Configuring the Unicast RIB and FIB

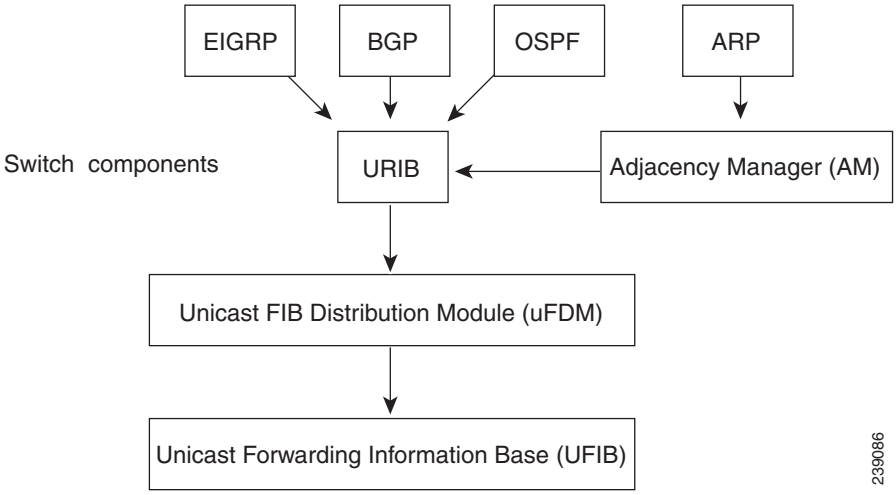
This chapter describes how to manage routes in the unicast Routing Information Base (RIB) and the Forwarding Information Base (FIB) on the Cisco NX-OS switch.

This chapter includes the following sections:

Information About the Unicast RIB and FIB

The unicast RIB (IPv4 RIB) and FIB are part of the Cisco NX-OS forwarding architecture, as shown in Figure 15-1.

Figure 15-1 Cisco NX-OS Forwarding Architecture



Text Part Number:

The unicast RIB 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 base (FIBs) 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. Cisco NX-OS supports the Layer 3 consistency checker. This feature detects inconsistencies between the unicast IPv4 RIB 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 15-8.

You can then manually clear any inconsistencies. See the “Clearing Forwarding Information in the FIB” section on page 15-9.

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 used for the /32 unicast entries. Each table has approximately 8000 routes.

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

Note IPv6 will use up to two 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 14, "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*.

Configuring 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 adjacency **Example:** switch# show forwarding ipv4 adjacency	Displays the adjacency information for IPv4.
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 route **Example:** switch# show forwarding ipv4 route	Displays the route table for IPv4.

This example shows how to display 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 {**source** | **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. A new load sharing option is supported for ECMP, based on the source IP address. The new CLI option looks for the source IP address only on the traffic to load the balance, ignoring the rest of the parameters, for example, the destination IP address and the Layer 4 source port/destination port.

Configuring Hash Offset

To avoid ECMP polarization in a multi-tier ECMP session, you must configure a different ECMP hash-offset on each tier. Starting with Release 6.0(2)U5(1), a new CLI for ECMP hash concatenation is introduced to achieve uniform distribution of the traffic across 16 way ECMP paths. The updated CLI support exists for Cisco Nexus 3100 Series switches and not on Cisco Nexus 3000 Series switches. You can configure the hash offset in the range of <0-15> in non-concatenate mode and in the range of <0-63> in concatenate mode.

In concatenation mode, if the hash-offset is set to 0 and concatenation is set, the **show running-config** command displays **hardware ecmp hash-offset 0 concatenation**. The hash-offset is programmed as per value. On downgrade, if concatenation is configured, the CAP check asks to remove the configuration.

In non-concatenation mode, if the hash-offset is set to 0 and concatenation is reset, the **show running-config** command does not display **hardware ecmp hash-offset 0**. The hash-offset is programmed as per value if the hash-offset value is in range 0-15. The hash-offset displays CLI error if the value is in range 16-63 (The non-concatenated mode supports hash-offset for 0-15 range).

The hash-offset in **show running-config** is visible as per the configured value.

To configure an ECMP hash-offset, use the following commands in global configuration mode:

Command	Purpose
#**hardware ecmp hash-offset** *?*	Configures the ECMP hash-offset. The range is from 0 to 63. The hash offset in the range <0-15> is for the non-concatenate mode. The hash offset in the range <0-63> is for the concatenate mode.

Command	Purpose
#**hardware ecmp hash-offset** *number* **Example:** switch(config)# hardware ecmp hash-offset 5	Configures the ECMP hash-offset in the non-concatenate mode. The range is from 0 to 15. The default value is 0.
#**hardware ecmp hash-offset** *<0-63>* **concatenation** **Example:** switch(config)# hardware ecmp hash-offset 63 concatenation	Configures the ECMP hash-offset in the concatenate mode. **Note** Concatenation support exists for Cisco Nexus 3100 series switches and not for Cisco Nexus 3000 series switches. This CLI generates an error on Cisco Nexus 3000 series switches.

You can use the ECMP hash-offset configured by using the **hardware ecmp hash-offset** command along with different universal IDs configured by using the **ip load-sharing address** command to produce various hash results in the load-sharing algorithm.

Configuring Hash Polynomial

Starting with Release 6.0(2)U5(1), new CLI is added for the CRC configuration.

Command	Purpose
switch# **config t**	Enters configuration mode.
switch(config)#**hardware ecmp** ? hash-offset Configure hash offset hash-polynomial Configure hash polynomial	Displays hash-offset and hash-polynomial as the configuration options for hardware ECMP.
switch config)#**hardware ecmp hash-polynomial ?** CRC16 Hash polynomial CRC16 CRC32HI Hash polynomial CRC32 HI	Displays CRC16 and CRC32HI as configuration options for Hash polynomial.
switch config)#**show running-config**	Displays the running configuration.

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 how to display 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
```

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 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 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 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 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 shows how to display 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 how to display 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 checkerfor IPv4 or IPv6 routes, use the following commands in global configuration mode:

Command	Purpose
test [**ipv4**] [**ipv6**] [**unicast**] **forwarding inconsistency** [**vrf** *vrf-name*] [**module** {**1** | **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 to be checked can be either 1 or all.

To stop the Layer 3 consistency checker for IPv4 or IPv6 routes, use the following commands in global configuration mode:

Command	Purpose
test forwarding [**ipv4**] [**ipv6**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** {**1** | **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 can be either 1 or all.

To display the Layer 3 inconsistencies for IPv4 or IPv6 routes, use the following commands in any mode:

Command	Purpose
show forwarding [**ipv4**] [**ipv6**] **inconsistency** [**vrf** *vrf-name*] [**module** {**1** | **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 can be either 1 or all.

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:

Estimating Memory Requirements for Routes

Command	Purpose
clear forwarding {**ipv4**} **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 prefix. The *vrf-name* can be any case-sensitive, alphanumeric string up to 32 characters. The *slot* range is from 1 to 10.

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 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 route. • *prefix/length*—Any IP 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 unicast [**ip** | **ipv4**] {* | {*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 route. • *prefix/length*—Any IP 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 route	Displays routes in the FIB.
show hardware forwarding dynamic-allocation status	Displays information about the TCAM allocation.
show ip adjacency	Displays the adjacency table.
show ip route	Displays IPv4 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 15-11

Related Documents

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