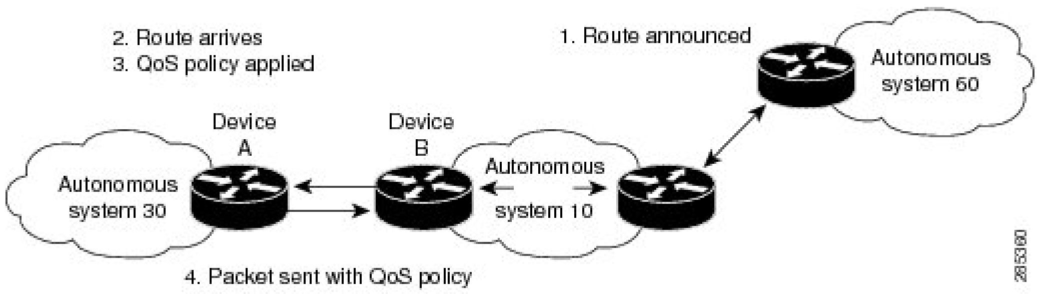
```
Device(config)# ip cef
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip route-cache flow
Device(config-if)# ip policy route-map test
Device(config-if)# exit
Device(config)# route-map test permit 10
Device(config-route-map)# match ip address 1
Device(config-route-map)# set ip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

Example: Configuring QoS Policy Propagation via BGP

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

In the figure below, Device A learns routes from autonomous system 10 and autonomous system 60. The quality of service (QoS) policy is applied to all packets that match defined route maps. Any packets from Device A to autonomous system 10 or autonomous system 60 are sent the appropriate QoS policy, as the numbered steps in the figure indicate.

Figure 4: Device Learning Routes and Applying QoS Policy



Device A 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 autonomous system 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 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
```

Device 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 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
```

Example: Managing Authentication Keys

The following example shows how to configure a key chain named kc1. In this example, the software will always accept and send ks1 as a valid key. The key ks2 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.

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip rip authentication key-chain kc1
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 kc1
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string ks1
Router(config-keychain-key)# key 2
Router(config-keychain-key)# key-string ks2
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 ks3
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
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping
IP Routing Protocol-Independent Features	*IP Routing Protocol-Independent Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	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

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 ISPs and large enterprise networks, many distribution routers have more than 200 interfaces. Obtaining routing information from these interfaces requires configuration of the routing protocol on all interfaces and manual configuration of the **passive-interface** command on interfaces where adjacencies were not desired. The Default Passive Interface feature simplifies the configuration of distribution routers by allowing all interfaces to be set as passive by default. The following commands were introduced or modified: **no passive-interface**, **passive-interface default**.
Fast-Switched Policy Routing	11.3	IP policy routing can 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 who need policy routing to occur at faster speeds can implement policy routing without slowing down the router.

Feature Name	Releases	Feature Information
IP Routing	11.0	The IP Routing feature introduces basic IP routing features.
NetFlow Policy Routing	12.0(3)T 15.3(1)S	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, distributed Cisco Express Forwarding, and NetFlow.
Policy-Based Routing	11.0 15.3(1)S	The Policy-Based Routing feature introduces a more flexible mechanism than destination routing for routing packets. Policy-based routing (PBR) 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 Default Next-Hop Route	12.1(11)E 15.3(1)S	The Policy-Based Routing 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. The following command was modified by this feature: **set ip default next-hop**.
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 and NetFlow. When both policy routing and NetFlow are enabled, redundant processing is avoided.

Feature Name	Releases	Feature Information
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.

CHAPTER 2

IPv6 Routing: Static Routing

This feature provides static routing for IPv6. Static routes are manually configured and define an explicit path between two networking devices.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

Prerequisites for IPv6 Routing: Static Routing

Before configuring the device with a static IPv6 route, you must enable the forwarding of IPv6 packets using the **ipv6 unicast-routing** global configuration command, enable IPv6 on at least one interface, and configure an IPv6 address on that interface.

Restrictions for IPv6 Routing: Static Routing

You should not configure static configurations over dynamic interfaces, because static configurations will be lost during reboot or when the user disconnects and reconnects the device.

Information About IPv6 Routing: Static Routing

Static Routes

Networking devices forward packets using route information that is either manually configured or dynamically learned using a routing protocol. Static routes are manually configured and define an explicit path between two networking devices. Unlike a dynamic routing protocol, static routes are not automatically updated and must be manually reconfigured if the network topology changes. The benefits of using static routes include security and resource efficiency. Static routes use less bandwidth than dynamic routing protocols and no CPU cycles are used to calculate and communicate routes. The main disadvantage to using static routes is the lack of automatic reconfiguration if the network topology changes.

Static routes can be redistributed into dynamic routing protocols but routes generated by dynamic routing protocols cannot be redistributed into the static routing table. No algorithm exists to prevent the configuration of routing loops that use static routes.

Static routes are useful for smaller networks with only one path to an outside network and to provide security for a larger network for certain types of traffic or links to other networks that need more control. In general, most networks use dynamic routing protocols to communicate between networking devices but may have one or two static routes configured for special cases.

Directly Attached Static Routes

In directly attached static routes, only the output interface is specified. The destination is assumed to be directly attached to this interface, so the packet destination is used as the next-hop address. This example shows such a definition:

```
ipv6 route 2001:DB8::/32 ethernet1/0
```

The example specifies that all destinations with address prefix 2001:DB8::/32 are directly reachable through interface Ethernet1/0.

Directly attached static routes are candidates for insertion in the IPv6 routing table only if they refer to a valid IPv6 interface; that is, an interface that is both up and has IPv6 enabled on it.

Recursive Static Routes

In a recursive static route, only the next hop is specified. The output interface is derived from the next hop. This definition is shown in the following example:

```
ipv6 route 2001:DB8::/32 2001:DB8:3000:1
```

This example specifies that all destinations with address prefix 2001:DB8::/32 are reachable via the host with address 2001:DB8:3000:1.

A recursive static route is valid (that is, it is a candidate for insertion in the IPv6 routing table) only when the specified next hop resolves, either directly or indirectly, to a valid IPv6 output interface, provided the route does not self-recurse, and the recursion depth does not exceed the maximum IPv6 forwarding recursion depth.

A route self-recurses if it is itself used to resolve its own next hop. For example, suppose we have the following routes in the IPv6 routing table:

```
IPv6 Routing Table - 9 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
R   2001:DB8::/32 [130/0]
     via ::, Serial2/0
B   2001:DB8:3000:0/16 [200/45]
     Via 2001:DB8::0104
```

The following examples defines a recursive IPv6 static route:

```
ipv6 route
2001:DB8::/32 2001:0BD8:3000:1
```

This static route will not be inserted into the IPv6 routing table because it is self-recursive. The next hop of the static route, 2001:DB8:3000:1, resolves via the BGP route 2001:DB8:3000:0/16, which is itself a recursive route (that is, it only specifies a next hop). The next hop of the BGP route, 2001:DB8::0104, resolves via the static route. Therefore, the static route would be used to resolve its own next hop.

It is not normally useful to manually configure a self-recursive static route, although it is not prohibited. However, a recursive static route that has been inserted in the IPv6 routing table may become self-recursive as a result of some transient change in the network learned through a dynamic routing protocol. If this occurs, the fact that the static route has become self-recursive will be detected and it will be removed from the IPv6 routing table, although not from the configuration. A subsequent network change may cause the static route to no longer be self-recursive, in which case it will be reinserted in the IPv6 routing table.

Fully Specified Static Routes

In a fully specified static route, both the output interface and the next hop are specified. This form of static route is used when the output interface is a multi-access one and it is necessary to explicitly identify the next hop. The next hop must be directly attached to the specified output interface. The following example shows a definition of a fully specified static route:

```
ipv6 route 2001:DB8:/32 ethernet1/0 2001:DB8:3000:1
```

A fully specified route is valid (that is, a candidate for insertion into the IPv6 routing table) when the specified IPv6 interface is IPv6-enabled and up.

Floating Static Routes

Floating static routes are static routes that are used to back up dynamic routes learned through configured routing protocols. A floating static route is configured with a higher administrative distance than the dynamic routing protocol it is backing up. As a result, the dynamic route learned through the routing protocol is always used in preference to the floating static route. If the dynamic route learned through the routing protocol is lost, the floating static route will be used in its place. The following example defines a floating static route:

