Managing the Unicast RIB and FIB

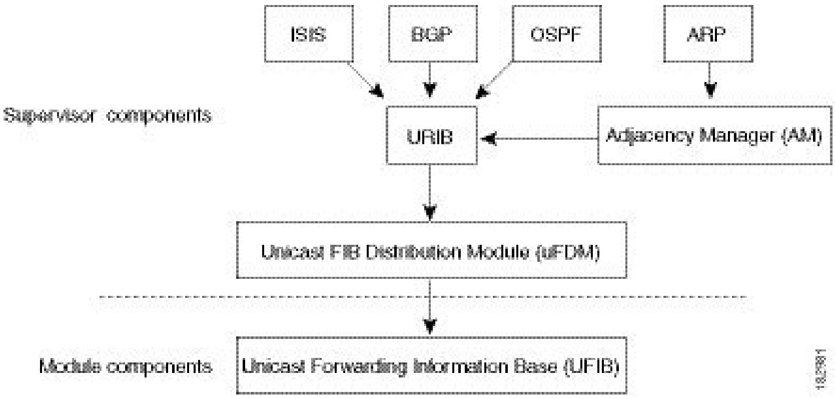
This chapter contains the following sections:

- About the Unicast RIB and FIB, on page 1
- Guidelines and Limitations for the Unicast RIB, on page 2
- Managing the Unicast RIB and FIB, on page 3
- Verifying the Unicast RIB and FIB Configuration, on page 10
- Additional References, on page 11

About the Unicast RIB and FIB

The unicast Routing Information Base (IPv4 RIB and IPv6 RIB) and Forwarding Information Base (FIB) are part of the Cisco NX-OS forwarding architecture, as shown in the following figure.

Figure 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).

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 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.

You can then manually clear any inconsistencies. See the Clearing Forwarding Information in the FIB section.

When more routes are learned exceeding the hardware limit, the **show consistency-checker forwarding ipv4** command is run, consistency may still show as pass. The same is true when it is transitioning from an inconsistent state to a consistent state. It may show as a failure. Until and unless the **test forwarding ipv4 inconsistency route** command is run again, it doesn't leave this state. This is an expected behavior.

Guidelines and Limitations for the Unicast RIB

The following guidelines and limitations apply to the URIB or U6RIB:

- In a virtual domain context (VDC), when modifying memory resource limits for the IPv4 or IPv6 unicast route, the modified limits do not take effect immediately.

  You must issue the **copy running-config startup-config** command followed by the **reload** command to activate the modified limits

  For example, if you issue either of the following commands, you will need to issue **copy running-config startup-config**, then reload the switch an extra time to activate the new setting:

  - **limit-resource u4route-mem**
  - **limit-resource u6route-mem**

Note If “feature pim” is configured for limit-resource, ensure that the value of **limit-resource u4route-mem** plus **limit-resource u6route-mem** is <= 1024 MB (1GB).

Managing the Unicast RIB and FIB

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

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

Command	Purpose
show forwarding {**ipv4** | **ipv6**} **adjacency module** *slot* **Example:** `switch# show forwarding ipv6 adjacency module 2`	Displays the adjacency information for IPv4 or IPv6.
show forwarding {**ipv4** | **ipv6**} **route module** *slot* **Example:** `switch# show forwarding ipv6 route module 2`	Displays the route table for IPv4 or IPv6.

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.

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:

Procedure

	Command or Action	Purpose
Step 1	**ip load-sharing address** {**destination port destination** | **source-destination** [**port source-destination** |]} [**universal-id** *seed*] [**rotate** *rotate*] [**concatenation**] **Example:** `ip load-sharing address source-destination`	Configures the unicast FIB load-sharing algorithm for data traffic. The following options are available for all IP load sharing configurations: • The **universal-id** option sets the random seed for the hash algorithm and shifts the flow from one link to another. You do not need to configure the universal ID. Cisco NX-OS chooses the universal ID if you do not configure it. The *universal-id* range is from 1 to 4294967295. • The **rotate** option causes the hash algorithm to rotate the link picking selection so that it does not continually choose the same link across all nodes in the network. It does so by influencing the bit pattern for the hash algorithm. This option shifts the flow from one link to another and load balances the already load-balanced (polarized) traffic from the first ECMP level across multiple links. If you specify a *rotate* value, the 64-bit stream is interpreted starting from that bit position in a cyclic rotation. The *rotate* range is from 1 to 63, and the default is 32. **Note** With multi-tier Layer 3 topology, polarization is possible. To avoid polarization, use a different rotate bit at each tier of the topology. **Note** To configure a rotation value for port channels, use the **port-channel load-balance src-dst ip-l4port rotate** *rotate* command. For more information on this command, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.

	Command or Action	Purpose
		• The **concatenation** option ties together the hash tag values for ECMP and the hash tag values for port channels in order to use a stronger 64-bit hash. If you do not use this option, you can control ECMP load-balancing and port-channel load-balancing independently. The default is disabled.
Step 2	(Optional) **show ip load-sharing** **Example:** `switch(config)# show ip load-sharing address source-destination`	Displays the unicast FIB load-sharing algorithm for data traffic.
Step 3	(Optional) **show routing hash** *source-addr dest-addr* [*source-port dest-port*] [**vrf** *vrf-name*] **Example:** `switch(config)# show routing hash 192.0.2.1 10.0.0.1`	Displays the route that the unicast RIB and unicast 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.

