NetFlow Policy Routing

Cisco IOS Release 12.0(3)T

October 11, 1999

Feature Summary

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

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

Cisco developed three technologies in Cisco IOS software:

- CEF, which looks at a Forwarding Information Base (FIB) instead of a routing table when switching packets.
- dCEF, which addresses the scalability and maintenance problems of a demand caching scheme.
- NetFlow, which provides for accounting, access list flow checking, and traffic monitoring.

NetFlow policy routing leverages these technologies.

Benefits

- Policy routing with CEF and Netflow takes advantage of the new switching services.
- Now that policy routing is integrated into CEF, policy routing can be deployed on a wide scale and on high-speed interfaces.

Restrictions

- Distributed FIB-based policy-routing is only available on platforms that support dCEF and images that support dCEF.

- The **set ip next-hop verify-availability** command is not supported in dCEF because dCEF does not support the Cisco Discovery Protocol (CDP) database.

Related Documentation

- Cisco IOS Release 12.0 *Network Protocols Configuration Guide, Part 1*, "Configuring IP Routing Protocol-Independent Features" chapter.
- Cisco IOS Release 12.0 *Network Protocols Command Reference, Part 1*, "IP Routing Protocol-Independent Commands" chapter.
- Cisco IOS Release 12.0 *Cisco IOS Switching Services Configuration Guide*
- Cisco IOS Release 12.0 *Cisco IOS Switching Services Command Reference*

Platforms

This feature is supported on the following platforms:

- Cisco 2600 series
- Cisco 3600 series
- Cisco 4000 and Cisco 4000-M series
- Cisco 7200 series
- Cisco 7500 series
- Cisco 5800
- C5000RSM
- UBR7200

Prerequisites

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

- CEF, dCEF, or NetFlow
- Policy routing

To configure CEF, dCEF, or NetFlow, refer to the appropriate chapter of the *Cisco IOS Switching Services Configuration Guide*.

To configure policy routing, refer to the "Configuring IP Routing Protocol-Independent Features" chapter of the *Network Protocols Configuration Guide, Part 1*.

Supported MIBs and RFCs

No new MIBs or RFCs are defined for this feature.

Configuration Task

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

There is one new, optional configuration command (**set ip next-hop verify-availability**). This command has the following restrictions:

- It can cause some performance degradation.
- CDP must be configured on the interface.
- The direct next hop must be a Cisco device with CDP enabled.
- The command is not available in dCEF, due to the dependency of the CDP neighbor database.

It is assumed that policy routing itself is already configured.

If the router is policy routing packets to the next hop and the next hop happens to be down, the router will try unsuccessfully to use Address Resolution Protocol (ARP) for the next hop (which is down). This behavior will continue forever.

To prevent this situation, you can configure the router to first verify that the next hop(s) of the route map is the router's CDP neighbor(s) before routing to that next hop.

This task is optional because some media or encapsulations do not support CDP, or it may not be a Cisco device that is sending the router traffic.

To configure the router to verify that the next hop is a CDP neighbor before the router tries to policy route to it, use the following command in route-map configuration mode:

Command	Purpose
set ip next-hop verify-availability	Causes the router to confirm that the next hop(s) of the route map is a CDP neighbor(s) of the router.

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

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

If you want to selectively verify availability of only some next hops, you can configure different route-map entries (under the same route-map name) with different criteria (using access list matching or packet size matching), and use the **set ip next-hop verify-availability** command selectively.

Monitor NetFlow Policy Routing

Typically, you would use existing policy routing and NetFlow **show** commands to monitor these features. For more information on these **show** commands, refer to the policy routing and NetFlow documentation.

To display the route map Inter Processor Communication (IPC) message statistics in the RP or VIP, use the following command in EXEC mode:

Command	Purpose
show route-map ipc	Displays the route map IPC message statistics in the RP or VIP.

Configuration Example

The following example configures CEF and Policy Routing with NetFlow access list acceleration. The route is configured to verify that next hop 50.0.0.8 of the route map named *example1* is a CDP neighbor before the router tries to policy route to it.

If the first packet is being policy routed via route map *example1* sequence 10, the subsequent packets of the same flow always take the same route map *example1* sequence 10, not route map *example1* sequence 20, because they all match or pass access list 1 check.

Therefore, policy routing can be accelerated by Netflow access-list matching, which bypasses the access list check for subsequent packets in the flow.

```
ip cef
interface ethernet0/0/1
 ip route-cache flow
 ip policy route-map example1
route-map example1 permit 10
 match ip address 1
 set ip precedence priority
 set ip next-hop 50.0.0.8
 set ip next-hop verify-availability
route-map example1 permit 20
 match ip address 101
 set interface Ethernet0/0/3
 set ip tos max-throughput
```

Command Reference

This section documents the following new commands. All other commands used with this feature are documented in the Cisco IOS 12.0 documentation set.