```
ipv6 route 2001:DB8:/32 ethernet1/0 2001:DB8:3000:1 210
```

Any of the three types of IPv6 static routes can be used as a floating static route. A floating static route must be configured with an administrative distance that is greater than the administrative distance of the dynamic routing protocol, because routes with smaller administrative distances are preferred.

Note By default, static routes have smaller administrative distances than dynamic routes, so static routes will be used in preference to dynamic routes.

How to Configure IPv6 Static Routing

Configuring a Static IPv6 Route

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *ipv6-prefix* / *prefix-length* *ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast**| **multicast**] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** *ipv6-prefix* / *prefix-length* *ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast**| **multicast**] [**tag** *tag*] **Example:** `Device(config)# ipv6 route ::/0 serial 2/0`	Configures a static IPv6 route. • A static default IPv6 route is being configured on a serial interface. • See the syntax examples that immediately follow this table for specific uses of the **ipv6 route** command for configuring static routes.

Configuring a Recursive IPv6 Static Route to Use a Default IPv6 Static Route

By default, a recursive IPv6 static route will not resolve using the default route (::/0). Perform this task to restore legacy behavior and allow resolution using the default route.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route static resolve default**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route static resolve default** **Example:** `Device(config)# ipv6 route static resolve default`	Allows a recursive IPv6 static route to resolve using the default IPv6 static route.

Configuring a Floating Static IPv6 Route

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *ipv6-prefix* **/** *prefix-length* {*ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** *ipv6-prefix* **/** *prefix-length* {*ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [**tag** *tag*] **Example:** `Device(config)# ipv6 route 2001:DB8::/32 serial 2/0 201`	Configures a static IPv6 route. • In this example, a floating static IPv6 route is being configured. • Default administrative distances are as follows: • Connected interface--0 • Static route--1 • Enhanced Interior Gateway Routing Protocol (EIGRP) summary route--5 • External Border Gateway Protocol (eBGP)--20 • Internal Enhanced IGRP--90 • IGRP--100 • Open Shortest Path First--110 • Intermediate System-to-Intermediate System (IS-IS)--115 • Routing Information Protocol (RIP)--120 • Exterior Gateway Protocol (EGP)--140 • EIGRP external route--170 • Internal BGP--200 • Unknown--255

Verifying Static IPv6 Route Configuration and Operation

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **show ipv6 static** [*ipv6-address* | *ipv6-prefix* / *prefix-length*][**interface** *interface-type interface-number*] [**recursive**] [**detail**]
   - **show ipv6 route** [*ipv6-address* | *ipv6-prefix* / *prefix-length* | *protocol* | *interface-type interface-number*]
3. **debug ipv6 routing**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	Do one of the following:
	
	• **show ipv6 static** [*ipv6-address* | *ipv6-prefix* / *prefix-length*][**interface** *interface-type interface-number*] [**recursive**] [**detail**]
	
	• **show ipv6 route** [*ipv6-address* | *ipv6-prefix* / *prefix-length* | *protocol* | *interface-type interface-number*]
	
	Example:
	
	`Device# show ipv6 static`
	
	Example:
	
	`Device# show ipv6 route static`
		Displays the current contents of the IPv6 routing table.
		
		• These examples show two different ways of displaying IPv6 static routes.

Step 3	**debug ipv6 routing** **Example:** `Device# debug ipv6 routing`	Displays debugging messages for IPv6 routing table updates and route cache updates.

Configuration Examples for IPv6 Static Routing

Static routes may be used for a variety of purposes. Common usages include the following:

- Manual summarization
- Traffic discard
- Fixed default route
- Backup route

In many cases, alternative mechanisms exist within Cisco software to achieve the same objective. Whether to use static routes or one of the alternative mechanisms depends on local circumstances.

Example: Configuring Manual Summarization

The following example shows a static route being used to summarize local interface prefixes advertised into RIP. The static route also serves as a discard route, discarding any packets received by the device to a 2001:DB8:1::/48 destination not covered by a more specific interface prefix.

```
Device> enable
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface ethernet0/0
Device(config-if)# ipv6 address 2001:DB8:2:1234/64
Device(config-if)# exit
Device(config)#
Device(config)# interface ethernet1/0
Device(config-if)# ipv6 address 2001:DB8:3:1234/64
Device(config-if)# exit
Device(config)# interface ethernet2/0
Device(config-if)# ipv6 address 2001:DB8:4:1234/64
Device(config-if)# exit
Device(config)# interface ethernet3/0
Device(config-if)# ipv6 address 2001:DB8::1234/64
Device(config-if)# ipv6 rip one enable
Device(config-if)# exit
Device(config)# ipv6 Device rip one
Device(config-rtr)# redistribute static
Device(config-rtr)# exit
Device(config)# ipv6 route 2001:DB8:1:1/48 null0
Device(config)# end
00:01:30: %SYS-5-CONFIG_I: Configured from console by console
Device# show ipv6 route static

IPv6 Routing Table - 3 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   2001:DB8:1::/48 [1/0]
     via ::, Null0
```

Example: Configuring Traffic Discard

Configuring a static route to point at interface null0 may be used for discarding traffic to a particular prefix. For example, if it is required to discard all traffic to prefix 2001:DB8:42:1/64, the following static route would be defined:

```
Device> enable
Device# configure
 terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ipv6 route 2001:DB8:42:1::/64 null0
Device(config)# end
```

Example: Configuring a Fixed Default Route

A default static route is often used in simple device topologies. In the following example, a device is connected to its local site via Ethernet0/0 and to the main corporate network via Serial2/0 and Serial3/0. All nonlocal traffic will be routed over the two serial interfaces.

```
Device(config)# interface ethernet0/0
Device(config-if)# ipv6 address 2001:DB8:17:1234/64
Device(config-if)# exit
Device(config)# interface Serial2/0
Device(config-if)# ipv6 address 2001:DB8:1:1234/64
Device(config-if)# exit
Device(config)# interface Serial3/0
Device(config-if)# ipv6 address 2001:DB8:2:124/64
Device(config-if)# exit
Device(config)# ipv6 route ::/0 Serial2/0
Device(config)# ipv6 route ::/0 Serial3/0
Device(config)# end
Device#
00:06:30: %SYS-5-CONFIG_I: Configured from console by console
Device# show ipv6 route static
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   ::/0 [1/0]
     via ::, Serial2/0
     via ::, Serial3/0
```

Example: Configuring a Floating Static Route

A floating static route often is used to provide a backup path in the event of connectivity failure. In the following example, the device has connectivity to the network core via Serial2/0 and learns the route 2001:DB8:1:1/32 via IS-IS. If the Serial2/0 interface fails, or if route 2001:DB8:1:1/32 is no longer learned via IS-IS (indicating loss of connectivity elsewhere in the network), traffic is routed via the backup ISDN interface.

```
Device> enable
Device# configure
 terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface ethernet0/0
```

```
Device(config-if)# ipv6 address 2001:DB8:17:1234/64
Device(config-if)# exit
Device(config)# interface Serial2/0
Device(config-if)# ipv6 address 2001:DB8:1:1234/64
Device(config-if)# ipv6 router isis
Device(config-if)# exit
Device(config)# router isis
Device(config-rtr)# net 42.0000.0000.0000.0001.00
Device(config-rtr)# exit
Device(config)# interface BRI1/0
Device(config-if)# encapsulation ppp
Device(config-if)# ipv6 enable
Device(config-if)# isdn switch-type basic-net3
Device(config-if)# ppp authentication chap optional
Device(config-if)# ppp multilink
Device(config-if)# exit
Device(config)# dialer-list 1 protocol ipv6 permit
Device(config)# ipv6 route 2001:DB8:1::/32 BRI1/0 200
Device(config)# end
Device#
00:03:07: %SYS-5-CONFIG_I: Configured from console by console
2001:DB8:5000:)/16, interface Ethernet3/0, distance 1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping
IP Routing Protocol-Independent Features	*IP Routing Protocol-Independent Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	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

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 IPv6 Routing: Static Routing

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 IPv6 Routing: Static Routing

Feature Name	Releases	Feature Information
IPv6 Routing: Static Routing	12.0(22)S 12.2(2)T 12.2(14)S 12.2(17a)SX1 12.2(25)SG 12.2(28)SB 12.2(33)SRA Cisco IOS XE Release 2.1	Static routes are manually configured and define an explicit path between two networking devices. The following commands were introduced or modified: **ipv6 route**, **ipv6 route static resolve default**, **show ipv6 route**, **show ipv6 static**.

CHAPTER 3

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 caveats and feature information, see Bug Search Tool and 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 module.

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.

