



Policy-based routing

- [Feature history for policy-based routing, on page 1](#)
- [Policy-based routing, on page 2](#)
- [Policy-based routing statements, on page 2](#)
- [Route map match criteria and set actions, on page 3](#)
- [Restrictions for policy based routing, on page 4](#)
- [Guidelines for policy based routing, on page 4](#)
- [Configure policy-based routing, on page 5](#)

Feature history for policy-based routing

This table provides release and related information for the features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	Policy-Based Routing: Use policy-based routing (PBR) to configure a defined policy for traffic flows. By using PBR, you can have more control over routing by reducing the reliance on routes derived from routing protocols.	Cisco C9350 Series Smart Switches
Cisco IOS XE 17.18.2	Policy-Based Routing: Use policy-based routing (PBR) to configure a defined policy for traffic flows. By using PBR, you can have more control over routing by reducing the reliance on routes derived from routing protocols.	Cisco C9610 Series Smart Switches

Policy-based routing

A policy-based routing (PBR) is a network routing mechanism that

- enables administrators to define routing decisions based on factors other than the destination IP address,
- uses route maps and access control lists (ACLs) to match specific types of traffic, and
- allows for customized traffic paths according to organizational or application needs.

Policy-Based Routing (PBR) allows you to define routing policies that allow or deny paths based on specific criteria, such as:

- The identity of an end system: This typically refers to attributes like the source IP address, source subnet, or sometimes the source MAC address (though most PBR configurations use IP-based matching).
- The application: This is usually identified by well-known destination or source port numbers associated with specific applications or services (for example, HTTP traffic on TCP port 80, or DNS on UDP port 53).
- The protocol: You can match traffic based on the protocol type, such as TCP, UDP, or ICMP.

By using these criteria in route-map match statements, PBR can direct traffic along different paths based on the actual content or origin of the packets, rather than solely relying on the destination IP address.

For example, you can transfer stock records to a corporate office on a high-bandwidth, high-cost link for a short time while transmitting routine application data, such as email, over a low-bandwidth, low-cost link.

PBR classifies incoming traffic using access control lists (ACLs) and then directs it through a different path. PBR applies to incoming packets. All packets received on an interface with PBR enabled are passed through route maps. Based on the criteria defined in the route maps, packets are forwarded (routed) to the appropriate next hop.

Policy-based routing statements

A route map statement is a configuration element in network devices that

- specifies match conditions (such as access control lists, or ACLs)
- applies permit or deny actions to matching packets, and
- determines subsequent packet processing based on these matches.

Permit statement

A route map entry that, when its match criteria are met, applies the configured set actions to the packet.

A match command can match on multiple ACLs. A route map statement can contain multiple match commands. A logical OR function is performed across all match commands to reach a permit or deny decision.

Route map statements can have multiple match commands:

- Within a single match command (e.g., **match ip address acl1 acl2**), the logic is OR - a match occurs if any criterion is met.

- Between multiple match commands(e.g., **match ip address acl3**), the logic is AND - all conditions must match for the statement to be considered a match.

A packet is permitted if it is permitted by match acl1 OR acl2 OR acl3.

Deny statement

A route map entry that, when its match criteria are met, causes the device to skip policy actions and continue with default routing or evaluate the next route map statement.

When a packet matches a **deny** statement in a PBR route-map, the action depends on both the route-map decision and the outcome of the ACL check:

- If the route-map statement is "deny" and the ACL matches ("permit"): PBR processing stops. The packet is not subject to any PBR action and is forwarded using the default IP routing table.
- If the route-map statement is "deny" and the ACL does not match ("deny"): The device moves to the next route-map statement (with the next higher sequence number) and continues processing.

Route map match criteria and set actions

A route map match criterion or set action is a configuration mechanism that

- enables conditional packet or route selection based on various parameters (such as source/destination addresses or protocol types),
- allows modification of routes, such as updating next-hop address or setting metrics, and
- dictates the sequence of route map processing, determining outcomes for matched and unmatched packets.

Match criterion is a condition in a route map that specifies which packets or routes will be acted upon.

Set action is an instruction in a route map that modifies routes or packets matching the criteria.

When configuring a route map, you can use:

- Standard IP ACLs to specify match criteria based on source addresses.
- Extended IP ACLs to specify match criteria based on application, protocol type, source and destination addresses, or end station.

If no match is found in the route map, normal destination-based routing occurs.

- An implicit deny exists at the end of match statements; unmatched packets are not processed further by that route map.
- Set clauses define what modifications (such as next hop or metric changes) are applied to matched traffic.
- Match clauses are evaluated only on incoming packets or routes.

Restrictions for policy based routing

To specify imitations and restrictions for policy based routing, enabling administrators to avoid unsupported configurations and ensure stable network operation.

