CHAPTER 13

Managing the Unicast RIB and FIB

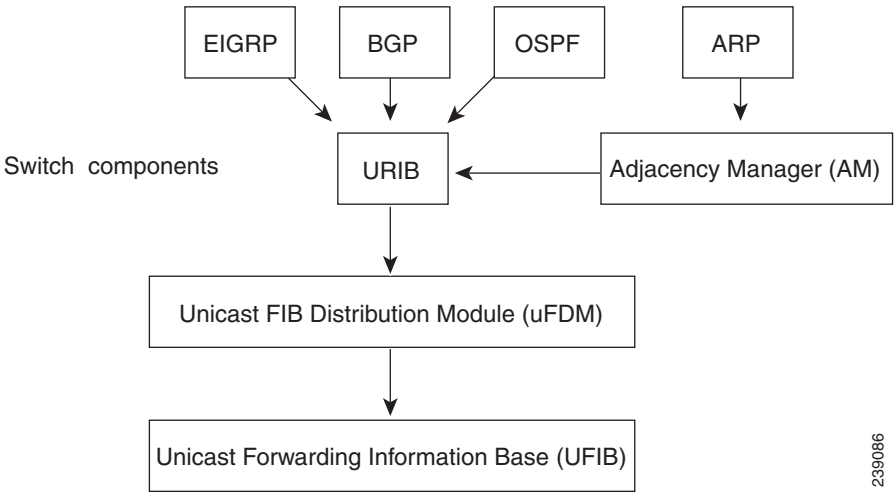
This chapter describes how to manage routes in the unicast Routing Information Base (RIB) and the Forwarding Information Base (FIB) on the Cisco NX-OS switch.

This chapter includes the following sections:

Information About the Unicast RIB and FIB

The unicast RIB (IPv4 RIB and IPv6 RIB) and FIB are part of the Cisco NX-OS forwarding architecture, as shown in Figure 13-1.

Figure 13-1 Cisco NX-OS Forwarding Architecture



The unicast RIB exists on the active supervisor. It maintains the routing table with directly connected routes, static routes, and routes learned from dynamic unicast routing protocols. The unicast RIB also collects adjacency information from sources such as the Address Resolution Protocol (ARP). The unicast RIB determines the best next hop for a given route and populates the unicast forwarding information bases (FIBs) on the modules by using the services of the unicast FIB distribution module (FDM).

Each dynamic routing protocol must update the unicast RIB for any route that has timed out. The unicast RIB then deletes that route and recalculates the best next hop for that route (if an alternate path is available).

This section includes the following topics:

Layer 3 Consistency Checker

In rare instances, an inconsistency can occur between the unicast RIB and the FIB on each module. In Cisco NX-OS Release 4.0(3) and later releases, Cisco NX-OS supports the Layer 3 consistency checker. This feature detects inconsistencies between the unicast IPv4 RIB on the supervisor module and the FIB on each interface module. Inconsistencies include the following:

- Missing prefix
- Extra prefix
- Wrong next-hop address
- Incorrect Layer 2 rewrite string in the ARP or neighbor discovery (ND) cache

The Layer 3 consistency checker compares the FIB entries to the latest adjacency information from the Adjacency Manager (AM) and logs any inconsistencies. The consistency checker then compares the unicast RIB prefixes to the module FIB and logs any inconsistencies. See the "Triggering the Layer 3 Consistency Checker" section on page 13-9.

You can then manually clear any inconsistencies. See the "Clearing Forwarding Information in the FIB" section on page 13-10.

FIB Tables

The hardware provides two tables, a TCAM table and a Hash table. The TCAM table is shared between the longest prefix match (LPM) route and the /32 unicast route. The Hash table is shared between the /32 unicast entries and the multicast entries. Each table has approximately 8000 routes.

Dynamic TCAM Allocation

Cisco NX-OS divides the FIB to support multiple address families.