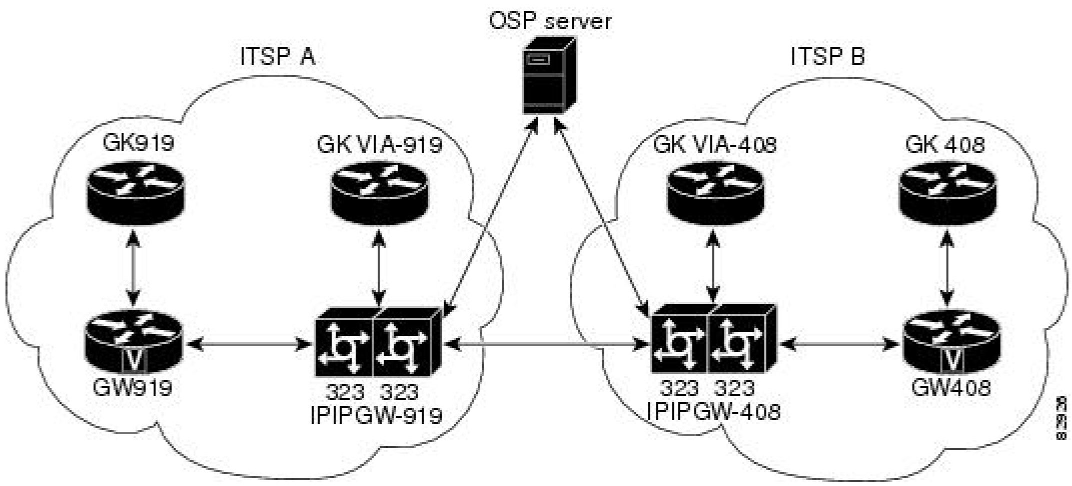
Reuse Threshold

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.
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.
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 5: 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.

CHAPTER 4

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 caveats and feature information, see Bug Search Tool and 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 module.

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

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.

Before You Begin

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.
Step 10	**interface** *type* *number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface type and number and enters interface configuration mode.

	Command or Action	Purpose
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.

Configuring PBR Support for Multiple Tracking Options

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.

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla monitor** *operation-number* **Example:** `Device(config)# ip sla monitor 1`	Starts a Cisco IOS IP Service Level Agreement (SLA) operation configuration and enters IP SLA monitor configuration mode.

	Command or Action	Purpose
Step 4	**type echo protocol ipIcmpEcho** {*destination-ip-address*| *destination-hostname*}[**source-ipaddr** {*ip-address*| *hostname*} | **source-interface** *interface-name*] **Example:** `Device(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:** `Device(config-sla-monitor)# exit`	Exits IP SLA monitor configuration mode and returns the device 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:** `Device(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:** `Device(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:** `Device(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:** `Device(config-track)# exit`	Exits tracking configuration mode and returns the device to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface serial 2/0`	Specifies an interface type and number and enters interface configuration mode.

	Command or Action	Purpose
Step 11	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(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:** `Device(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:** `Device(config-if)# exit`	Exits interface configuration mode and returns the device to global configuration mode.
Step 14	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(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:** `Device(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.
Step 16	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and returns the device to privileged EXEC mode.
Step 17	**show track** *object-number* **Example:** `Device# 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**]	(Optional) Displays route map information.

	Command or Action	Purpose
	Example: `Device# show route-map alpha`	• 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.

```
Device# 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.

```
Device# 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
```

Example: Configuring PBR Support for Multiple Tracking Options

The following example shows how to configure PBR support for multiple tracking options.

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
Cisco IOS commands	Cisco IOS Master Command List, All Releases
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*

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 6: Feature Information for PBR Support for Multiple Tracking Options

Feature Name	Releases	Feature Information
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. The following commands were introduced or modified by this feature: **set ip next-hop verify-availability**.

CHAPTER 5

IPv6 Policy-Based Routing

Policy-based routing (PBR) in both IPv6 and IPv4 allows a user to manually configure how received packets should be routed. PBR allows the user to identify packets by using several attributes and to specify the next hop or the output interface to which the packet should be sent. PBR also provides a basic packet-marking capability.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 IPv6 Policy-Based Routing

Policy-Based Routing Overview

Policy-based routing (PBR) gives you a flexible means of routing packets by allowing you to configure a defined policy for traffic flows, which lessens reliance on routes derived from routing protocols. Therefore, PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to set the IPv6 precedence. For a simple policy, you can use any one of these tasks; for a complex policy, you can use all of them. It also allows you to specify a path for certain traffic, such as priority traffic over a high-cost link.

PBR for IPv6 may be applied to both forwarded and originated IPv6 packets. For forwarded packets, PBR for IPv6 will be implemented as an IPv6 input interface feature, supported in the following forwarding paths:

- Process
- Cisco Express Forwarding (formerly known as CEF)
- Distributed Cisco Express Forwarding

Policies can be based on the IPv6 address, port numbers, protocols, or packet size.

PBR allows you to perform the following tasks:

- Classify traffic based on extended access list criteria. Access lists, then, establish the match criteria.
- Set IPv6 precedence bits, giving the network the ability to enable differentiated classes of service.
- Route packets to specific traffic-engineered paths; you might need to route them to allow a specific quality of service (QoS) through the network.

PBR allows you to classify and mark packets at the edge of the network. PBR marks a packet by setting precedence value. The precedence value can be used directly by devices in the network core to apply the appropriate QoS to a packet, which keeps packet classification at your network edge.

How Policy-Based Routing Works

All packets received on an interface with policy-based routing (PBR) enabled are passed through enhanced packet filters called route maps. The route maps used by PBR dictate the policy, determining where to forward packets.

Route maps are composed of statements. The route map statements can be marked as permit or deny, and they are interpreted in the following ways:

- If a packet matches all match statements for a route map that is marked as permit, the device attempts to policy route the packet using the set statements. Otherwise, the packet is forwarded normally.
- If the packet matches any match statements for a route map that is marked as deny, the packet is not subject to PBR and is forwarded normally.
- If the statement is marked as permit and the packets do not match any route map statements, the packets are sent back through normal forwarding channels and destination-based routing is performed.

You must configure policy-based routing (PBR) on the interface that receives the packet, and not on the interface from which the packet is sent.

Packet Matching

Policy-based routing (PBR) for IPv6 will match packets using the **match ipv6 address** command in the associated PBR route map. Packet match criteria are those criteria supported by IPv6 access lists, as follows:

- Input interface

- Source IPv6 address (standard or extended access control list [ACL])
- Destination IPv6 address (standard or extended ACL)
- Protocol (extended ACL)
- Source port and destination port (extended ACL)
- DSCP (extended ACL)
- Flow-label (extended ACL)
- Fragment (extended ACL)

Packets may also be matched by length using the **match length** command in the PBR route map.

Match statements are evaluated first by the criteria specified in the **match ipv6 address** command and then by the criteria specified in the **match length** command. Therefore, if both an ACL and a length statement are used, a packet will first be subject to an ACL match. Only packets that pass the ACL match will be subject to the length match. Finally, only packets that pass both the ACL and the length statement will be policy routed.

Packet Forwarding Using Set Statements

Policy-based routing (PBR) for IPv6 packet forwarding is controlled by using a number of set statements in the PBR route map. These set statements are evaluated individually in the order shown, and PBR will attempt to forward the packet using each of the set statements in turn. PBR evaluates each set statement individually, without reference to any prior or subsequent set statement.

You may set multiple forwarding statements in the PBR for IPv6 route map. The following set statements may be specified:

- IPv6 next hop. The next hop to which the packet should be sent. The next hop must be present in the Routing Information Base (RIB), it must be directly connected, and it must be a global IPv6 address. If the next hop is invalid, the set statement is ignored.
- Output interface. A packet is forwarded out of a specified interface. An entry for the packet destination address must exist in the IPv6 RIB, and the specified output interface must be in the set path. If the interface is invalid, the statement is ignored.
- Default IPv6 next hop. The next hop to which the packet should be sent. It must be a global IPv6 address. This set statement is used only when there is no explicit entry for the packet destination in the IPv6 RIB.
- Default output interface. The packet is forwarded out of a specified interface. This set statement is used only when there is no explicit entry for the packet destination in the IPv6 RIB.

Note The order in which PBR evaluates the set statements is the order in which they are listed above. This order may differ from the order in which route-map set statements are listed by **show** commands.

When to Use Policy-Based Routing

Policy-based routing (PBR) can be used if you want certain packets to be routed some way other than the obvious shortest path. For example, PBR can be used to provide the following functionality:

- Equal access
- Protocol-sensitive routing
- Source-sensitive routing
- Routing based on interactive traffic versus batch traffic
- Routing based on dedicated links

Some applications or traffic can benefit from Quality of Service (QoS)-specific routing; for example, you could transfer stock records to a corporate office on a higher-bandwidth, higher-cost link for a short time while sending routine application data such as e-mail over a lower-bandwidth, lower-cost link.

How to Enable IPv6 Policy-Based Routing

Enabling PBR on an Interface

To enable PBR for IPv6, create a route map that specifies the packet match criteria and the desired policy-route action. Then, associate the route map on the required interface. All packets arriving on the specified interface that match the match clauses will be subject to PBR.

Depending on your release, IPv6 PBR allows users to override normal destination IPv6 address-based routing and forwarding results. VPN routing and forwarding (VRF) allows multiple routing instances in Cisco software. The PBR feature is VRF-aware, which means that it works under multiple routing instances, beyond the default or global routing table.

In PBR, the **set vrf** command decouples the VRF and interface association and allows the selection of a VRF based on ACL-based classification using existing PBR or route-map configurations. PBR, through the **set vrf** command, provides a single device with multiple routing tables and the ability to select routes based on ACL classification. The device classifies packets based on ACL, selects a routing table, looks up the destination address, and then routes the packet.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. Enter one of the following commands:
   - **match length** *minimum-length maximum-length*
   - **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
5. Enter one of the following commands:
   - **set ipv6 precedence** *precedence-value*
   - **set ipv6 next-hop** *global-ipv6-address* [*global-ipv6-address...*]
   - **set interface** *type number* [*...type number*]
   - **set ipv6 default next-hop** *global-ipv6-address* [*global-ipv6-address...*]
   - **set default interface** *type number* [*...type number*]
   - **set vrf** *vrf-name*
6. **exit**
7. **interface** *type number*
8. **ipv6 policy route-map** *route-map-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map rip-to-ospf permit`	Enters route-map configuration mode and defines conditions for redistributing routes from one routing protocol into another or enables policy routing.
Step 4	Enter one of the following commands:	Specifies the match criteria.

	Command or Action	Purpose

	• **match length** *minimum-length maximum-length*
	• **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
	
	Example:
	`Device(config-route-map)# match length 3 200`
	`Device(config-route-map)# match ipv6 address marketing`
		• You can specify any or all of the following:
		  • Match the Level 3 length of the packet.
		  • Match a specified IPv6 access list.
		  • If you do not specify a **match** command, the route map applies to all packets.

Step 5	Enter one of the following commands:
	• **set ipv6 precedence** *precedence-value*
	• **set ipv6 next-hop** *global-ipv6-address* [*global-ipv6-address...*]
	• **set interface** *type number* [*...type number*]
	• **set ipv6 default next-hop** *global-ipv6-address* [*global-ipv6-address...*]
	• **set default interface** *type number* [*...type number*]
	• **set vrf** *vrf-name*
	
	Example:
	`Device(config-route-map)# set ipv6 precedence 1`
	`Device(config-route-map)# set ipv6 next-hop 2001:DB8:2003:1::95`
	`Device(config-route-map)# set interface serial 0/0`
	`Device(config-route-map)# set ipv6 default next-hop 2001:DB8:2003:1::95`
	`Device(config-route-map)# set default interface ethernet 0`
	`Device(config-route-map)# set vrf vrfname`
		Specifies the action or actions to be taken on the packets that match the criteria.
		• You can specify any or all of the following actions:
		  • Set the precedence value in the IPv6 header.
		  • Set the next hop for the packet (the next hop must be an adjacent device).
		  • Set output interface for the packet.
		  • Set the next hop for the packet, if there is no explicit route for this destination.
		  • Set the output interface for the packet, if there is no explicit route for this destination.
		  • Set VRF instance selection within a route map for a policy-based routing VRF selection.

Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number and enters interface configuration mode.

	Command or Action	Purpose
Step 8	**ipv6 policy route-map** *route-map-name* **Example:** `Device(config-if)# ipv6 policy-route-map interactive`	Identifies a route map to be used for IPv6 PBR on an interface.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Enabling Local PBR for IPv6

Packets that are generated by the device are not normally policy routed. Perform this task to enable local IPv6 policy-based routing (PBR) for such packets, indicating which route map the device should use.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 local policy route-map** *route-map-name*
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ipv6 local policy route-map** *route-map-name* **Example:** `Device(config)# ipv6 local policy route-map pbr-src-90`	Configures IPv6 PBR for packets generated by the device.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Verifying the Configuration and Operation of PBR for IPv6

SUMMARY STEPS

1. **enable**
2. **show ipv6 policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 policy** **Example:** `Device# show ipv6 policy`	Displays IPv6 policy routing packet activity.

Troubleshooting PBR for IPv6

Policy routing analyzes various parts of the packet and then routes the packet based on certain user-defined attributes in the packet.

SUMMARY STEPS

1. **enable**
2. **show route-map** [*map-name* | **dynamic** [*dynamic-map-name* | **application** [*application-name*]] | **all**] [**detailed**]
3. **debug ipv6 policy** [*access-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show route-map** [*map-name* | **dynamic** [*dynamic-map-name* | **application** [*application-name*]] | **all**] [**detailed**] **Example:** `Device# show route-map`	Displays all route maps configured or only the one specified.
Step 3	**debug ipv6 policy** [*access-list-name*] **Example:** `Device# debug ipv6 policy`	Enables debugging of the IPv6 policy routing packet activity.

Configuration Examples for IPv6 Policy-Based Routing

Example: Enabling PBR on an Interface

In the following example, a route map named pbr-dest-1 is created and configured. The route map specifies the packet match criteria and the desired policy-route action. PBR is then enabled on Ethernet interface 0/0.

