Managing the Unicast RIB and FIB

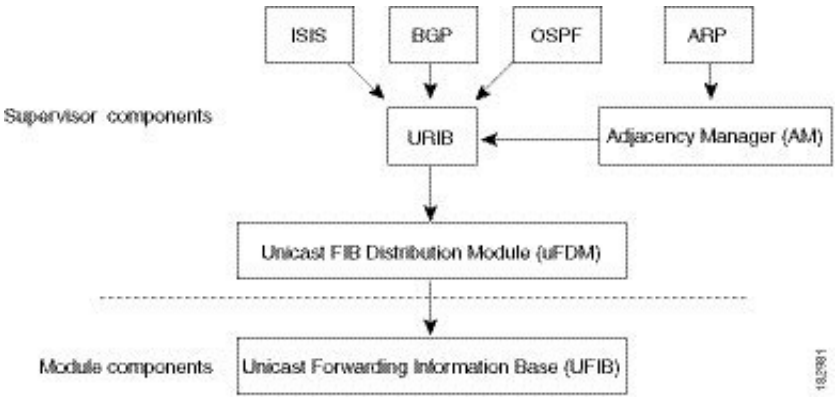
Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

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,

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.

Dynamic TCAM Allocation

Dynamic TCAM allocation reallocates unused TCAM blocks on M1 Series non-XL modules to an adjacent region when all existing blocks in that region are full. Dynamic TCAM allocation allows more flexibility in the number of routes that the FIB can allocate for a route type.

Cisco NX-OS divides the FIB to support multiple address families. The FIB TCAM for M1 Series non-XL modules has 128K physical entries.

Table 1: Default FIB TCAM Allocation

Region	Default Number of Routes	Number of TCAM Blocks	Entry Size
IPv4 Unicast Routes	56,000	7	72 bits
IPv4 Multicast Routes or IPv6 Unicast Routes	32,000	8	144 bits
IPv6 Multicast Routes	2,000	1	288 bits

Maximum TCAM Entries and FIB Scale Limits

The FIB TCAM entries are system wide resources that are shared across virtual device contexts (VDC) configured on the module. Table 16-2 describes the supported maximum FIB scale entries on the Nexus 7000 system configuration per route-type.

Table 2: Maximum Supported TCAM Entries and FIB Scale Limits

Module Type in a VDC	Maximum TCAM Physical Entries in a VDC	Maximum Supported IPv4 Unicast Routes	Maximum Supported IPv4 Multicast Routes	Maximum Supported IPv6 Unicast Routes	Maximum Supported IPv6 Multicast Routes
Only non-XL modules in a VDC	128,000	112,000	32,000 mroutes	56,000 routes	2,000 routes
Only XL modules in a VDC	900,000	900,000	32,000 mroutes	350,000 routes	2,000 routes
Mix of XL/non-XL modules in the same VDC	128,000	112,000	32,000 mroutes	56,000 routes	2,000 routes
Only F2 Series modules in a VDC[1]	32,000	32,768	16,384 mroutes	16,384 routes	8,192 routes

[1] Utilization may vary based on the sequence of routes being added and on the mix of unicast and multicast routes.

Note The table above captures the scale limits in a VDC. In a Cisco Nexus 7000 system, the total memory on the supervisor module restricts the actual route scale limits across all VDCs in the system.

Note Do not exceed the maximum route limits for non-XL modules in a VDC that contains both XL modules and non-XL modules.

Note The actual FIB TCAM can scale to a higher scale number from a hardware perspective. The table above captures the currently supported FIB sizes.

Note The maximum routes are individual route-type maximum values and these values are not cumulative across each route-type.

Configure the higher shared memory sizes (see the Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 5.x) for the routing table to enable the higher FIB scale on the XL modules. See the Cisco Nexus 7000 Series Hardware Installation and Reference Guide for more information on the XL modules.

Note The full IPv4 Internet route tables currently have more than 500,000 routes and require the XL modules.

The unicast RIB and FIB support virtual routing and forwarding (VRF) instances. VRF exists within VDCs. By default, Cisco NX-OS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

Default Settings for the Unicast RIB and FIB

Table 3: Default Unicast RIB and FIB Parameters

Parameters	Default
Dynamic TCAM allocation	Enabled by default and cannot be disabled.

Managing the Unicast RIB and FIB

Note If you are unfamiliar 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

The following show commands can be entered in any mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **show forwarding** {**ipv4** | **ipv6**} **adjacency module** *slot*	Displays the adjacency information for IPv4 or IPv6.
Step 2	switch# **show forwarding** {**ipv4** | **ipv6**} **route module***slot*	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.

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
switch(config)# **ip load-sharing address** {**destination port destination** | **source-destination** [**port source-destination**]} [**universal-id** *seed*] [**gtp-teid**] [**rotate** *rotate*] [**concatenation**]	Configures the unicast FIB load-sharing algorithm for data traffic. • 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 7000 Series NX-OS Interfaces Configuration Guide*. • For packets with GTP header, **gtp-teid** specifies that 32-bit TEID value has to be considered for the path calculation. • 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.

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

Command	Purpose
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
switch# **show routing hash** *source-addr dest-addr* [*source-port dest-port*] [**vrf** *vrf-name*]	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.

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

```
switch# show routing hash 10.0.0.5 30.0.0.2
   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 *20.0.0.2 (hash: 0x0e), for route:
```

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

Procedure

	Command or Action	Purpose
Step 1	switch(config-if)# **ip load-sharing per-packet**	Configures per-packet load sharing on an interface.