- **set ip next-hop verify-availability**
- **show route-map ipc**

set ip next-hop verify-availability

To configure policy routing to verify if the next hop(s) of a route map is a CDP neighbor(s) before policy routing to that next hop, use the **set ip next-hop verify-availability** route-map configuration command.

set ip next-hop verify-availability

Syntax Description

This command has no arguments or keywords.

Command Mode

Route-map configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 12.0(3)T.

One example of when you might configure this command is if you have some traffic traveling via a satellite to a next hop. It might be prudent to verify that the next hop is reachable before trying to policy route to it.

This command has the following restrictions:

- It causes some performance degradation.
- CDP must be configured on the interface.
- The next hop must be a Cisco device with CDP enabled.
- It is supported in process switching and CEF policy routing, but not available in DCEF, due to the dependency of the CDP neighbor database.

If the router is policy routing packets to the next hop and the next hop happens to be down, the router will try unsuccessfully to use Address Resolution Protocol (ARP) for the next hop (which is down). This behavior will continue forever.

To prevent this situation, use this command to configure the router to first verify that the next hop(s) of the route map is the router's CDP neighbor(s) before routing to that next hop.

This command is optional because some media or encapsulations do not support CDP, or it may not be a Cisco device that is sending the router traffic.

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

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

If you want to selectively verify availability of only some next hops, you can configure different route-map entries (under the same route-map name) with different criteria (using access list matching or packet size matching), and use the **set ip next-hop verify-availability** command selectively.

Example

The following example configures CEF and Policy Routing. It also configures policy routing to verify that next hop 50.0.0.8 of route map *example1* is a CDP neighbor before the router tries to policy route to it.

If the first packet is being policy routed via route map *example1* sequence 10, the subsequent packets of the same flow always take the same route map *example1* sequence 10, not route map *example1* sequence 20, because they all match or pass access list 1 check.

```
ip cef
interface ethernet0/0/1
 ip policy route-map example1
route-map example1 permit 10
 match ip address 1
 set ip precedence priority
 set ip next-hop 50.0.0.8
 set ip next-hop verify-availability
route-map example1 permit 20
 match ip address 101
 set interface Ethernet0/0/3
 set ip tos max-throughput
```

show route-map ipc

To display counts of the one-way route map IPC messages sent from the RP to the VIP when NetFlow policy routing is configured, use the **show route-map ipc** EXEC command.

show route-map ipc

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 12.0(3)T.

This command displays the counts of one-way route map IPC messages from the RP to the VIP when NetFlow policy routing is configured. If you execute this command on the RP, the messages are shown as "Sent." If you execute this command on the VIP console, the IPC messages are shown as "Received."

Examples

The following is sample output of the **show route-map ipc** command when it is executed on the RP:

```
Router# show route-map ipc

Route-map RP IPC Config Updates Sent
Name: 4
Match access-list: 2
Match length: 0
Set precedence: 1
Set tos: 0
Set nexthop: 4
Set interface: 0
Set default nexthop: 0
Set default interface: 1
Clean all: 2
```

The following is sample output of the **show route-map ipc** command when it is executed on the VIP:

```
VIP-Slot0# show route-map ipc

Route-map LC IPC Config Updates Received
Name: 4
Match access-list: 2
Match length: 0
Set precedence: 1
Set tos: 0
Set nexthop: 4
Set interface: 0
Set default nexthop: 0
Set default interface: 1
Clean all: 2
```

Table 1 describes the significant fields in the first display.

Table 1 show route-map ipc Field Descriptions

Field	Description
Route-map RP IPC Config Updates Sent	IPC messages are being sent from the RP to the VIP.
Name:	Number of IPC messages sent about the name of the route map.
Match access-list:	Number of IPC messages sent about the access list.
Match length	Number of IPC messages sent about the length to match.
Set precedence:	Number of IPC messages sent about the precedence.
Set tos:	Number of IPC messages sent about the type of service (TOS).
Set nexthop:	Number of IPC messages sent about the next hop.
Set interface:	Number of IPC messages sent about the interface.
Set default nexthop:	Number of IPC messages sent about the default next hop.
Set default interface:	Number of IPC messages sent about the default interface.
Clean all:	Number of IPC messages sent about clearing the policy routing configuration from the VIP. When DCEF is disabled and reenabled, the configuration related to policy routing must be removed (cleaned) from the VIP before the new information is downloaded from the RP to the VIP.