```
ipv6 access-list match-dest-1
  permit ipv6 any 2001:DB8:2001:1760::/32
route-map pbr-dest-1 permit 10
  match ipv6 address match-dest-1
  set interface serial 0/0
interface Ethernet0/0
  ipv6 policy route-map interactive
```

Example: Enabling Local PBR for IPv6

In the following example, packets with a destination IPv6 address that match the IPv6 address range allowed by access list pbr-src-90 are sent to the device at IPv6 address 2001:DB8:2003:1::95:

```
ipv6 access-list src-90
  permit ipv6 host 2001:DB8:2003::90 2001:DB8:2001:1000::/64
route-map pbr-src-90 permit 10
  match ipv6 address src-90
  set ipv6 next-hop 2001:DB8:2003:1::95
ipv6 local policy route-map pbr-src-90
```

Example: Verifying IPv6 PBR Configuration

The following sample output from the **show ipv6 policy** command displays the IPv6 PBR configuration enabled on interface Ethernet interface 0/0:

```
Device# show ipv6 policy

Interface                 Routemap
Ethernet0/0               src-1
```

Example: Verifying Route-Map Information

The following sample output from the **show route-map** command displays specific route-map information, such as a count of policy matches:

```
Device# show route-map

route-map bill, permit, sequence 10
  Match clauses:
  Set clauses:
  Policy routing matches:0 packets, 0 bytes
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

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 IPv6 Policy-Based Routing

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 7: Feature Information for IPv6 Policy-Based Routing

Feature Name	Releases	Feature Information
IPv6 Policy-Based Routing	15.1(1)SY	Policy-based routing for IPv6 allows a user to manually configure how received packets should be routed. The following commands were introduced or modified: **debug fm ipv6 pbr**, **debug ipv6 policy**, **ipv6 local policy route-map**, **ipv6 policy route-map**, **match ipv6 address**, **match length**, **route-map**, **set default interface**, **set interface**, **set ipv6 default next-hop**, **set ipv6 next-hop (PBR)**, **set ipv6 precedence**, **set vrf**, **show fm ipv6 pbr all**, **show fm ipv6 pbr interface**, **show ipv6 policy**, and **show route-map**.

CHAPTER 6

Multi-VRF Selection Using Policy-Based Routing

The Multi-VRF Selection Using Policy-Based Routing (PBR) feature allows a specified interface on a provider edge (PE) device to route packets to Virtual Private Networks (VPNs) based on packet length or match criteria defined in an IP access list.

You can enable VPN routing and forwarding (VRF) selection by policy routing packets through a route map, through the global routing table, or to a specified VRF.

You can enable policy-routing packets for VRF instances by using route map commands with **set** commands.

On supported hardware, you can configure both the Multi-VRF Selection Using Policy-Based Routing feature and the MPLS VPN VRF Selection Based on a Source IP Address feature on the same interface.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

Prerequisites for Multi-VRF Selection Using Policy-Based Routing

- The device must support policy-based routing (PBR) in order for you to configure this feature. For platforms that do not support PBR, use the MPLS VPN VRF Selection Based on a Source IP Address feature.
- A Virtual Private Network (VPN) virtual routing and forwarding (VRF) instance must be defined before you configure this feature. An error message is displayed on the console if no VRF exists.