Table 13-1 describes the default FIB TCAM allocation.

Table 13-1 Default FIB TCAM Allocation

Region	Default # Routes	URPF Enabled/Disabled Globally	Verified Maximum
IPv4 hosts	16,000	-	16,000
IPv6 hosts	8,000	-	8,000
IPv4 routes (LPM)	-	8,192/16,000	8,192/16,000
IPv6 routes (LPM less than or equal to 64 bits	-	4000/8000	4000/8000
IPv6 routes (LPM greater than 64 bits and less than or equal to 127 bits	-	128/256	128/256
IPv4 multicast routes	4,000	-	8,000

Note The Cisco Nexus 3064PQ offers half the scalability listed.

Note IPv6 will use up to 2 entries for every route in the hardware.

Virtualization Support

The Unicast RIB and FIB support Virtual Routing and Forwarding instances (VRFs). For more information, see Chapter 12, "Configuring Layer 3 Virtualization."

Licensing Requirements for the Unicast RIB and FIB

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	The unicast RIB and FIB require no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations

Unicast RIB and FIB have the following configuration guidelines and limitations:

- You must install the Scalable Services license and configure the higher shared memory sizes to enable the higher FIB sizes.

- Dynamic TCAM allocation is enabled by default.
- IP protocol information is needed for correct resolution of ECMP next hop. In the absense of this input the user IP Protocol UDP is assumed and the resolution computed.
- Routing hash configuration is only supported on unicast packets. The following scenarios are not supported:
  - Unknown unicast Layer 2 forwarding over a port channel.
  - Multicast Layer 2 and Layer 3 forwarding over a port channel.
  - Broadcast Layer 2 forwarding over a port channel.

Note Member interfaces of a port channel that are not active are not included in the port channel load balancing hash computation.

Managing the Unicast RIB and FIB

This section includes the following topics:

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Displaying Module FIB Information

You can display the FIB information on a switch.

DETAILED STEPS

To display the FIB information on a switch, use the following commands in any mode:

Command	Purpose
show ip fib adjacency **Example:** switch# show ip fib adjacency	Displays the adjacency information for FIB.
show forwarding ipv4 | **ipv6**} **adjacency** **Example:** switch# show forwarding ipv4 adjacency	Displays the adjacency information for IPv4 or IPv6.
show ip fib interfaces **Example:** switch# show ip fib interfaces	Displays the FIB interface information for IPv4.
show ip fib route **Example:** switch# show ip fib route	Displays the route table for IPv4.
show forwarding {**ipv4** | **ipv6**} **route** **Example:** switch# show forwarding ipv4 route	Displays the route table for IPv4 or IPv6.

This example shows the FIB contents on a switch:

```
switch# show ip fib route

IPv4 routes for table default/base

------------------+------------------+---------------------
Prefix            | Next-hop         | Interface
------------------+------------------+---------------------
0.0.0.0/32          Drop               Null0
255.255.255.255/32  Receive            sup-eth1
```

Configuring Load Sharing in the Unicast FIB

Dynamic routing protocols, such as Open Shortest Path First (OSPF), support load balancing with equal-cost multipath (ECMP). The routing protocol determines its best routes based on the metrics configured for the protocol and installs up to the protocol-configured maximum paths in the unicast RIB. The unicast RIB compares the administrative distances of all routing protocol paths in the RIB and selects a best path set from all of the path sets installed by the routing protocols. The unicast RIB installs this best path set into the FIB for use by the forwarding plane.

The forwarding plane uses a load-sharing algorithm to select one of the installed paths in the FIB to use for a given data packet.

You can globally configure the following load-sharing settings:

- load-share mode—Selects the best path based on the destination address and port or the source and the destination address and port.
- Universal ID—Sets the random seed for the hash algorithm. You do not need to configure the Universal ID. Cisco NX-OS chooses the Universal ID if you do not configure it.

