CHAPTER 35

Configuring Policy-Based Routing

This chapter describes the tasks for configuring policy-based routing (PBR) on a Catalyst 4500 series switch and includes these major sections:

- About Policy-Based Routing, page 35-1
- Policy-Based Routing Configuration Tasks, page 35-6
- Policy-Based Routing Configuration Examples, page 35-8

Note For complete syntax and usage information for the switch commands used in this chapter, look at the *Cisco Catalyst 4500 Series Switch Command Reference* and related publications at this location:

http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html

If the command is not found in the Catalyst 4500 Command Reference, it is located in the larger Cisco IOS library. Refer to the *Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/en/US/products/ps6350/index.html

Note To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

About Policy-Based Routing

This section contains the following sections:

- About PBR, page 35-2
- PBR Flow Switching, page 35-5
- Using Policy-Based Routing, page 35-5

PBR gives you a flexible method of routing packets by allowing you to define policies for traffic flows, lessening reliance on routes derived from routing protocols. PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to specify paths for certain traffic, such as priority traffic over a high-cost link.

You can set up PBR as a way to route packets based on configured policies. For example, you can implement routing policies to allow or deny paths based on the identity of a particular end system, or an application protocol.

PBR allows you to perform the following tasks:

- Classify traffic based on extended access list criteria. Access lists, and then establish the match criteria.
- Route packets to specific traffic-engineered paths.

Policies can be based on IP address, port numbers, or protocols. For a simple policy, use any one of these descriptors; for a complicated policy, all of them.

About PBR

All packets received on an interface with PBR enabled are passed through enhanced packet filters known as route maps. The route maps used by PBR dictate the policy, determining to where the packets are forwarded.

Route maps are composed of statements, which can be marked as permit or deny. They are interpreted in the following ways:

- If a statement is marked as deny, the packets meeting the match criteria are sent back using the normal forwarding channels and destination-based routing is performed.
- If the statement is marked as permit and a packet matches the access-lists, then the first valid set clause is applied to that packet.

This is explained in more detail in the section Understanding Route-Maps, page 35-2.

You specify PBR on the incoming interface (the interface on which packets are received), not outgoing interface.

Understanding Route-Maps

PBR is implemented by applying a route-map on an incoming interface. A given interface can have only one route-map configured.

A route-map is configured at the global configuration parser mode. You can then apply this route-map on one or more interfaces (in the interface configuration parser sub-mode).

A route-map is comprised of one or more route-map statements. Each statement has a sequence number, as well as a permit or deny clause.

Each route-map statement contains **match** and **set** commands. The **match** command denotes the match criteria to be applied on the packet data. The **set** command denote the PBR action to be taken on the packet.

The following example shows a single route-map called rm-test and six route-map statements:

```
route-map rm-test permit 21
 match ip address 101
 set ip next-hop 21.1.1.1
!
route-map rm-test permit 22
 match ip address 102
 set ip next-hop 22.2.2.1
!
```

```
    route-map rm-test permit 23
     match ip address 101 2102
     set interface vlan23
!
    route-map rm-test deny 24
     match ip address 104
     set ip next-hop 24.4.4.1
    !
    route-map rm-test deny 25
     match ip address 105
     set ip next-hop 25.5.5.1
    !
    route-map rm-test permit 26
     match ip address 2104
     set ip next-hop 26.6.6.1
```

The numbers 21, 22, ... 26 are the sequence numbers of the route-map statements.

The following topics are discussed:

- PBR Route-Map Processing Logic, page 35-3
- PBR Route-Map Processing Logic Example, page 35-4

PBR Route-Map Processing Logic

When a packet is received on an interface configured with a route-map, the forwarding logic processes each route-map statement according to the sequence number.