RestrictionsforMulti-VRFSelectionUsingPolicy-BasedRouting

- All commands that aid in routing also support hardware switching, except for the **set ip next-hop verify availability** command because Cisco Discovery Protocol information is not available in the line cards.
- Protocol Independent Multicast (PIM) and multicast packets do not support policy-based routing (PBR) and cannot be configured for a source IP address that is a match criterion for this feature.
- The **set vrf** and **set ip global next-hop** commands can be configured with the **set default interface**, **set interface** , **set ip default next-hop**, and **set ip next-hop** commands. But the **set vrf** and **set ip global next-hop** commands take precedence over the **set default interface**, **set interface** , **set ip default next-hop**, and **set ip next-hop** commands. No error message is displayed if you attempt to configure the **set vrf** command with any of these three **set** commands.
- The Multi-VRF Selection Using Policy-Based Routing feature cannot be configured with IP prefix lists.
- The **set global** and **set vrf** commands cannot be simultaneously applied to a route map.
- The Multi-VRF Selection Using Policy-Based Routing feature supports VRF-lite; that is, only IP routing protocols run on the device. Multiprotocol Label Switching (MPLS) and Virtual Private Networks (VPNs) cannot be configured. However, the **set vrf** command will work in MPLS VPN scenarios.

Information About Multi-VRF Selection Using Policy-Based Routing

Policy Routing of VPN Traffic Based on Match Criteria

The Multi-VRF Selection Using Policy-Based Routing feature is an extension of the MPLS VPN VRF Selection Based on a Source IP Address feature. The Multi-VRF Selection Using Policy-Based Routing feature allows you to policy route Virtual Private Network (VPN) traffic based on match criteria. Match criteria are defined in an IP access list and/or are based on packet length. The following match criteria are supported in Cisco software:

- IP access lists—Define match criteria based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access list configuration options in Cisco software can be used to define match criteria.
- Packet lengths—Define match criteria based on the length of a packet, in bytes. The packet length filter is defined in a route map with the **match length** route-map configuration command.

Policy routing is defined in the route map. The route map is applied to the incoming interface with the **ip policy route-map** interface configuration command. An IP access list is applied to the route map with the **match ip address** route-map configuration command. Packet length match criteria are applied to the route map with the **match length** route-map configuration command. The **set** action is defined with the **set vrf** route-map configuration command. The match criteria are evaluated, and the appropriate VRF is selected by the **set** command. This combination allows you to define match criteria for incoming VPN traffic and policy route VPN packets out to the appropriate virtual routing and forwarding (VRF) instance.

Policy-Based Routing set Commands

Policy-routing Packets for VRF Instances

To enable policy-routing packets for virtual routing and forwarding (VRF) instances, you can use route map commands with the following **set** commands. They are listed in the order in which the device uses them during the routing of packets.

- **set tos**—Sets the Type of Service (TOS) bits in the header of an IP packet.
- **set df**—Sets the Don't Fragment (DF) bit in the header of an IP packet.
- **set vrf**—Routes packets through the specified interface. The destination interface can belong only to a VRF instance.
- **set global**—Routes packets through the global routing table. This command is useful for routing ingress packets belonging to a specific VRF through the global routing table.
- **set ip vrf next-hop**—Indicates where to output IPv4 packets that pass a match criteria of a route map for policy routing when the IPv4 next hop must be under a specified VRF.
- **set ipv6 vrf next-hop**—Indicates where to output IPv6 packets that pass a match criteria of a route map for policy routing when the IPv6 next hop must be under a specified VRF.
- **set ip global next-hop**—Indicates where to forward IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table. The global keyword explicitly defines that IPv4 next-hops are under the global routing table.
- **set ipv6 global next-hop**—Indicates where to forward IPv6 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table. The global keyword explicitly defines that IPv6 next-hops are under the global routing table.
- **set interface**—When packets enter a VRF, routes the packets out of the egress interface under the same VRF according to the set interface policy, provided that the Layer 2 rewrite information is available.
- **set ip default vrf**—Provides IPv4 inherit-VRF and inter-VRF routing. With inherit-VRF routing, IPv4 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv4 packets arriving at a VRF interface are routed through any other outgoing VRF interface.

- **set ipv6 default vrf—**Provides IPv6 inherit-VRF and inter-VRF routing. With inherit-VRF routing, IPv6 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv6 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set ip default global—**Provides IPv4 VRF to global routing.
- **set ipv6 default global—**Provides IPv6 VRF to global routing.
- **set default interface—**Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination. The interface can belong to any VRF.
- **set ip default next-hop—**Indicates where to output IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.
- **set ipv6 default next-hop—**Indicates where to IPv6 output packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.

Change of Normal Routing and Forwarding Behavior

When you configure policy-based routing (PBR), you can use the following six **set** commands to change normal routing and forwarding behavior. Configuring any of these **set** commands, with the potential exception of the **set ip next-hop** command, overrides the routing behavior of packets entering the interface if the packets do not belong to a virtual routing and forwarding (VRF) instance. The packets are routed from the egress interface across the global routing table.

- **set default interface—**Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination.
- **set interface—**When packets enter a VRF interface, routes the packets out of the egress interface under the same VRF according to the set interface policy, provided that the Layer 2 rewrite information is available.

  Note The interface must be a peer-to-peer (P2P) interface.

- **set ip default next-hop—**Indicates where to output IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.
- **set ipv6 default next-hop—**Indicates where to output IPv6 packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.
- **set ip next-hop—**Indicates where to output IPv4 packets that pass a match criterion of a route map for policy routing. If an IPv4 packet is received on a VRF interface and is transmitted from another interface within the same VPN, the VRF context of the incoming packet is inherited from the interface.
- **set ipv6 next-hop—**Indicates where to output IPv6 packets that pass a match criterion of a route map for policy routing. If an IPv6 packet is received on a VRF interface and is transmitted from another interface within the same Virtual Private Network (VPN), the VRF context of the incoming packet is inherited from the interface.

Support of Inherit-VRF Inter-VRF and VRF-to-Global Routing

The Multi-VRF Selection Using Policy-Based Routing (PBR) feature supports inherit-VRF and inter-VRF. With inherit-VRF routing, packets arriving at a virtual routing and forwarding (VRF) interface are routed by the same outgoing VRF interface. With inter-VRF routing, packets arriving at a VRF interface are routed through any other outgoing VRF interface.

VRF-to-global routing causes packets that enter any VRF interface to be routed through the global routing table. When a packet arrives on a VRF interface, the destination lookup normally is done only in the corresponding VRF table. If a packet arrives on a global interface, the destination lookup is done in the global routing table.

The Multi-VRF Selection Using Policy-Based Routing feature modifies the following **set** commands to support inherit-VRF, inter-VRF, and VRF-to-global routing. The commands are listed in the order in which the device uses them during the routing of packets.

- **set global—**Routes packets through the global routing table. This command is useful for routing ingress packets belonging to a specific VRF through the global routing table.
- **set ip global next-hop—**Indicates where to forward IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table.
- **set ipv6 global next-hop—**Indicates where to forward IPv6 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table.
- **set ip vrf next-hop—**Causes the device to look up the IPv4 next hop in the VRF table. If an IPv4 packet arrives on an interface that belongs to a VRF and the packet needs to be routed through a different VRF, you can use the **set ip vrf next-hop** command.
- **set ipv6 vrf next-hop—**Causes the device to look up the IPv6 next hop in the VRF table. If an IPv6 packet arrives on an interface that belongs to a VRF and the packet needs to be routed through a different VRF, you can use the **set ipv6 vrf next-hop** command.
- **set ip default vrf—**Provides IPv4 inherit-VRF and inter-VRF routing. With IPv4 inherit-VRF routing, IPv4 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv4 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set ipv6 default vrf—**Provides IPv6 inherit-VRF and inter-VRF routing. With IPv6 inherit-VRF routing, IPv6 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv6 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set interface—**When packets enter a VRF, routes the packets out of the egress interface under the same VRF, according to the set interface policy, provided that the Layer 2 rewrite information is available.
- **set default interface—**Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination. The interface can belong to any VRF.
- **set ip next-hop—**Routes IPv4 packets through the global routing table in an IPv4-to-IPv4 routing and forwarding environment.
- **set ipv6 next-hop—**Routes IPv6 packets through the global routing table in an IPv6-to-IPv6 routing and forwarding environment.
- **set vrf—**Selects the appropriate VRF after a successful match occurs in the route map. VRS-aware PSV allows only inter-VRF (or VRF-to-VRF) switching.

