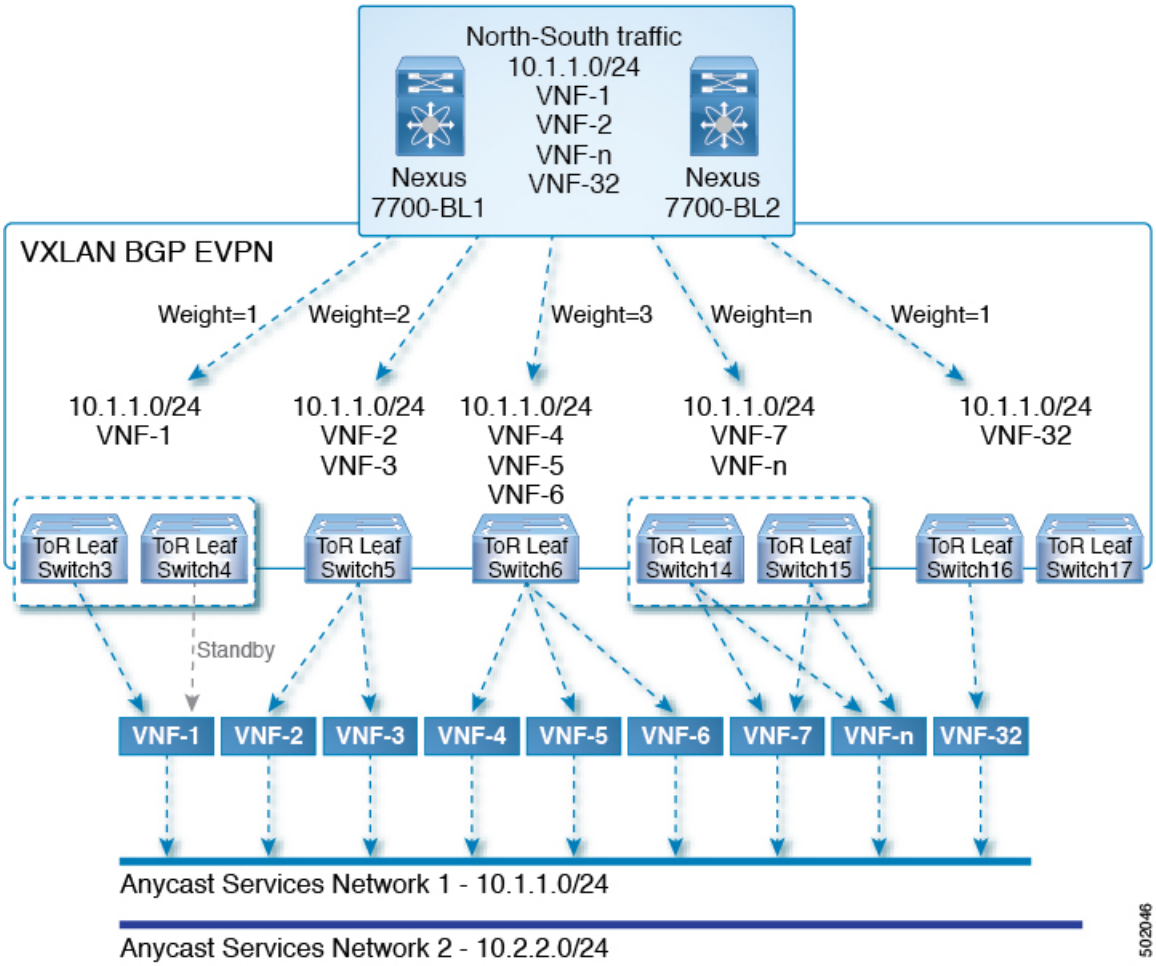
Proportional Multipath for VNF

This chapter contains the following sections:

Information About Proportional Multipath for VNF

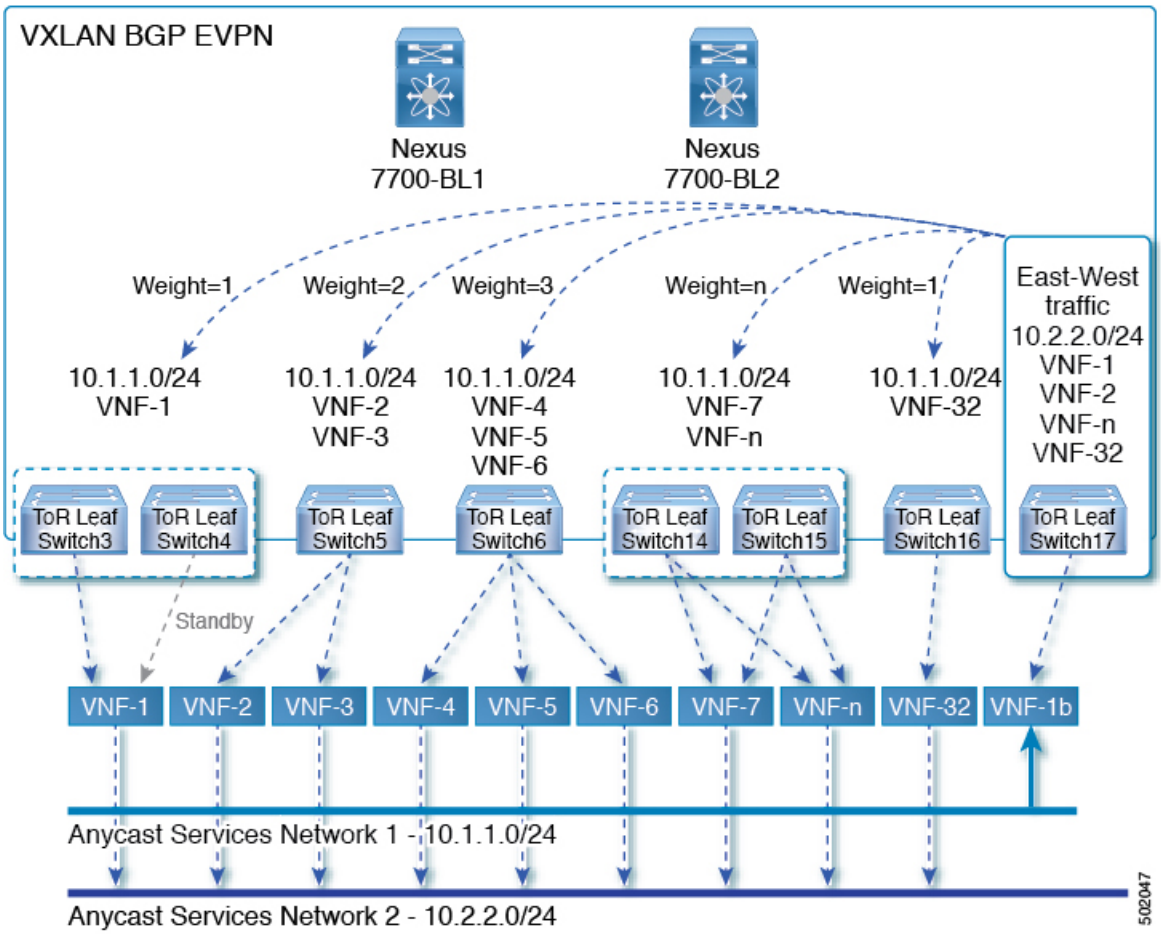
In Network Function Virtualization Infrastructures (NFVi), anycast services networks are advertised from multiple Virtual Network Functions (VNFs). The Proportional Multipath for VNF feature enables advertising of all the available next hops to a given destination network. This feature enables the switch to consider all paths to a given route as equal cost multipath (ECMP) allowing the traffic to be forwarded using all the available links stretched across multiple ToRs.

Figure 1: Sample Topology (North-South Traffic)



As shown in Figure 1, North-South traffic that enters the VXLAN fabric at a border leaf is sent across all egress endpoints with the traffic forwarded proportional to the number of links from the egress Top-Of-Rack (TOR) to the destination network.

Figure 2: Sample Topology (East-West Traffic)



As shown in Figure 2, East-West traffic is forwarded between the VXLAN tunnel End Points (VTEPs) proportional to the number of next hops that are advertised by each Top-Of-Rack (TOR) switch to the destination network.