If the route-map statement encountered is a **route-map...permit** statement:

- The packet is matched against the criteria in the **match** command. This command may refer to an ACL that may itself have one or more permit and/or deny expressions. The packet is matched against the expressions in the ACL, and a permit/deny decision is reached.
- If the decision reached is permit, then the PBR logic executes the action specified by the **set** command on the packet.
- If the decision reached is deny, then the PBR action (specified in the **set** command) is not applied. Instead the processing logic moves forward to look at the next route-map statement in the sequence (the statement with the next higher sequence number). If no next statement exists, PBR processing terminates, and the packet is routed using the default IP routing table.

If the route-map statement encountered is a **route-map... deny** statement:

- The packet is matched against the criteria given in the **match** command. This command may refer to an ACL that may itself have one or more permit and/or deny expressions. The packet is matched against the expressions in the ACL, and a permit/deny decision is reached.
- If the criteria decision is permit, then the PBR processing terminates, and the packet is routed using the default IP routing table.
- If the criteria decision is deny, then the PBR processing logic moves forward to look at the next route-map statement in the sequence (the statement with the next higher sequence number). If no next statement exists, PBR processing terminates, and the packet is routed using the default IP routing table.

Note The **set** command has no effect inside a **route-map... deny** statement.

PBR Route-Map Processing Logic Example

Consider a route-map called rm-test defined as follows:

```
access-list 101  permit tcp host 61.1.1.1 host 133.3.3.1 eq 101
access-list 102  deny   tcp host 61.1.1.1 host 133.3.3.1 eq 102
access-list 2102 permit tcp host 61.1.1.1 host 133.3.3.1 eq 102
access-list 104  deny   tcp host 61.1.1.1 host 133.3.3.1 eq 104
access-list 2104 permit tcp host 61.1.1.1 host 133.3.3.1 eq 104
access-list 105  permit tcp host 61.1.1.1 host 133.3.3.1 eq 105


route-map rm-test permit 21
 match ip address 101
 set ip next-hop 21.1.1.1
!
route-map rm-test permit 22
 match ip address 102
 set ip next-hop 22.2.2.1
!
route-map rm-test permit 23
 match ip address 101 2102
 set interface vlan23
!
route-map rm-test deny 24
 match ip address 104
 set ip next-hop 24.4.4.1
!

route-map rm-test deny 25
 match ip address 105
 set ip next-hop 25.5.5.1
!
route-map rm-test permit 26
 match ip address 2104
 set ip next-hop 26.6.6.1
```

- TCP packet from 61.1.1.1 to 133.3.3.1 with destination port 101
  - Matches ACL 101 in sequence #21.
  - PBR is switched through next-hop 21.1.1.1.

Note ACL 101 is also matched in sequence #23, but the processing doesn't reach that point

- TCP packet from 61.1.1.1 to 133.3.3.1 with destination port 102
  - In sequence #21, the ACL 101 action denies this packet (because all ACLs have an implicit deny). Processing advances to sequence #22.
  - In sequence #22, ACL 102 matches TCP port 102, but the ACL action is deny. Processing advances to sequence #23.
  - In sequence #23, ACL 2102 matches TCP port 102, and the ACL action is permit.
  - Packet is switched to output interface VLAN 23.

- TCP packet from 61.1.1.1 to 133.3.3.1 with destination port 105
  - Processing moves from sequence #21 to #24, because all ACLs in these sequence numbers have a deny action for port 105.
  - In sequence #25, ACL 105 has a permit action for TCP port 105.
  - The route-map deny takes effect, and the packet is routed using the default IP routing table.

PBR on Supervisor Engine 6-E, Supervisor Engine 6L-E, Catalyst 4900M, and Catalyst 4948E

Supervisor Engine 6-E, Supervisor Engine 6L-E, Catalyst 4900M, and Catalyst 4948E support matching route-map actions with a packet by installing entries in the TCAM that match the set of packets described by the ACLs in the match criteria of the route map. These TCAM entries point at adjacencies that either perform the necessary output actions or forward the packet to software if either hardware does not support the action or its resources are exhausted.

If the route-map specifies a **set interface …** action, packets that match the **match** statement are routed in software. Similarly, if the route-map specifies a **set default interface…** action and there is no matching IP route for the packet, the packet is routed in software.