How to Configure Multi-VRF Selection Using Policy-Based Routing

Defining the Match Criteria for Multi-VRF Selection Using Policy-Based Routing

Define the match criteria for the Multi-VRF Selection using Policy-Based Routing (PBR) feature so that you can selectively route the packets instead of using their default routing and forwarding.

The match criteria for the Multi-VRF Selection using Policy-Based Routing are defined in an access list. Standard, named, and extended access lists are supported.

You can define the match criteria based on the packet length by configuring the **match length** route-map configuration command. This configuration option is defined entirely within a route map.

The following sections explain how to configure PBR route selection:

Configuring Multi-VRF Selection Using Policy-Based Routing with a Standard Access List

Before You Begin

The tasks in the following sections assume that the virtual routing and forwarding (VRF) instance and associated IP address are already defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} [**source** *source-wildcard*] [**log**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} [**source** *source-wildcard*] [**log**]	Creates an access list and defines the match criteria for the route map.

	Command or Action	Purpose
	Example: `Device(config)# access-list 40 permit source 10.1.1.0/24 0.0.0.255`	• Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options to define match criteria. • The example creates a standard access list numbered 40. This filter permits traffic from any host with an IP address in the 10.1.1.0/24 subnet.

Configuring Multi-VRF Selection Using Policy-Based Routing with a Named Extended Access List

To configure Multi-VRF Selection using Policy-Based Routing (PBR) with a named extended access list, complete the following steps.

Before You Begin

The tasks in the following sections assume that the virtual routing and forwarding (VRF) instance and associated IP address are already defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*]
4. [*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos***tos*] [**ttl** *operator-vaue*] [**log**] [**time-range** *time-range-name*] [**fragments**]

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device(config)# ip access-list extended NAMEDACL`	Specifies the IP access list type and enters the corresponding access list configuration mode. • You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos***tos*] [**ttl** *operator-vaue*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit ip any any option any-options`	Defines the criteria for which the access list will permit or deny packets. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options to define match criteria. • The example creates a named access list that permits any configured IP option.

Configuring Multi-VRF Selection in a Route Map

Incoming packets are filtered through the match criteria that are defined in the route map. After a successful match occurs, the **set** command configuration determines the VRF through which the outbound Virtual Private Network (VPN) packets will be policy routed.

Before You Begin

You must define the virtual routing and forwarding (VRF) instance before you configure the route map; otherwise an error message appears on the console.

A receive entry must be added to the VRF selection table with the **ip vrf receive** command. If a match and set operation occurs in the route map but there is no receive entry in the local VRF table, the packet will be dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. Do one of the following :
    - **set ip vrf** *vrf-name* **next-hop** *global-ipv4-address* [*...global-ipv4-address*]
    - **set ipv6 vrf** *vrf-name* **next-hop** *global-ipv6-address* [*...global-ipv6-address*]
    - **set ip next-hop recursive vrf** *global-ipv4-address* [*...global-ipv4-address*]
    - **set ip global next-hop** *global-ipv4-address* [*...global-ipv4-address*]
    - **set ipv6 global next-hop** *global-ipv6-address* [*...global-ipv6-address*]
5. Do one of the following:
    - **match ip address** {*acl-number* [*acl-name* | *acl-number*]}
    - **match length** *minimum-lengthmaximum-length*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map map1 permit 10`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. • Enters route-map configuration mode.
Step 4	Do one of the following : • **set ip vrf** *vrf-name* **next-hop** *global-ipv4-address* [*...global-ipv4-address*] • **set ipv6 vrf** *vrf-name* **next-hop** *global-ipv6-address* [*...global-ipv6-address*]	Indicates where to forward packets that pass a match criterion of a route map for policy routing when the IPv4 next hop must be under a specified VRF. Indicates where to forward packets that pass a match criterion of a route map for policy routing when the IPv6 next hop must be under a specified VRF.

	Command or Action	Purpose

	• **set ip next-hop recursive vrf** *global-ipv4-address* [...*global-ipv4-address*]
	• **set ip global next-hop** *global-ipv4-address* [...*global-ipv4-address*]
	• **set ipv6 global next-hop** *global-ipv6-address* [...*global-ipv6-address*]
	
	Example:
	`Device(config-route-map)# set ip vrf myvrf next-hop 10.0.0.0`
	
	Example:
	`Device(config-route-map)# set ipv6 vrf myvrf next-hop 2001.DB8:4:1::1/64`
	
	Example:
	`Device(config-route-map)# set ip next-hop recursive vrf 10.0.0.0`
	
	Example:
	`Device(config-route-map)# set ip global next-hop 10.0.0.0`
	
	Example:
	`Device(config-route-map)# set ipv6 global next-hop 2001.DB8:4:1::1/64`
		Indicates the IPv4 address to which destination or next hop is used for packets that pass the match criterion configured in the route map.
		Indicates the IPv4 address to forward packets that pass a match criterion of a route map for policy routing and for which the software uses the global routing table.
		Indicates the IPv6 address to forward packets that pass a match criterion of a route map for policy routing and for which the software uses the global routing table.

Step 5	Do one of the following:
	• **match ip address** {*acl-number* [*acl-name* | *acl-number*]}
	• **match length** *minimum-lengthmaximum-length*
	
	Example:
	`Device(config-route-map)# match ip address 1 or`
	
	Example:
	`Device(config-route-map)# match length 3 200`
		Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. IP access lists are supported.
		• The example configures the route map to use standard access list 1 to define match criteria.
		
		Specifies the Layer 3 packet length in the IP header as a match criterion in a class map.
		• The example configures the route map to match packets that are 3 to 200 bytes in length.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-route-map)# end`	Returns to privileged EXEC mode.

Configuring Multi-VRF Selection Using Policy-Based Routing and IP VRF Receive on the Interface

The route map is attached to the incoming interface with the **ip policy route-map** interface configuration command.

The source IP address must be added to the virtual routing and forwarding (VRF) selection table. VRF selection is a one-way (unidirectional) feature. It is applied to the incoming interface. If a **match** and **set** operation occurs in the route map but there is no receive entry in the local VRF table, the packet is dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip policy route-map** *map-tag*
5. **ip vrf receive** *vrf-name*
6. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface FastEthernet 0/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map map1`	Identifies a route map to use for policy routing on an interface. • The configuration example attaches the route map named map1 to the interface.
Step 5	**ip vrf receive** *vrf-name* **Example:** `Device(config-if)# ip vrf receive VRF-1`	Adds the IP addresses that are associated with an interface into the VRF table. • This command must be configured for each VRF that will be used for VRF selection.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying the Configuration of Multi-VRF Selection Using Policy-Based Routing

To verify the configuration of the Multi-VRF Selection Using Policy-Based Routing (PBR) feature, perform the following steps. You can enter the commands in any order.

SUMMARY STEPS

1. **show ip access-list** [*access-list-number* | *access-list-name*]
2. **show route-map** [*map-name*]
3. **show ip policy**

DETAILED STEPS

Step 1 **show ip access-list** [*access-list-number* | *access-list-name*]
Verifies the configuration of match criteria for Multi-VRF Selection Using Policy-Based Routing. The command output displays three subnet ranges defined as match criteria in three standard access lists:

Example:

```
Device# show ip access-list

Standard IP access list 40
 10 permit 10.1.0.0, wildcard bits 0.0.255.255
Standard IP access list 50
 10 permit 10.2.0.0, wildcard bits 0.0.255.255
Standard IP access list 60
 10 permit 10.3.0.0, wildcard bits 0.0.255.255
```

Step 2 **show route-map** [*map-name*]

Verifies **match** and **set** commands within the route map:

Example:

```
Device# show route-map
```

The output displays the match criteria and set action for each route-map sequence. The output also displays the number of packets and bytes that have been policy routed per each route-map sequence.

