

Troubleshooting Routing

- About Troubleshooting Routing Issues, on page 1
- Initial Troubleshooting Routing Checklist, on page 1
- Troubleshooting Routing, on page 2
- Troubleshooting Policy-Based Routing, on page 5
- Troubleshoot Dynamic Load Balancing, on page 5

About Troubleshooting Routing Issues

Layer 3 routing involves determining optimal routing paths and packet switching. You can use routing algorithms to calculate the optimal path from the router to a destination. This calculation depends on the algorithm selected, route metrics, and other considerations such as load balancing and alternate path discovery.

Cisco NX-OS supports multiple virtual routing and forwarding (VRF) instances and multiple routing information bases (RIBs) to support multiple address domains. Each VRF is associated with a RIB, and this information is collected by the Forwarding Information Base (FIB).

See the following documents for more information on routing:

- *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*
- *Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide*

Initial Troubleshooting Routing Checklist

You can troubleshoot routing issues by checking these items first:

Checklist	Done
Verify that the routing protocol is enabled.	
Verify that the address family is configured if necessary.	
Verify that you have configured the correct VRF for your routing protocol.	

Use the following commands to display routing information:

- **show ip arp**

- **show ip traffic**
- **show ip static-route**
- **show ip client**
- **show ip fib**
- **show ip process**
- **show ip route**
- **show vrf**
- **show vrf interface**

Troubleshooting Routing

SUMMARY STEPS

1. switch# **show ospf**
2. switch# **show running-config eigrp all**
3. switch# **show running-config eigrp**
4. switch# **show processes memory** | **include isis**
5. switch# **show ip client pim**
6. switch# **show ip interface** *loopback-interface*
7. switch# **show vrf interface** *loopback -interface*
8. switch# **show routing unicast clients**
9. switch# **show forwarding distribution multicast client**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	switch# **show ospf**
	Example:
	`switch# show ospf`
	`                    ^`
	`% invalid command detected at '^' marker.`
		Verifies that the routing protocol is enabled.
		If the feature is not enabled, Cisco NX-OS reports that the command is invalid.

Step 2	switch# **show running-config eigrp all** **Example:** `switch# show running-config eigrp all`	Verifies the configuration for this routing protocol.
Step 3	switch# **show running-config eigrp** **Example:**	Verifies the VRF configuration for this routing protocol.

	Command or Action	Purpose

	`switch# show running-config eigrp`
	`version 6.1(2)I1(1)`
	`feature eigrp`
	`router eigrp 99`
	`  address-family ipv4 unicast`
	`    router-id 192.0.2.1`
	`  vrf red`
	`    stub`	

Step 4	switch# **show processes memory | include isis**
	Example:
	`switch# show processes memory | include isis`
	` 8913   9293824  bffff1d0/bffff0d0  isis`
	`32243   8609792  bfffe0c0/bfffdfc0  isis`
		Checks the memory utilization for this routing protocol.

Step 5	switch# **show ip client pim**
	Example:
	`switch# show ip client pim`
	`  Client: pim, uuid: 284, pid: 3839, extended pid:`
	` 3839`
	`  Protocol: 103, client-index: 10, routing VRF id:`
	` 255`
	`  Data MTS-SAP: 1519`
	`  Data messages, send successful: 2135, failed: 0`
		Verifies that the routing protocol is receiving packets.

Step 6	switch# **show ip interface** *loopback-interface*
	Example:
	`switch# show ip interface loopback0`
	`loopback0, Interface status:`
	`protocol-up/link-up/admin-up, iod: 36,`
	`Context:"default"`
	`  IP address: 1.0.0.1, IP subnet: 1.0.0.0/24`
	`  ...`
	`  IP multicast groups locally joined:`
	`      224.0.0.2  224.0.0.1  224.0.0.13`
	`  ...`
		Verifies that the routing protocol is enabled on an interface.

Step 7	switch# **show vrf interface** *loopback -interface*
	Example:
	`switch# show vrf interface loopback 99`
	`Interface                    VRF-Name`
	`         VRF-ID`
	`loopback99                    default`
	`             1`
		Verifies that the interface is in the correct VRF.