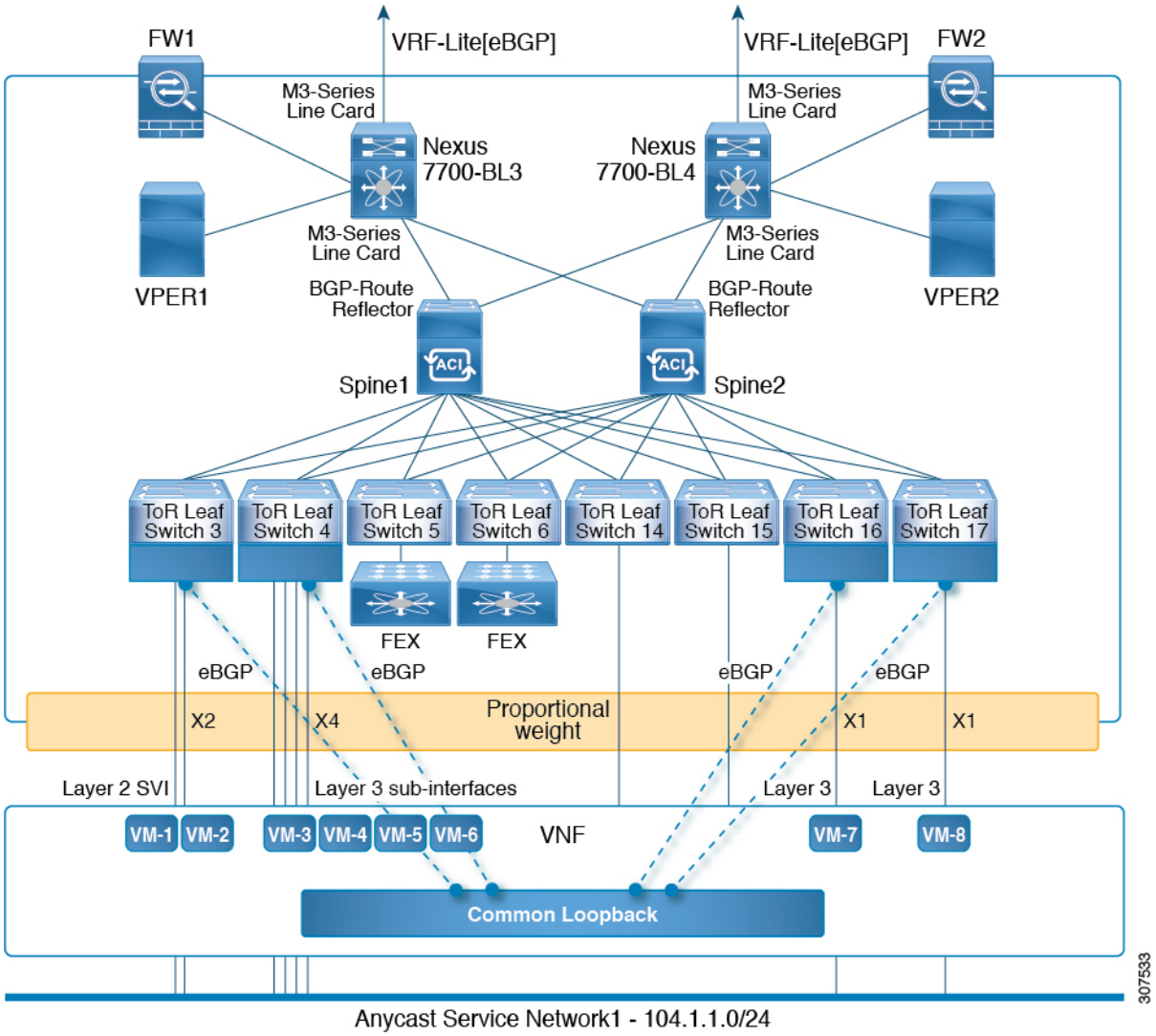
The switch uses BGP to advertise reachability within the fabric using the Layer 2 VPN (L2VPN)/Ethernet VPN (EVPN) address family. If all TOR switches and border leaves are within the same autonomous system (AS), a full internal BGP (iBGP) mesh is configured by using route reflectors or by having each BGP router peer with every other router.

Each TOR and border leaf constitutes a VTEP in the VXLAN fabric. You can use a BGP route reflector to reduce the full mesh BGP sessions across the VTEPs to a single BGP session between a VTEP and the route reflector. Virtual Network Identifiers (VNIs) are globally unique within the overlay. Each Virtual Routing and Forwarding (VRF) instance is mapped to a unique VNI. The inner destination MAC address in the VXLAN header belongs to the receiving VTEP that does the routing of the VXLAN payload. This MAC address is distributed as a BGP attribute along with the EVPN routes.