Example:

```
Device# show route-map map1

route-map map1, permit, sequence 10
Match clauses:
Set clauses:
 ip next-hop vrf myvrf 10.5.5.5  10.6.6.6  10.7.7.7
 ip next-hop global 10.8.8.8  10.9.9.9
Policy routing matches: 0 packets, 0 bytes
Device# show route-map map2
route-map map2, permit, sequence 10
Match clauses:
Set clauses:
 vrf myvrf
Policy routing matches: 0 packets, 0 bytes
Device# show route-map map3
route-map map3, permit, sequence 10
Match clauses:
Set clauses:
 global
Policy routing matches: 0 packets, 0 bytes
```

The following **show route-map** command displays output from the **set ip vrf next-hop** command:

Example:

```
Device(config)# route-map test

Device(config-route-map)# set ip vrf myvrf next-hop
Device(config-route-map)# set ip vrf myvrf next-hop 192.168.3.2
Device(config-route-map)# match ip address 255 101
Device(config-route-map)# end
Device# show route-map

route-map test, permit, sequence 10
 Match clauses:
  ip address (access-lists): 101
 Set clauses:
  ip vrf myvrf next-hop 192.168.3.2
 Policy routing matches: 0 packets, 0 bytes
```

The following **show route-map** command displays output from the **set ip global** command:

Example:

```
Device(config)# route-map test
Device(config-route-map)# match ip address 255 101
Device(config-route-map)# set ip global next-hop 192.168.4.2
Device(config-route-map)# end
Device# show route-map

*May 25 13:45:55.551: %SYS-5-CONFIG_I: Configured from console by consoleout-map
route-map test, permit, sequence 10
 Match clauses:
  ip address (access-lists): 101
 Set clauses:
  ip global next-hop 192.168.4.2
 Policy routing matches: 0 packets, 0 bytes
```

Step 3 **show ip policy**

Verifies the Multi-VRF Selection Using Policy-Based Routing policy.

Example:

```
Device# show ip policy
```

The following **show ip policy** command output displays the interface and associated route map that is configured for policy routing:

Example:

```
Device# show ip policy

Interface                Route map
FastEthernet0/1/0        PBR-VRF-Selection
```

Configuration Examples for Multi-VRF Selection Using Policy-Based Routing

Example: Defining the Match Criteria for Multi-VRF Selection Using Policy-Based Routing

In the following example, three standard access lists are created to define match criteria for three different subnetworks. Any packets received on FastEthernet interface 0/1/0 will be policy routed through the PBR-VRF-Selection route map to the virtual routing and forwarding (VRF) that is matched in the same route-map sequence. If the source IP address of the packet is part of the 10.1.0.0/24 subnet, VRF1 will be used for routing and forwarding.

```
access-list 40 permit source 10.1.0.0 0.0.255.255
```

```
access-list 50 permit source 10.2.0.0 0.0.255.255
access-list 60 permit source 10.3.0.0 0.0.255.255
route-map PBR-VRF-Selection permit 10
 match ip address 40
 set vrf VRF1
 !
route-map PBR-VRF-Selection permit 20
 match ip address 50
 set vrf VRF2
 !
route-map PBR-VRF-Selection permit 30
 match ip address 60
 set vrf VRF3
 !
interface FastEthernet 0/1/0
 ip address 192.168.1.6 255.255.255.252
 ip vrf forwarding VRF4
 ip policy route-map PBR-VRF-Selection
 ip vrf receive VRF1
 ip vrf receive VRF2
 ip vrf receive VRF3
```

Example: Configuring Multi-VRF Selection in a Route Map

The following example shows a **set ip vrf next-hop** command that applies policy-based routing to the virtual routing and forwarding (VRF) interface named myvrf and specifies that the IP address of the next hop is 10.0.0.2:

```
Device(config)# route-map map1 permit
Device(config)# set vrf myvrf
Device(config-route-map)# set ip vrf myvrf next-hop 10.0.0.2
Device(config-route-map)# match ip address 101
Device(config-route-map)# end
```