Example

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

```
switch# show routing hash 10.0.0.5 192.0.0.2
Load-share parameters used for software forwarding:
load-share mode: address source-destination port source-destination
Universal-id seed: 0xe05e2e85
Hash for VRF "default"
Hashing to path *172.0.0.2 (hash: 0x0e), for route:
```

Displaying Routing and Adjacency Information

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

Command	Purpose
show {**ip** | **ipv6**} **route** [*route-type* | **interface** *interface-type number* | **next-hop**] `switch# show ip route`	Displays the unicast route table. The *route-type* argument can be a single route prefix or a direct, static, or dynamic route protocol. Use the **?** command to see the supported interfaces.

Command	Purpose
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 **?** command to see the supported interfaces. • *vrf-id*—Any case-sensitive, alphanumeric string up to 64 characters.
show {**ip** | **ipv6**} **routing** [*route-type* | **interface** *interface-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 or a direct, static, or dynamic route protocol. Use the **?** command to see the supported interfaces.

This example shows how to display the unicast route table:

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

0.0.0.0/0, 1 ucast next-hops, 0 mcast next-hops
   *via 10.1.1.1, mgmt0, [1/0], 5d21h, static
0.0.0.0/32, 1 ucast next-hops, 0 mcast next-hops
   *via Null0, [220/0], 1w6d, local, discard
10.1.0.0/22, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.55, mgmt0, [0/0], 5d21h, direct
10.1.0.0/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.0.0, Null0, [0/0], 5d21h, local
10.1.1.1/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.1, mgmt0, [2/0], 5d16h, am
10.1.1.55/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.55, mgmt0, [0/0], 5d21h, local
10.1.1.253/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.253, mgmt0, [2/0], 5d20h, am
10.1.3.255/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.3.255, mgmt0, [0/0], 5d21h, local
255.255.255.255/32, 1 ucast next-hops, 0 mcast next-hops
   *via Eth Inband Port, [0/0], 1w6d, local
```

This example shows how to display the adjacency information:

```
switch# show ip adjacency
IP Adjacency Table for context 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:

Procedure

	Command or Action	Purpose
Step 1	**test forwarding** [**ipv4** | **ipv6**] [**unicast**] **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 64 characters. The *slot* range is from 1 to 26.
Step 2	**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 26.
Step 3	**show forwarding** [**ipv4** | **ipv6**] [**unicast**] **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 64 characters. The *slot* range is from 1 to 26.
Step 4	**show consistency-checker forwarding unicast** **Example:** `switch(config)# show consistency-checker forwarding unicast`	Displays the results of a Layer 3 consistency check for unicast routes.

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 device.

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

Command	Purpose

clear forwarding{**ipv4** | **ipv6**} **route** {***** | *prefix*} [**vrf** *vrf-name*] **module** {*slot* | **all**}
Example:
`switch# clear forwarding ipv4 route * module 1`
	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 64 characters. The *slot* range is from 1 to 26.

Configuring Maximum Routes for the Unicast RIB

You can configure the maximum number of routes allowed in the routing table.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**vrf context** *vrf-name*
	Example:
	`switch(config)# vrf context management2`
	`switch(config-vrf)#`
		Creates a VRF and enters VRF configuration mode.

Step 3	**address-family {ipv4 | ipv6} unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)`
		Enters the address-family configuration mode.

Step 4	**maximum routes** *max-routes* [*threshold* [**reinstall** *threshold*] | **warning -only**] **Example:** `switch(config-vrf-af-ipv4)# maximum routes 300000`	Configures the maximum number of routes allowed in the routing table. The range is from 1 to 4294967295. You can optionally specify the following: • *threshold*—Percentage of maximum routes that triggers a warning message. The range is from 1 to 100. • **warning-only**—Logs a warning message when the maximum number of routes is exceeded.

	Command or Action	Purpose
		• **reinstall** *threshold*—Reinstalls routes that previously exceeded the maximum route limit and were rejected and specifies the threshold value at which to reinstall them. The threshold range is from 1 to 100.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-vrf-af-ipv4)# copy running-config startup-config`	Saves this configuration change.

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 {**ipv6**} **memory estimate routes** *num-routes* **next-hops** *num-nexthops* **Example:** `switch# show routing memory estimate routes 5000 next-hops 2`	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 configuration mode:

Command	Purpose
clear {**ip** | **ip4** | **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 an case-sensitive, alphanumeric string up to 64 characters.
clear routing [**multicast** | **unicast**] [**ip** | **ip4** | **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 an case-sensitive, alphanumeric string up to 64 characters.

Verifying the Unicast RIB and FIB Configuration

To display the unicast RIB and FIB configuration information, perform one 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 {**ip** | **ipv4** | **ipv6**} **route**	Displays routes in the FIB.
show {**ip** | **ipv6**} **adjacency**	Displays the adjacency table.
show {**ip** | **ipv6**} **route**	Displays the IPv4 or IPv6 routes from the unicast RIB.

Command	Purpose
show routing	Displays routes from the unicast RIB.
`show system internal access-list dest-miss stats`	Displays statistics for packets dropped due to missing the FIB routes for the destinations, also called as DEST MISS. The output displays increment in the DEST MISS counters. **Note** Beginning with Cisco NX-OS Release 10.1(1), this feature is supported on Cisco Nexus 9300-FX3 platform switches.

Additional References

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

- Related Documents

Related Documents

Related Topic	Document Title
Configuring EEM	Cisco Nexus 9000 Series NX-OS System Management Configuration Guide