Multipath flattening is also supported. This leads to flattened IPv4 and IPv6 routes being sent to the FIB. The FIB then has a flattened set of next-hops that are sorted based on the increasing order of IP address.

The alternate deployment topology given below shows how a common loopback address is configured for the BGP connection between the ToR switches and the VNF instance.

Figure 3: Sample Topology for Common Loopback Address



Advertisement of Customer Networks

Customer networks are configured statically or learned locally by using Interior Gateway Protocol (IGP) or external BGP (eBGP) over a Provider Edge(PE)-Customer Edge(CE) link. In the data center fabric, the PE-CE link corresponds to access links from leaf devices to attached hosts (physical devices/VMs) or routers. These networks are redistributed into BGP and advertised to the VXLAN fabric.

The networks that are advertised to the TORs by the Virtual Machines (VMs) attached to them are advertised to the VXLAN fabric as EVPN Type-5 routes with the following:

- The Route Distinguisher (RD) will be the L3VNI's configured RD.
- The gateway IP field will be populated with the next-hop.
- The next-hop of the EVPN route continues to be the VTEP-IP.

- The export route targets of the routes are derived from the configured export route targets of the associated Layer 3 VNI.

Multiple VRF routes may generate the same Type-5 Network Layer Reachability Information (NLRI) differentiated only by the gateway IP field as the routes are advertised with the L3VNI's RD and the gateway IP is not part of the Type-5 NLRI's key. The Network Layer Reachability Information (NLRI) is exchanged between BGP routers using UPDATE messages. These routes are advertised to the EVPN AF by extending the BGP export mechanism to include ECMPs and by using the add-path BGP feature in the EVPN AF.

Each Type-5 route within the EVPN AF that is created by using this feature may have multiple paths that are imported into the corresponding VRF based on the matching of the received route targets and by having ECMP enabled within the VRF and in the EVPN AF. Within the VRF, the route will be a single prefix with multiple paths. Each path represents a Type-5 EVPN path or those paths that are learned locally within the VRF. Those EVPN Type-5 routes that are enabled for this feature will have their next hop (NH) in the VRF derived from their gateway IP field. Use the **export-gateway-ip** command to enable BGP to advertise the gateway IP in the EVPN Type-5 routes.

Use the **maximum-paths mixed** command to enable BGP and the Unicast Routing Information Base (URIB) to consider the following paths as ECMP:

- iBGP paths
- eBGP paths
- Paths from other protocols (such as static) that are redistributed or injected into BGP.

The paths can be either local to the device (redistributed static or network-originated) or remote (eBGP or iBGP learned over BGP-EVPN or PE-CE). This overrides the default route selection behavior in which local routes are preferred over remote routes. URIB downloads all NHs of the route, including locally learned and user-configured routes, to the Unicast FIB Distribution Module (uFDM)/Forwarding Information Base (FIB).

Legacy Peer Support

Use the **advertise-gw-ip** command to advertise EVPN Type-5 routes with the gateway IP set. TORs will then advertise the gateway IP in the Type-5 NLRI. However, legacy peers running on NX-OS versions older than Cisco NX-OS Release 8.3(1) cannot process the Gateway IP which may lead to unexpected behavior. To prevent this scenario from occurring, use the **no advertise-gw-ip** command. BGP will then set the gateway IP field of the Type-5 NLRI to zero even if the path being advertised has a valid gateway IP.

The **no advertise-gw-ip** command flaps the specified peer session as gracefully as possible. The remote peer triggers graceful restart if the peer supports this capability. When the session is reestablished, the local peer advertises EVPN Type-5 routes with the gateway IP set or with the gateway IP as zero depending on whether the **advertise-gw-ip** command has been used or not. By default, this knob is enabled and the gateway IP field is populated with the appropriate next hop value.

Guidelines and Limitations for Proportional Multipath for VNF

- Static and direct routes have to be redistributed into the BGP when the Proportional Multipath for VNF feature is enabled.
- Routes cannot be redistributed into BGP if OSPF or EIGRP is being used as an IGP.