Note Load sharing uses the same path for all packets in a given flow. A flow is defined by the load-sharing method that you configure. For example, if you configure source-destination load sharing, then all packets with the same source IP address and destination IP address pair follow the same path.

To configure the unicast FIB load-sharing algorithm, use the following command in global configuration mode:

Command	Purpose
ip load-sharing address {**destination port destination** | **source-destination** [**port source-destination**]} [**universal-id** *seed*] **Example:** `switch(config)# ip load-sharing address source-destination`	Configures the unicast FIB load-sharing algorithm for data traffic. The *universal-id* range is from 1 to 4294967295.

To avoid ECMP polarization in a multi-tier ECMP session you must configure a different ECMP hash-offset on each tier. To confiugre an ECMP hash-offset, use the following command in global configuration mode:

Command	Purpose
hardware ecmp hash-offset *number* **Example:** `switch(config)# hardware ecmp hash-offset 5`	Configures the ECMP hash-offset.The range is from 0 to 15. The default value is 0.

To display the unicast FIB load-sharing algorithm, use the following command in any mode:

Command	Purpose
show ip load-sharing **Example:** `switch(config)# show ip load-sharing`	Displays the unicast FIB load-sharing algorithm for data traffic.

To display the route that the unicast RIB and FIB use for a particular source address and destination address, use the following command in any mode:

Command	Purpose
show routing hash *source-addr dest-addr* [**ip-proto** *ip-protocol*] [*source-l4-port dest-l4-port*] [**vrf** *vrf-name*] **Example:** switch# show routing hash 1.1.1.6.5.5 5.3 ip-proto 0x11 10 234	Displays the route that the unicast RIB FIB use for a source and destination address pair. The source address and destination address format is x.x.x.x. The source port and destination port range is from 1 to 65535. The VRF name can be any case-sensitive, alphanumeric string up to 64 characters. The ip-proto option corresponds to the protocol field of the IP header.

This example shows the route selected for a source/destination pair:

```
switch# show routing hash 1.1.1.6.5.5.5.3 ip-proto 0x11 10 234
Load-share parameters used for software forwarding:
load-share mode: address source-destination port source-destination
Universal-id seed: 0xe05e2e85
Invoking pc_ic_ecmp_resolution
Hash for VRF "default"
Hashing to path *Eth1/29%
For route:
5.5.5.0/24 ubest/mbest: 3/0
    *via 2.2.2.1, Eth1/18, [1/0], 00:14:14, static
    *via 3.3.3.1, Eth1/29, [1/0], 00:14:14, static
    *via 4.4.4.1, Eth1/34, [1/0], 00:14:14, static
```

Configuring Per-Packet Load Sharing

You can use per-packet load sharing to evenly distribute data traffic in an IP network over multiple equal-cost connections. Per-packet load sharing allows the router to send successive data packets over paths on a packet-by-packet basis rather than on a per-flow basis.

Note Using per-packet load sharing can result in out-of-order packets. Packets for a given pair of source-destination hosts might take different paths and arrive at the destination out of order. Make sure you understand the implications of out-of-order packets to your network and applications. Per-packet load sharing is not appropriate for all networks. Per-flow load sharing ensures packets always arrive in the order that they were sent.

Per-packet load sharing uses the round-robin method to determine which path each packet takes to the destination. With per-packet load sharing enabled on interfaces, the router sends one packet for destination1 over the first path, the second packet for (the same) destination1 over the second path, and so on. Per-packet load sharing ensures balancing over multiple links.

Use per-packet load sharing to ensure that a path for a single source-destination pair does not get overloaded. If most of the traffic passing through parallel links is for a single pair, per-destination load sharing will overload a single link while other links will have very little traffic. Enabling per-packet load sharing allows you to use alternate paths to the same busy destination.

Note Per-packet load sharing on an interface overrides the global load-sharing configuration.

You configure per-packet load sharing on the input interface. This configuration determines the output interface that Cisco NX-OS chooses for the packet.