Note The scale of hardware-based PBR is determined by TCAM size and the time required for the CPU to flatten the ACL before programming into hardware. The latter will noticeably increase if a PBR policy requires a considerable number of class-maps. For example, a PBR policy of 1,200 class-maps may require 60-90 minutes of "flatten" time before programming into hardware. This process may repeat if an adjacency change requires PBR reprogramming.

PBR Flow Switching

Note Supervisor Engine 6-E, Supervisor Engine 6L-E, Catalyst 4900M, and Catalyst 4948E do not implement PBR using flow switching.

The Catalyst 4500 switching engine supports matching a set next-hop route-map action with a packet on a permit ACL. All other route-map actions, as well as matches of deny ACLs, are supported by a flow switching model. In this model, the first packet on a flow that matches a route-map is delivered to the software for forwarding. Software determines the correct destination for the packet and installs an entry into the TCAM so that future packets on that flow are switched in hardware. The Catalyst 4500 switching engine supports a maximum of 4096 flows.

Using Policy-Based Routing

You can enable PBR to change the routing path of certain packets from the default path that would be chosen by IP routing. For example, you can use PBR to provide the following functionality:

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

- Routing based on dedicated links

Some applications or traffic can benefit from source-specific routing; for example, you can 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.

Note PBR configuration is only allowed on interfaces belonging to the global routing table. PBR is not supported on interfaces that belong to VRFs.

Policy-Based Routing Configuration Tasks

To configure PBR, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional. For configuration examples, see the "Policy-Based Routing Configuration Examples" section on page 35-8.

- Enabling PBR, page 35-6 (Required)
- Enabling Local PBR, page 35-8 (Optional)

Enabling PBR

To enable PBR, you must create a route map that specifies the match criteria and the resulting action if all of the match clauses are met. Then you must apply that route-map on a particular interface. All packets arriving on the specified interface matching the match clauses are subject to PBR.

To enable PBR on an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# route-map map-tag [permit | deny] [sequence-number]`	Defines a route map to control where packets are sent. This command puts the switch into route-map configuration mode.
Step 2	`Switch(config-route-map)# match ip address {access-list-number | name} [...access-list-number | name]`	Specifies the match criteria. The match criteria take the form of one or more Standard or Extended IP access-lists. The access-lists can specify the source and destination IP addresses, protocol types, and port numbers. See Chapter 47, "Configuring Network Security with ACLs" for more information on Standard and Extended IP access-lists.

Step 3	`Switch(config-route-map)# set ip next-hop ip-address [... ip-address]`
	
	`Or`
		Specifies the next-hop IP address to which matching packets are sent. The next-hop IP address specified here must belong to a subnet that is directly connected to this switch.
		
		If more than one next-hop IP address is specified, the first usable next-hop is chosen for routing matching packets. If the next-hop is (or becomes) unavailable for some reason, the next one in the list is chosen.

	Command	Purpose

Step 4	`Switch(config-route-map)# set interface interface-type interface-number [... type number]`
	
	`Or`
		Specifies the output interface from which the packet will be sent. This action specifies that the packet is forwarded out of the local interface. The interface must be a Layer 3 interface (not a switchport).
		
		Packets are forwarded on the specified interface only if one of the following conditions is met:
		• The destination IP address in the packet lies within the IP subnet to which the specified interface belongs.
		• The destination IP address in the packet is reachable through the specified interface (as per the IP routing table).
		
		If the destination IP address on the packet does not meet either of these conditions, the packet is dropped. This action forces matching packets to be switched in software.k

Step 5	`Switch(config-route-map)# set ip default next-hop ip-address [... ip-address]`
	
	`Or`
		Sets next hop to which to route the packet if there is no explicit route for the destination IP address in the packet. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by way of the routing table. If no match is found, the packet is forwarded to the specified next hop.