- If the Proportional Multipath for VNF feature is enabled and the routes are not redistributed into BGP, asymmetric load balancing of traffic may occur as the local routes from URIB may not show up in BGP and on remote TORs as EVPN paths.
- Devices on which mixed-multipath is enabled must support the same load-balancing algorithm.Otherwise, traffic tromboning may occur.
- If a VNF instance is multi-homed to multiple TORs, policies have to be configured or BGP routes have to be originated using a network command. This results in each TOR's connection to the VNF being displayed in the BGP routing table. Each TOR can now see the VNF's direct routes to the other TORs in which the VNF is multi-homed. This allows each TOR to advertise paths to the Gateway IPs through other TORs leading to a next hop resolution loop.

  Consider a scenario in which a VNF is multi-homed to two TORs, TOR1 and TOR2. Individual links to the TORs are addressed as 1.1.1.1 and 2.2.2.2. If the VNF advertises a service 192.168.1.0/24 through the TORs, the TORs advertise EVPN routes to 192.168.1.0/24 with Gateway IPs of 1.1.1.1 and 2.2.2.2 respectively.

  This causes an issue with the Recursive Next Hop (RNH) resolution on a remote TOR (for example, TOR3). The gateway IP is resolved to a /24 route pointing to another gateway IP. That second gateway IP is resolved by a route pointing to the first gateway IP. So, in our scenario, the gateway IP 1.1.1.1 is resolved by 1.1.1.0/24 which points to 2.2.2.2. And 2.2.2.2 is resolved by 2.2.2.0/24 which points to 1.1.1.1.

  The above condition occurs as both TORs connected to the VNF are advertising the VNF’s connected routes. TOR1 is advertising 1.1.1.0/24 and 2.2.2.0/24. However, 1.1.1.0 is advertised without a gateway IP as it is a connected subnet on TOR1. Also, 2.2.2.0 is an OSPF route pointing to 1.1.1.1 which is the VNF’s address connected to TOR1.

  Similarly, TOR2 advertises both subnets and 2.2.2.0/24 is sent without a gateway IP as it is directly connected to TOR2. 1.1.1.0 is learned via OSPF and is sent with a gateway IP of 2.2.2.2 which is the VNF’s address connected to TOR2. 1.1.1.1/32 and 2.2.2.2/32 will not be advertised as they are Adjacency Manager (AM) routes on each TOR.

  This issue does not have a resolution when Type-5 routes are involved. However, this scenario can be avoided if the TORs advertise the gateway IP’s /32 address using a network command. And if the gateway IPs are being resolved by Type-2 EVPN MAC/IP routes, this scenario can be avoided as the gateway IP will be resolved by the /32 IP route.

Default Setting for Proportional Multipath for VNF

Parameter	Default
Proportional Multipath for VNF	Disabled

Configuring Proportional Multipath for VNF

Configuring the Route Reflector

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Assign an autonomous system (AS) number to a device and enter the BGP configuration mode:

switch(config)# **router bgp** *<500000>*

Step 3 Required: Configure Layer-2 VPN EVPN parameters in the VXLAN EVPN fabric:

switch(config-router)# **address-family l2vpn evpn**

Step 4 Configure the capability of sending and receiving additional paths to and from all the neighbors in this address family:

switch(config-router-af)# **additional paths send**

switch(config-router-af)# **additional paths receive**

switch(config-router-af)# **additional paths selection route-map** *<passall>*

Step 5 Exit the address family configuration mode:

switch(config-router-af)# **exit**

Step 6 Exit the BGP configuration mode:

switch(config-router)# **exit**

Step 7 Enter the route-map configuration mode and define the conditions for redistributing routes from one routing protocol to another:

switch(config)# **route-map** *<passall>* **permit** *<10>*

Step 8 Set the path selection criteria:

switch(config-route-map)# **set path-selection all advertise**

Running Configuration

This example shows a running configuration for a route reflector device.

```
configure terminal
  router bgp 500000
    address-family l2vpn evpn
      additional paths send
```

```
    additional paths receive
    additional paths selection route-map passall
route-map passall permit 10
  set path-selection all advertise
```

Configuring the ToR

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure BGP:

switch(config)# **router bgp** *<500000>*

Step 3 Required: Configure address family Layer 2 VPN EVPN under router bgp context:

switch(config-router)# **address-family l2vpn evpn**

Step 4 Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):