Step 8	switch# **show routing unicast clients** **Example:** `switch# show routing unicast clients`	Verifies that the routing protocol is registered with the RIB.

Step 9	switch# **show forwarding distribution multicast client**
	Example:
	`switch# show forwarding distribution multicast`
	`client`
		Verifies that the RIB is interacting with the forwarding plane.

	Command or Action	Purpose
	Number of Clients Registered: 3 Client-name Client-id Shared Memory Name igmp 1 N/A mrib 2 /procket/shm/mrib-mfdm	

Example

This example shows how to display the EIGRP routing protocol configuration:

```
switch# show running-config eigrp all
version 6.1(2)I1(1)
feature eigrp
router eigrp 99
log-neighbor-warnings
  log-neighbor-changes
  log-adjacency-changes
  graceful-restart
  nsf
  timers nsf signal 20
  distance 90 170
  metric weights 0 1 0 1 0 0
  metric maximum-hops 100
  default-metric 100000 100 255 1 1500
  maximum-paths 16
  address-family ipv4 unicast
    log-neighbor-warnings
    log-neighbor-changes
    log-adjacency-changes
    graceful-restart
    router-id 192.0.2.1
    nsf
    timers nsf signal 20
    distance 90 170
    metric weights 0 1 0 1 0 0
    metric maximum-hops 100
    default-metric 100000 100 255 1 1500
    maximum-paths 16
```

This example shows how to display that the unicast routing protocol is registered with the RIB:

```
switch# show routing unicast clients
CLIENT: am
 index mask: 0x00000002
 epid: 3908      MTS SAP: 252       MRU cache hits/misses:        2/1
 Routing Instances:
  VRF: management        table: base
 Messages received:
  Register          : 1      Add-route         : 2      Delete-route      : 1
 Messages sent:
  Add-route-ack     : 2      Delete-route-ack  : 1
CLIENT: rpm
 index mask: 0x00000004
 epid: 4132      MTS SAP: 348       MRU cache hits/misses:        0/0
 Messages received:
  Register          : 1
 Messages sent:
```

```
...
CLIENT: eigrp-99
 index mask: 0x00002000
 epid: 3148      MTS SAP: 63775     MRU cache hits/misses:        0/1
 Routing Instances:
  VRF: default           table: base                notifiers: self
 Messages received:
  Register          : 1      Delete-all-routes : 1
 Messages sent:
...
```

Troubleshooting Policy-Based Routing

- Make sure the ACLs match the incoming traffic.
- Make sure the route is available:
  - For IP network routes, use the **show ip route** command to make sure the IP network route is available for the next hop specified in the **set ip next-hop** command.
  - For IP host routes, use the **show ip arp** command to make sure the IP host route is available for the next hop specified in the **set ip next-hop** command.
  - For IPv6 network routes, use the **show ipv6 route** command to make sure the IPv6 network route is available for the next hop specified in the **set ipv6 next-hop** command.
  - For IPv6 host routes, use the **show ipv6 neighbor** command to make sure the IPv6 host route is available for the next hop specified in the **set ipv6 next-hop** command.
- Make sure the policy is active in the system (using the **show ip policy** command).
- Check the statistics for the entry (using the **show route-map** *map-name* **pbr-statistics** command).

Troubleshoot Dynamic Load Balancing

Consistency checker can be used to troubleshoot the routes using DLB ECMP as follows:

- Global Consistency Checker
  - **test consistency-checker forwarding ipv4 unicast**
  - **show consistency-checker forwarding ipv4 unicast**

Sample output

```
Leaf1# test consistency-checker forwarding ipv4 unicast
Consistency check started.
Leaf1#
Leaf1#
Leaf1# show consistency-checker forwarding ipv4 unicast
IPV4 Consistency check : table_id(0x1)
Execution time : 28 ms ()
No inconsistent adjacencies.
No inconsistent routes.
Consistency-Checker: PASS for ALL
```

- Single Route Consistency Checker
  - **show consistency-checker forwarding single-route ipv4** *ipv4 address* **vrf** *vrf*

Sample output

```
Leaf1# show consistency-checker forwarding single-route ipv4 64.60.60.0/24 vrf default


Consistency checker passed for 64.60.60.0/24
Leaf1#
```