Displaying Routing and Adjacency Information

Procedure

	Command or Action	Purpose
Step 1	switch# **show** {**ip** | **ipv6**} **route** [*route-type* | **interface** *int-type number* | **next-hop**]	Displays the unicast route table. The route-type argument can be a single route prefix, direct, static, or a dynamic route protocol. Use the **?** command to see the supported interfaces.
Step 2	switch# **show** {**ip** | **ipv6**} **adjacency** [*prefix* | *interface-type number* [**summary**] | **non-best**] [**detail**] [**vrf** *vrf-id*]	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.
Step 3	switch# **show** {**ip** | **ipv6**} **routing** [*route-type* | **interface** *int-type number* | **next-hop** | **recursive-next-hop** | **summary** | **updated** {**since** | **until**} *time*]	Displays the unicast route table. The route-type argument can be a single route prefix, direct, static, or a dynamic route protocol. Use the **?** command to see the supported interfaces.

Example

The following example displays 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
```

The following example shows 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:

Command	Purpose
test forwarding [**ipv4** | **ipv6**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot*| **all**}]	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 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**	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**}]	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 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 device.

Procedure

	Command or Action	Purpose
Step 1	switch# **clear forwarding** {**ipv4** | **ipv6**} **route** {* | *prefix*} [**vrf** *vrf-name*] **module** {*slot* | **all**}	Clears one or more entries from the FIB. The route options are: • *—all routes. • *prefix*—Any IP or IPv6 prefix. The *vrf-name* can be a case-sensitive, alphanumeric string up to 64 characters. The *slot* range is from 1 to 10.
Step 2	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to clear one or more entries from the FIB:

```
switch(config)# clear forwarding ipv4 route * module 1
switch(config-if)# copy running-config startup-config
```

Configuring Maximum Routes for the Unicast RIB

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

Before you begin

Ensure that you are in the default VDC (or use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch (config)# **vrf context** *vrf name*	Creates a VRF and enters VRF configuration mode.
Step 3	switch (config-vrf)# **ip4 unicast**	Enters address family configuration mode.

	Command or Action	Purpose
Step 4	switch (config vrf-af-ipv4)# **maximum routes** *max routes* [*threshold* [**reinstall** *threshold*] | **warning -only**]	Configures the maximum number of routes allowed in the routing table. 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. • **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) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure maximum routes for the unicast RIB:

```
switch# configure terminal
switch(config)# vrf context Red
switch(config-vrf)# ipv4 unicast
switch(config-vrf-af-ipv4)# maximum routes 250 90
switch(config-vrf-af-ipv4)# copy running-config startup-config
```

Estimating Memory Requirements for Routes

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

Procedure

	Command or Action	Purpose
Step 1	switch# **show routing** {**ipv6**} **memory estimate routes** *num-routes* **next-hops** *num-nexthops*	Displays the memory requirements for routes. The *num-routes* range is from 1000 to 1000000. The *num-nexthops* range is from 1 to 16.
Step 2	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to estimate the memory requirements for routes:

```
switch# show routing memory estimate routes 5000 next-hops
switch(config-if)# copy running-config startup-config
```

Clearing Routes in the Unicast RIB

You can estimate the memory to be used by a number of routes and next-hop addresses.

Caution The ***** keyword is severely disruptive to routing.

Procedure

	Command or Action	Purpose
Step 1	**clear** {**ip** | **ip4** | **ipv6**} **route** {* | *prefix/length*} [**next hop interface**] [**vrf** *vrf-name*] **module** {*slot* | **all**}	Clears one or more routes from both the unicast RIB and all the module FIBs. The route options are: • *****—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 32 64characters.
Step 2	**clear routing** [**multicast** | **unicast**]{**ip** |**ip4** | **ipv6**} {* | {*route* | *prefix/length*} [*next-hop interface*]} [**vrf** *vrf-name*]	Clears one or more routes from both the unicast RIB and all the module FIBs. The route options are: • *****—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 64 characters.
Step 3	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Verifying the Unicast RIB and FIB

To display advanced BGP statistics, use the following commands:

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** |**ipv4** | **ipv6**} **adjacency**}	Displays the adjacency table.
show {**ip** | **ipv6**} **route**}	Displays the IPv4 or IPv6 routes from the unicast RIB.
show routing	Displays routes from the unicast RIB.

Related Documents for the Unicast RIB and FIB

Feature Name	Feature Information
Unicast RIB and FIB CLI commands	*Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*

Feature History for the Unicast RIB and FIB

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 4: Feature History for the Unicast RIB and FIB

Feature Name	Release	Feature Information
Load Sharing in Unicast FIB	7.3(0)DX(1)	Added support for GTP headers.
Unicast FIB	6.2(2)	Added the ability to check for inconsistent, missing, or failed routes in the unicast FIB.
TCAM utilization	6.2(2)	Added the ability to monitor TCAM utilization on M1 Series modules.
Unicast RIB	6.2(2)	Added the optional keyword **longer-prefixes** [**detail**] to the **show routing** command to display specific routes for a particular prefix.
Maximum routes	5.2(1)	Added support to configure the maximum number of routes allowed in the routing table.
TCAM Size for XL Modules	5.0(2)	Added support for larger TCAM and FIB sizes with XL modules.

Feature Name	Release	Feature Information
Dynamic TCAM allocation	5.0(2)	Enabled by default and cannot be disabled.
IPv6 forwarding inconsistency checker	4.2(1)	Added support to check for inconsistencies in the IPv6 forwarding table.
Dynamic TCAM allocation	4.2(1)	Added support for dynamically allocating TCAM blocks in the FIB.
Per-packet load sharing	4.1(2)	Added support to load balance per packet on an interface.
Unicast RIB and FIB	4.0(3)	Added support to clear individual routes in unicast RIB and FIB.
Unicast RIB and FIB	4.0(1)	This feature was introduced.