For example, if you have ECMP paths on two output interfaces, Cisco NX-OS uses the following load-sharing methods for input packets on Ethernet 1/1:

- Per-packet load sharing if you configure per-packet load sharing on Ethernet 1/1.
- Per-flow load sharing.

The configuration for the other interfaces have no effect on the load-sharing method used for Ethernet 1/1 in this example.

To configure per-packet load sharing, use the following command in interface configuration mode:

Command	Purpose
ip load-sharing per-packet **Example:** `switch(config-if)# ip load-sharing per-packet`	Configures per-packet load sharing on an interface.

Displaying Routing and Adjacency Information

You can display the routing and adjacency information.

To display the routing and adjacency information, use the following commands in any mode:

Command	Purpose
show {**ip** | **ipv6**} **route** [*route-type* | **interface** *int-type number* | **next-hop**] **Example:** `switch# show ip route`	Displays the unicast route table. The *route-type* argument can be a single route prefix, direct, static, or a dynamic route protocol. Use the **?** keyword to see the supported interfaces.
show {**ip** | **ipv6**} **adjacency** [*prefix* | *interface-type number* [**summary**]| **non-best**] [**detail**] [**vrf** *vrf-id*] **Example:** `switch# show ip adjacency`	Displays the adjacency table. The argument ranges are as follows: • *prefix*—Any IPv4 or IPv6 prefix address. • *interface-type number*—Use the **?** keyword to see the supported interfaces. • *vrf-id*—Any case-sensitive, alphanumeric string up to 32 characters.
show {**ip** | **ipv6**} **routing** [*route-type* | **interface** *int-type number* | **next-hop** | **recursive-next-hop** | **summary** | **updated** {**since** | **until**} *time*] **Example:** `switch# show routing summary`	Displays the unicast route table. The *route-type* argument can be a single route prefix, direct, static, or a dynamic route protocol. Use the **?** keyword to see the supported interfaces.

This example displays the unicast route table:

```
switch# show ip route
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
```

```
192.168.0.2/24, ubest/mbest: 1/0, attached
    *via 192.168.0.32, Eth1/5, [0/0], 22:34:09, direct
192.168.0.32/32, ubest/mbest: 1/0, attached
    *via 192.168.0.32, Eth1/5, [0/0], 22:34:09, local
```

This example shows the adjacency information:

```
switch# show ip adjacency

IP Adjacency Table for VRF default
Total number of entries: 2
Address         Age       MAC Address     Pref Source     Interface       Best
10.1.1.1        02:20:54  00e0.b06a.71eb  50   arp        mgmt0           Yes
10.1.1.253      00:06:27  0014.5e0b.81d1  50   arp        mgmt0           Yes
```

Triggering the Layer 3 Consistency Checker

You can manually trigger the Layer 3 consistency checker.

To manually trigger the Layer 3 consistency checker, use the following commands in global configuration mode:

Command	Purpose
test [**ipv4** | **ipv6**] [**unicast**] **forwarding inconsistency** [**vrf** *vrf-name*] [**module** {*slot*| **all**}] **Example:** `switch(config)# test forwarding inconsistency`	Starts a Layer 3 consistency check. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters. The *slot* range is from 1 to 10.

To stop the Layer 3 consistency checker, use the following commands in global configuration mode:

Command	Purpose
test forwarding [**ipv4** | **ipv6**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot*| **all**}] **stop** **Example:** `switch(config)# test forwarding inconsistency stop`	Stops a Layer 3 consistency check. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters. The *slot* range is from 1 to 10.

To display the Layer 3 inconsistencies, use the following commands in any mode:

Command	Purpose
show forwarding [**ipv4** | **ipv6**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot*| **all**}] **Example:** `switch(config)# show forwarding inconsistency`	Displays the results of a Layer 3 consistency check. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters. The *slot* range is from 1 to 10.