Debug Commands

This section describes the following new and revised **debug** commands:

- **debug ip policy**
- **debug route-map ipc**

debug ip policy

Use the **debug ip policy** EXEC command to display IP policy routing packet activity. The **no** form of this command disables debugging output.The **no** form of this command disables debugging output.

[**no**] **debug ip policy** [*access-list-name*]

Syntax Description

access-list-name	(Optional) Name of the access list. Displays packets permitted by the access-list that are policy routed in process level, CEF, and DCEF (with NetFlow enabled or disabled). If no access list is specified, information about all policy-matched and policy-routed packets is displayed.

Usage Guidelines

After you configure IP policy routing with the **ip policy** and **route-map** commands, use the **debug ip policy** command to ensure that the IP policy is configured correctly.

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

The **debug ip policy** command helps you determine what policy routing is doing. It displays information about whether a packet matches the criteria, and if so, the resulting routing information for the packet.

Caution Because the **debug ip policy** command generates a substantial amount of output, use it only when traffic on the IP network is low, so other activity on the system is not adversely affected.

Example

The following is sample output of the **debug ip policy** command:

```
Router# debug ip policy 3

IP: s=30.0.0.1 (Ethernet0/0/1), d=40.0.0.7, len 100,FIB flow policy match
IP: s=30.0.0.1 (Ethernet0/0/1), d=40.0.0.7, g=10.0.0.8, len 100, FIB policy routed
```

Table 2 describes the fields in the display.

Table 2 debug ip policy Field Descriptions

Field	Description
IP: s=	IP source address and interface of the packet being routed.
d=	IP destination address of the packet being routed.
len	Length of the packet
g=	IP gateway address of the packet being routed.

debug route-map ipc

To display a summary of the one-way IPC messages set from the RP to the VIP about NetFlow policy routing when DCEF is enabled, use the **debug route-map ipc** EXEC command. The **no** form of this command disables debugging output.

[**no**] **debug route-map ipc**

Usage Guidelines

This command first appeared in Cisco IOS Release 12.0(3)T.

This command is especially helpful for policy routing with DCEF switching.

This command displays a summary of one-way IPC messages from the RP to the VIP about NetFlow policy routing. If you execute this command on the RP, the messages are shown as "Sent." If you execute this command on the VIP console, the IPC messages are shown as "Received."

Examples

The following is sample output of the **debug route-map ipc** command executed at the RP:

```
Router# debug route-map ipc

Routemap related IPC debugging is on

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip cef distributed
Router(config)#^Z
Router#

RM-IPC: Clean routemap config in slot 0
RM-IPC: Sent clean-all-routemaps; len 12
RM-IPC: Download all policy-routing related routemap config to slot 0
RM-IPC: Sent add routemap example1(seq:10); n_len 5; len 17
RM-IPC: Sent add acl 1 of routemap example1(seq:10); len 21
RM-IPC: Sent add min 10 max 300 of routemap example1(seq:10); len 24
RM-IPC: Sent add preced 1 of routemap example1(seq:10); len 17
RM-IPC: Sent add tos 4 of routemap example1(seq:10); len 17
RM-IPC: Sent add nexthop 50.0.0.8 of routemap example1(seq:10); len 20
RM-IPC: Sent add default nexthop 50.0.0.9 of routemap example1(seq:10); len 20
RM-IPC: Sent add interface Ethernet0/0/3(5) of routemap example1(seq:10); len 20
RM-IPC: Sent add default interface Ethernet0/0/2(4) of routemap example1(seq:10); len
20
```

The following is sample output of the **debug route-map ipc** command executed at the VIP:

```
VIP-Slot0# debug route-map ipc

Routemap related IPC debugging is on

VIP-Slot0#
RM-IPC: Rcvd clean-all-routemaps; len 12
RM-IPC: Rcvd add routemap example1(seq:10); n_len 5; len 17
RM-IPC: Rcvd add acl 1 of routemap example1(seq:10); len 21
RM-IPC: Rcvd add min 10 max 300 of routemap example1(seq:10); len 24
RM-IPC: Rcvd add preced 1 of routemap example1(seq:10); len 17
RM-IPC: Rcvd add tos 4 of routemap example1(seq:10); len 17
RP-IPC: Rcvd add nexthop 50.0.0.8 of routemap example1(seq:10); len 20
RP-IPC: Rcvd add default nexthop 50.0.0.9 of routemap example1(seq:10); len 20
RM-IPC: Rcvd add interface Ethernet0/3 of routemap tes; len 20
RM-IPC: Rcvd add default interface Ethernet0/2 of routemap example1(seq:10); len 20
```