The following example shows a **set ip global** command that specifies that the device should use the next hop address 10.0.0.1 in the global routing table:

```
Device(config-route-map)# set ip global next-hop 10.0.0.1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference
IP access list commands	*Cisco IOS Security Command Reference*

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 Multi-VRF Selection Using Policy-Based Routing

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 8: Feature Information for Multi-VRF Selection Using Policy-Based Routing

Feature Name	Releases	Feature Information
Multi-VRF Selection Using Policy-Based Routing (PBR)	12.2(33)SRB1 12.2(33)SXH1 12.4(24)T Cisco IOS XE Release 2.2	The Multi-VRF Selection Using Policy-Based Routing (PBR) feature allows a specified interface on a provider edge (PE) router to route packets to Virtual Private Networks (VPNs) based on packet length or match criteria defined in an IP access list. This feature and the MPLS VPN VRF Selection Based on Source IP Address feature can be configured together on the same interface In Cisco IOS Release 12.2(33)SRB1, this feature was introduced. In Cisco IOS Release 12.2(33)SXH1, support was added. In Cisco IOS Release 12.4(24)T, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were modified: **set ip global next-hop** and **set ip vrf next-hop**.
IPv6 VRF-Aware PBR Next-hop Enhancement	15.2(2)S Cisco IOS XE Release 3.6S	In Cisco IOS Release 15.2(2)S, this feature was introduced. In Cisco IOS XE Release 3.6S, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced: **set ipv6 default next-hop**, **set ipv6 next-hop (PBR)**

Glossary

CE device—customer edge device. A device that is part of a customer network and that interfaces to a provider edge (PE) device.

Inherit-VRF routing—Packets arriving at a VRF interface are routed by the same outgoing VRF interface.

Inter-VRF routing—Packets arriving at a VRF interface are routed via any other outgoing VRF interface.

IP—Internet Protocol. Network layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security. Defined in RFC 791.

PBR—policy-based routing. PBR allows a user to manually configure how received packets should be routed.

PE device—provider edge device. A device that is part of a service provider's network and that is connected to a CE device. It exchanges routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.

VPN—Virtual Private Network. A collection of sites sharing a common routing table. A VPN provides a secure way for customers to share bandwidth over an ISP backbone network.

VRF—A VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.

VRF-lite—A feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs.

CHAPTER 7

Recursive Static Route

The Recursive Static Route feature enables you to install a recursive static route into the Routing Information Base (RIB) even if the next-hop address of the static route or the destination network itself is already available in the RIB as part of a previously learned route. This module explains recursive static routes and how to configure the Recursive Static Route feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 Recursive Static Route

When recursive static routes are enabled using route maps, only one route map can be entered per virtual routing and forwarding (VRF) instance or topology. If a second route map is entered, the new map will overwrite the previous one.

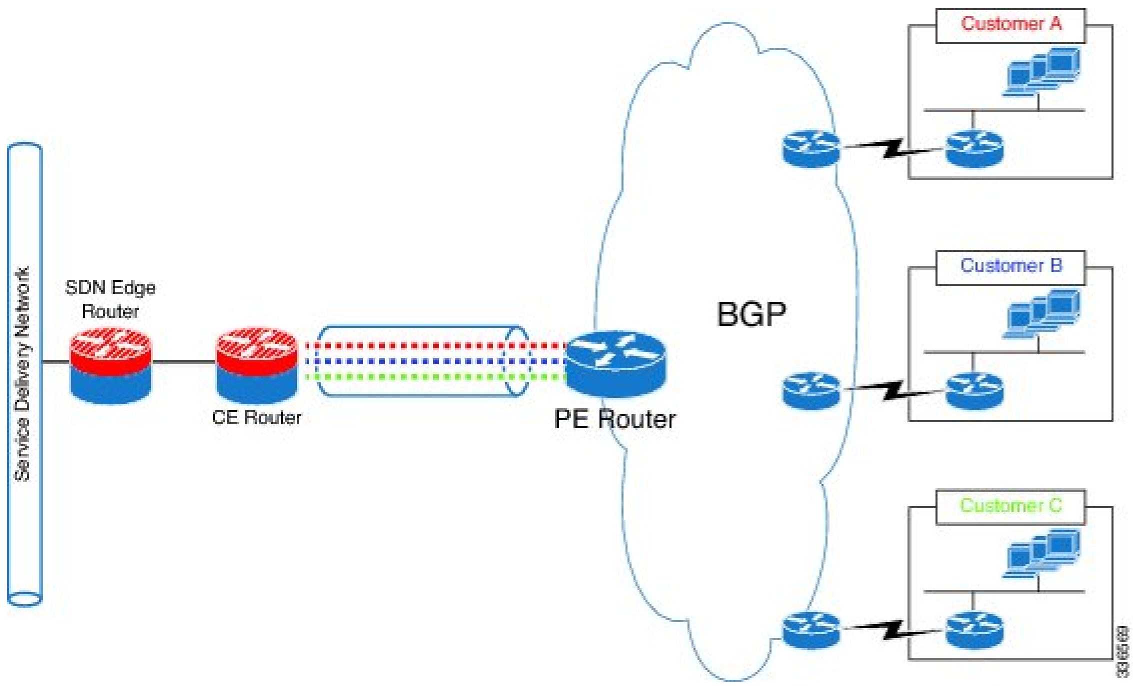
Information About Recursive Static Route

Recursive Static Routes

A recursive static route is a route whose next hop and the destination network are covered by another learned route in the Routing Information Base (RIB). Such static routes cannot be installed in the RIB because they are considered redundant routes. The Recursive Static Route feature allows you to install recursive static routes in the RIB, thereby allowing the redistribution of such specific routes within the network. When the learned route covering the next hop or the destination network is withdrawn from the RIB, the recursive static route also gets withdrawn from the RIB.

Given below is a detailed explanation of how recursive static routes work.

Figure 6: How Recursive Static Routes Work



The figure above shows three customers connected to three private virtual routing and forwarding (VRF) instances on a provider edge (PE) device. All three of them have private addressing in their networks.

Let us assume that the network on these VRFs is 10.0.0.0/8. The PE communicates these routes as is to the customer edge (CE) device, along with the VRF information of each customer. For security purposes and to avoid overlapping routes, the CE advertises only /32 routes to the service delivery network (SDN) edge device (which has only a service VRF) rather than advertising the whole 10.0.0.0/8 network for each customer. Static routes can be used to configure /32 routes for specific hosts. However, static routes that are recursive in nature cannot be configured for specific hosts. By default, recursive static routes are eliminated from the Routing Information Base (RIB) because these routes or the next hops to these routes may already be covered by another learned route in the RIB. The Recursive Static Route feature enables a recursive static route to be part of the RIB even if the next-hop address of the static route or the destination network of the static route is already available in the RIB as part of a previously learned route. Additionally, if the learned route that covers the next-hop gateway is withdrawn from the RIB, the recursive static route is also deleted from the RIB.

How to Install Recursive Static Route

Installing Recursive Static Routes in a VRF

Perform these steps to install recursive static routes in a specific virtual routing and forwarding (VRF) instance. You can configure the recursive-static-route functionality on any number of VRFs. Installing recursive static routes in specific VRFs allows you to retain the default RIB behavior (of removing recursive static routes) for the rest of the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4** | **ipv6**}
6. **exit**
7. **exit**
8. **ip route** [**vrf** *vrf-name*] *prefix mask ip-address*
9. **ip route static install-routes-recurse-via-nexthop** [**vrf** *vrf-name*]
10. **end**
11. **show running-config | include install**
12. **show ip route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Creates a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.

	Command or Action	Purpose
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Specifies a route distinguisher for a VRF instance.
Step 5	**address-family** {**ipv4** | **ipv6**} **Example:** `Device(config-vrf)# address-family ipv4`	Enters VRF address family configuration mode to specify an IPv4 or IPv6 address family for a VRF.
Step 6	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits VRF address family configuration mode.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode.
Step 8	**ip route** [**vrf** *vrf-name*] *prefix mask ip-address* **Example:** `Device(config)# ip route vrf vrf1 10.0.2.0 255.255.255.0 10.0.1.1`	Configures a static route for a specific VRF instance.
Step 9	**ip route static install-routes-recurse-via-nexthop** [**vrf** *vrf-name*] **Example:** `Device(config)# ip route static install-routes-recurse-via-nexthop vrf vrf1`	Enables recursive static routes to be installed in the RIB of a specific VRF instance.
Step 10	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 11	**show running-config | include install** **Example:** `Device# show running-config | inc install`	Displays all recursive static route configurations.
Step 12	**show ip route vrf** *vrf-name* **Example:** `Device# show ip route vrf vrf1`	Displays the IP routing table associated with a specific VRF.

Installing Recursive Static Routes Using a Route Map

Perform this task to install recursive static routes in a virtual routing and forwarding (VRF) instance defined by a route map. You can perform this task if you want to install recursive static routes for only a certain range of networks. If the **route-map** keyword is used without the **vrf** keyword, recursive static routes defined by the route map will be applicable for the global VRF or topology.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4** | **ipv6**}
6. **exit**
7. **exit**
8. **ip route** [**vrf** *vrf-name*] *prefix mask ip-address*
9. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
10. **route-map** *map-tag*
11. **match ip address** *access-list-number*
12. **exit**
13. **ip route static install-routes-recurse-via-nexthop** [**vrf** *vrf-name*] [**route-map** *map-name*]
14. **end**
15. **show running-config | include install**
16. **show ip route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Creates a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.

	Command or Action	Purpose
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Specifies a route distinguisher for a VRF instance.
Step 5	**address-family** {**ipv4** | **ipv6**} **Example:** `Device(config-vrf)# address-family ipv4`	Enters VRF address family configuration mode to specify an IPv4 or an IPv6 address-family type for a VRF.
Step 6	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits VRF address family configuration mode.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode.
Step 8	**ip route** [**vrf** *vrf-name*] *prefix mask ip-address* **Example:** `Device(config)# ip route vrf vrf1 10.0.2.0 255.255.255.0 10.0.1.1`	Configures a static route for a specific VRF instance.
Step 9	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 10 permit 10.0.2.0 255.255.255.0`	Defines a standard access list permitting addresses that need to be translated.
Step 10	**route-map** *map-tag* **Example:** `Device(config)# route-map map1`	Defines a route map to control route redistribution and enters route-map configuration mode.
Step 11	**match ip address** *access-list-number* **Example:** `Device(config-route-map)# match ip address 10`	Matches routes that have a destination network address that is permitted by a standard or extended access list.
Step 12	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode.
Step 13	**ip route static install-routes-recurse-via-nexthop** [**vrf** *vrf-name*] [**route-map** *map-name*] **Example:** `Device(config)# ip route static install-routes-recurse-via-nexthop vrf vrf1 route-map map1`	Enables installation of recursive static routes defined by a route map into the RIB of a specific VRF.

	Command or Action	Purpose
Step 14	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 15	**show running-config | include install** **Example:** `Device# show running-config | inc install`	Displays all recursive static route configurations.
Step 16	**show ip route vrf** *vrf-name* **Example:** `Device# show ip route vrf vrf1`	Displays the IP routing table associated with a specific VRF.

Configuration Examples for Recursive Static Route

•

Example: Installing Recursive Static Routes in a VRF

The following example shows how to install recursive static routes into a specific virtual routing and forwarding instance. By using the **vrf** keyword, you can ensure that recursive static routes are installed in the Routing Information Base (RIB) of only the specified VRF. The rest of the network retains the default behavior of not installing recursive static routes in the RIB. This example is based on the assumption that a 10.0.0.0/8 route is already installed dynamically or statically in the RIB of vrf1.

```
Device> enable
Device# configure terminal
Device(config)# vrf definition vrf1
Device(config-vrf)# rd 1:100
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit
Device(config-vrf)# exit
Device(config)# ip route vrf vrf1 10.0.2.0 255.255.255.0 10.0.1.1
Device(config)# ip route static install-routes-recurse-via-nexthop vrf vrf1
Device(config)# end
```

Example: Installing Recursive Static Routes using a Route Map

You can use the **route-map** keyword to install recursive static routes defined by the route map into the Routing Information Base (RIB). You can also specify a route map for a specific virtual routing and forwarding (VRF) instance to ensure that the route map is applied to only the specified VRF. In the example given below, a route map is specified for a specific VRF. This example is based on the assumption that a 10.0.0.0/8 route is already installed statically or dynamically in the RIB of vrf1.

```
Device> enable
Device# configure terminal
```

```
Device(config)# vrf definition vrf1
Device(config-vrf)# rd 100:2
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit
Device(config-vrf)# exit
Device(config)# access-list 10 permit 10.0.2.0 255.255.255.0
Device(config)# route-map map1
Device(config-route-map)# match ip address 10
Device(config-route-map)# exit
Device(config)# ip route static install-routes-recurse-via-nexthop vrf vrf1 route-map map1
Device(config)# ip route vrf vrf1 10.0.2.0 255.255.255.0 10.0.1.1
Device(config)# ip route vrf vrf1 10.0.3.0 255.255.255.0 10.0.1.1
Device(config)# end
```

In the example above, route 10.0.2.0 255.255.255.0 10.0.1.1 will be installed in the RIB, but the route 10.0.3.0 255. 255.255.0 10.0.1.1 will not be installed in the RIB because this route does not match the network defined in the route map.

Additional References for Recursive Static Route

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IP routing protocol-independent commands	Cisco IOS IP Routing: Protocol-Independent Command Reference

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 Recursive Static Route

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 9: Feature Information for Recursive Static Routes

Feature Name	Releases	Feature Information
Recursive Static Route	15.3(2)S 15.3(3)M 15.2(1)SY	The Recursive Static Route feature enables you to install a recursive static route into the Routing Information Base (RIB) even if the next-hop address of the static route or the destination network itself is already available in the RIB as part of a previously learned route. The following command was introduced: **ip route static install-routes-recurse-via-nexthop**.