Clearing Forwarding Information in the FIB

You can clear one or more entries in the FIB. Clearing a FIB entry does not affect the unicast RIB.

Caution The **clear forwarding** command disrupts forwarding on the switch.

To clear an entry in the FIB, including a Layer 3 inconsistency, use the following command in any mode:

Command	Purpose
clear forwarding {**ipv4** | **ipv6**} **route** {* | *prefix*} [**vrf** *vrf-name*] [**module** {*slot*| **all**}] **Example:** `switch(config)# clear forwarding ipv4 route *`	Clears one or more entries from the FIB. The route options are as follows: • *—All routes. • *prefix*—Any IP or IPv6 prefix. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters. The *slot* range is from 1 to 10.

Estimating Memory Requirements for Routes

You can estimate the memory that a number of routes and next-hop addresses will use.

To estimate the memory requirements for routes, use the following command in any mode:

Command	Purpose
show routing memory estimate routes *num-routes* **next-hops** *num-nexthops* **Example:** `switch# show routing memory estimate routes 1000 next-hops 1`	Displays the memory requirements for routes. The *num-routes* range is from 1000 to 1000000. The *num-nexthops* range is from 1 to 16.

Clearing Routes in the Unicast RIB

You can clear one or more routes from the unicast RIB.

Caution The * keyword is severely disruptive to routing.

To clear one or more entries in the unicast RIB, use the following commands in any mode:

Command	Purpose
clear ip | **ipv6 route** {* | {*route* | *prefix/length*}[*next-hop interface*]} [**vrf** *vrf-name*] **Example:** `switch(config)# clear ip route 10.2.2.2`	Clears one or more routes from both the unicast RIB and all the module FIBs. The route options are as follows: • *—All routes. • *route*—An individual IP or IPv6 route. • *prefix/length*—Any IP or IPv6 prefix. • *next-hop*—The next-hop address • *interface*—The interface to reach the next-hop address. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters.
clear routing [**multicast** | **unicast**] [**ip** | **ipv4** | **ipv6**] {* | {*route* | *prefix/length*}[*next-hop interface*]} [**vrf** *vrf-name*] **Example:** `switch(config)# clear routing ip 10.2.2.2`	Clears one or more routes from the unicast RIB. The route options are as follows: • *—All routes. • *route*—An individual IP or IPv6 route. • *prefix/length*—Any IP or IPv6 prefix. • *next-hop*—The next-hop address • *interface*—The interface to reach the next-hop address. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters.

Verifying the Unicast RIB and FIB Configuration

To display the unicast RIB and FIB configuration information, perform one of the following tasks:

Command	Purpose
show forwarding adjacency	Displays the adjacency table on a module.
show forwarding distribution {**clients** | **fib-state**}	Displays the FIB distribution information.
show forwarding interfaces module *slot*	Displays the FIB information for a module.
show forwarding ipv4 | **ipv6 route**	Displays routes in the FIB.

Command	Purpose
show hardware forwarding dynamic-allocation status	Displays information about the TCAM allocation.
show {ip | ipv6} adjacency	Displays the adjacency table.
show {ip | ipv6} route	Displays IPv4 or IPv6 routes from the unicast RIB.
show routing	Displays routes from the unicast RIB.

Additional References

For additional information related to managing unicast RIB and FIB, see the following sections:

- Related Documents, page 13-12
- Feature History for Unicast RIB and FIB, page 13-12

Related Documents

Related Topic	Document Title
Unicast RIB and FIB CLI commands	*Cisco Nexus 3000 Series Command Reference,*

Feature History for Unicast RIB and FIB

Table 13-2 lists the release history for this feature.

Table 13-2 Feature History for Unicast RIB and FIB

Feature Name	Releases	Feature Information
Unicast RIB and FIB	5.0(3)U1(1)	This feature was introduced.
IPv6	5.0(3)U3(1)	Added support for IPv6.