The following are restrictions for policy-based routing (PBR):

- PBR configuration to forward traffic into GRE tunnels is not supported. This applies to PBR applied on any interface and forwarding traffic into a GRE tunnel (by means of PBR next-hop or set interface).
- PBR is not supported on directly GRE tunnel interface.
- PBR does not policy-route fragmented traffic. Fragmented traffic follows a normal routing path.
- PBR and Network Address Translation (NAT) are not supported on the same interface. PBR and NAT work together only if they are configured on different interfaces
- Web Cache Communication Protocol (WCCP) and PBR are mutually exclusive on a switch interface. Enabling WCCP when PBR is enabled on an interface is not supported, and vice versa.
- PBR does not support TOS, DSCP, or IP Precedence-based routing.
- PBR does not support set interface, set default next-hop, or set default interface commands.
- The ip next-hop recursive and ip next-hop verify availability features are not supported; ensure the next-hop is directly connected.

Guidelines for policy based routing

To provide a comprehensive list of policy-based routing (PBR) guidelines.

The following are guidelines for policy-based routing (PBR):

- PBR applies only to unicast traffic; it does not policy-route multicast traffic.
- You can enable PBR on a routed port or a Switched Virtual Interface (SVI).
- You can apply a policy route map to an EtherChannel port channel in Layer 3 mode. However, you cannot apply a policy route map to a physical interface that is a member of an EtherChannel. Attempting to do so rejects the command.
- When a policy route map is applied to a physical interface, that interface cannot become a member of an EtherChannel.
- When configuring match criteria in a route map, do not match ACLs that permit packets destined for a local address.
- The number of hardware entries used by PBR depends on the route map itself, the ACLs used, and the order of the ACLs and route map entries.
- Policy maps without set actions are supported; matching packets are routed normally.
- Policy maps without match clauses are supported; set actions are applied to all packets.

Configure policy-based routing

By default, PBR is disabled on the switch. To enable PBR, you must create a route map that specifies the match criteria and the resulting action. Then, you must enable PBR for that route map on an interface.

All packets arriving on the specified interface matching the match clauses are subject to PBR.

Follow these steps to configure policy-based routing:

Procedure

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**] *sequence number*

Example:

```
Device(config)# route-map pbr-map permit
```

Defines route maps that are used to control where packets are output, and enters route-map configuration mode.

- *map-tag* - A meaningful name for the route map. The **ip policy route-map** interface configuration command uses this name to reference the route map. Multiple route-map statements with the same map tag define a single route map.
- (Optional) **permit** - If **permit** is specified and the match criteria are met for this route map, the route is policy routed as defined by the set actions.
- *sequence number* - The sequence number shows the position of the route-map statement in the given route map.

Step 4

match ip address {*access-list-number* | *access-list-name*} [*access-list-number*] ...*access-list-name*]

Example:

```
Device(config- route-map)# match ip address 110 140
```

Matches the source and destination IP addresses that are permitted by one or more standard or extended access lists. ACLs can match on more than one source and destination IP address.

If you do not specify a **match** command, the route map is applicable to all packets.

Step 5

set ip next-hop *ip-address* [...*ip-address*]

Example:

```
Device(config-route-map)# set ip next-hop 10.1.6.2
```

Specifies the action to be taken on the packets that match the criteria. Sets next hop to which to route the packet (the next hop must be adjacent).

Step 6 **exit****Example:**

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

Returns to global configuration mode.

Step 7 **interface** *interface-id***Example:**

```
Device(config)# interface gigabitethernet 1/0/1
```

Enters interface configuration mode and specifies the interface to be configured.

Step 8 **ip policy route-map** *map-tag***Example:**

```
Device(config)# ip policy route-map pbr-map
```

Enables PBR on a Layer 3 interface, and identify the routemap to use. You can configure only one route map on an interface. However, you can have multiple routemap entries with different sequence numbers. These entries are evaluated in the order of sequence number until the first match. If there is no match, packets are routed as usual.

Step 9 **ip route cache policy****Example:**

```
Device(config)# ip route-cache policy
```

Enables PBR on a Layer 3 interface, and identify the routemap to use. You can configure only one route map on an interface. However, you can have multiple routemap entries with different sequence numbers. These entries are evaluated in the order of sequence number until the first match. If there is no match, packets are routed as usual.

Step 10 **exit****Example:**

```
Device(config)# ip route-cache policy
```

Enables PBR on a Layer 3 interface, and identify the routemap to use. You can configure only one route map on an interface. However, you can have multiple routemap entries with different sequence numbers. These entries are evaluated in the order of sequence number until the first match. If there is no match, packets are routed as usual.

Step 11 **exit****Example:**

```
Device(config)# exit
```

Returns to global configuration mode.

Step 12 **end****Example:**

```
Device(config)# end
```

Returnsto privileged EXEC mode.

Step 13 **end**

Example:

```
Device(config)# end
```

Returns to privileged EXEC mode.

Step 14 **end**

Example:

```
Device(config)# end
```

Returns to privileged EXEC mode.

Step 15 **show ip policy**

Example:

```
Device(config)# show ip policy
```

(Optional) Displays policy routemaps attached to the interface.