Step 6	`Switch(config-route-map)# set default interface interface-type interface-number [...type ...number]`	Specifies the output interface from which the packet will be sent if there is no explicit route for this destination. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by using the routing table. If no match is found, the packet is forwarded to the specified output interface. Packets are forwarded on the specified interface only if one of the following conditions is met: • The destination IP address in the packet lies within the IP subnet to which the specified interface belongs. • The destination IP address in the packet is reachable through the specified interface (as per the IP routing table). If the destination IP address on the packet does not meet either of these conditions, the packet is dropped. This action forces matching packets to be switched in software.
Step 7	`Switch(config-route-map)# interface interface-type interface-number`	Specifies the interface. This command puts the switch into interface configuration mode.
Step 8	`Switch(config-if)# ip policy route-map map-tag`	Identifies the route map to use for PBR. One interface can only have one route map tag, but you can have multiple route map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If no match exists, packets are routed as usual.

Use the **set** commands with each other. These commands are evaluated in the order shown in Step 3 in the previous task table. A usable next hop implies an interface. Once the local switch finds a next hop and a usable interface, it routes the packet.

Enabling Local PBR

Packets that are generated by the switch are not normally policy-routed. To enable local PBR for such packets, indicate which route map the switch should use by entering this command:

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

All packets originating on the switch are then subject to local PBR.

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

Unsupported Commands

The following PBR commands in config-route-map mode are in the CLI but not supported in Cisco IOS for the Catalyst 4500 series switches. If you attempt to use these commands, an error message displays:

- **match-length**
- **set ip qos**
- **set ip tos**
- **set ip precedence**

Policy-Based Routing Configuration Examples

The following sections provide PBR configuration examples:

- Equal Access, page 35-8
- Differing Next Hops, page 35-9
- Deny ACE, page 35-9

For information on how to configure policy-based routing, see the section "Policy-Based Routing Configuration Tasks" in this chapter.

Equal Access

The following example provides two sources with equal access to two different service providers. Packets arriving on interface fastethernet 3/1 from the source 1.1.1.1 are sent to the switch at 6.6.6.6 if the switch has no explicit route for the destination of the packet. Packets arriving from the source 2.2.2.2 are sent to the switch at 7.7.7.7 if the switch has no explicit route for the destination of the packet. All other packets for which the switch has no explicit route to the destination are discarded.

```
Switch (config)# access-list 1 permit ip 1.1.1.1
access-list 1 permit ip 1.1.1.1
!
interface fastethernet 3/1
 ip policy route-map equal-access
!

route-map equal-access permit 10
 match ip address 1
 set ip default next-hop 6.6.6.6
route-map equal-access permit 20
 match ip address 2
 set ip default next-hop 7.7.7.7
route-map equal-access permit 30
 set default interface null0
```

Note If the packets you want to drop do not match either of the first two route-map clauses, then change | **set default interface null0** to **set interface null0**.

Differing Next Hops

The following example illustrates how to route traffic from different sources to different places (next hops). Packets arriving from source 1.1.1.1 are sent to the next hop at 3.3.3.3; packets arriving from source 2.2.2.2 are sent to the next hop at 3.3.3.5.

```
access-list 1 permit ip 1.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface fastethernet 3/1
 ip policy route-map Texas
!
route-map Texas permit 10
 match ip address 1
 set ip next-hop 3.3.3.3
!
route-map Texas permit 20
 match ip address 2
 set ip next-hop 3.3.3.5
```

Deny ACE

The following example illustrates how to stop processing a given route map sequence, and to jump to the next sequence. Packets arriving from source 1.1.1.1 skip sequence 10 and jump to sequence 20. All other packets from subnet 1.1.1.0 follow the set statement in sequence 10.

```
access-list 1 deny ip 1.1.1.1
access-list 1 permit ip 1.1.1.0 0.0.0.255
access-list 2 permit ip 1.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface fastethernet 3/1
ip policy route-map Texas
!
```

```
route-map Texas permit 10
match ip address 1
set ip next-hop 3.3.3.3
!
route-map Texas permit 20
match ip address 2
set ip next-hop 3.3.3.5
```