- iBGP paths
- eBGP paths
- Paths from other protocols (such as static) that are redistributed or injected into BGP

switch(config-router-af)# **maximum-paths mixed** *<32>*

Step 5 The additional-paths configuration for sending:

switch(config-router-af)# **additional paths send**

Step 6 The additional-paths configuration for receiving:

switch(config-router-af)# **additional paths receive**

Step 7 The additional-paths configuration applied for the route-map:

switch(config-router-af)# **additional paths selection route-map** *<passall>*

Step 8 Exits the command mode:

switch(config-router-af)# **exit**

Step 9 Enter the VRF configuration mode:

switch(config-router)# **vrf evpn-tenant-1001**

Step 10 Configure address family for IPv4.

switch(config-router-vrf)# **address-family ipv4 unicast**

Step 11 Enable BGP to advertise the gateway IP in EVPN Type-5 routes:

switch(config-router-vrf-af)# **export-gateway-ip**

Step 12 Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):

- iBGP paths
- eBGP paths
- Paths from other protocols (such as static) that are redistributed or injected into BGP

switch(config-router-vrf-af)# **maximum-paths mixed** *<32>*

Step 13 Exits the command mode:

switch(config-router-vrf-af)# **exit**

Step 14 Enter the address family configuration mode:

switch(config-router-vrf)# **address-family ipv6 unicast**

Step 15 Enable BGP to advertise the gateway IP in EVPN Type-5 routes:

switch(config-router-vrf-af)# **export-gateway-ip**

Step 16 Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):

- iBGP paths
- eBGP paths
- Paths from other protocols (such as static) that are redistributed or injected into BGP

switch(config-router-vrf-af)# **maximum-paths mixed** *<32>*

Step 17 Exits the comand mode:

switch(config-router-vrf-af)# **exit**

Step 18 Configure the route-map:

switch(config)# **route-map passall permit** *<10>*

Step 19 Sets the route-map related to the additional-paths feature:

switch(config-route-map)# **set path-selection all advertise**

Running Configuration

This example shows a running configuration for a ToR device.

```
configure terminal
  router bgp 500000
    address-family l2vpn evpn
      maximum-paths mixed 32
      additional paths send
```

```
    additional paths receive
    additional paths selection route-map passall
  vrf cust_1
    address-family l2vpn evpn
      export-gateway-ip
      maximum-paths mixed 32
route-map passall permit 10
  set path-selection all advertise
```

Configuring the Border Leaf

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure BGP:

switch(config)# **router bgp** *<500000>*

Step 3 Required: Configure address family Layer 2 VPN EVPN under router bgp context:

switch(config-router)# **address-family l2vpn evpn**

Step 4 Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):

- iBGP paths
- eBGP paths
- Paths from other protocols (such as static) that are redistributed or injected into BGP.

switch(config-router-af)# **maximum-paths mixed** *<32>*

Step 5 The additional-paths configuration for sending:

switch(config-router-af)# **additional paths send**

Step 6 The additional-paths configuration for receiving:

switch(config-router-af)# **additional paths receive**

Step 7 The additional-paths configuration enables the additional-paths feature:

switch(config-router-af)# **additional paths selection route-map** *<passall>*

Step 8 Exits the command mode:

switch(config-router-af)# **exit**

Step 9 Enter the VRF configuration mode:

switch(config-router)# **vrf evpn-tenant-1001**

Step 10 Configure address family for IPv4:

switch(config-router-vrf)# **address-family ipv4 unicast**

Step 11 Enable BGP to advertise the gateway IP in EVPN Type-5 routes:

switch(config-router-vrf-af)# **export-gateway-ip**

Step 12 Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):

- iBGP paths
- eBGP paths
- Paths from other protocols (such as static) that are redistributed or injected into BGP.

switch(config-router-vrf-af)# **maximum-paths mixed** *<32>*

Step 13 Exit the address family configuration mode:

switch(config-router-vrf-af)# **exit**

Step 14 Enter the address family configuration mode:

switch(config-router-vrf)# **address-family ipv6 unicast**

Step 15 Enable BGP to advertise the gateway IP in EVPN Type-5 routes:

switch(config-router-vrf-af)# **export-gateway-ip**

Step 16 Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):

- iBGP paths
- eBGP paths
- Paths from other protocols (such as static) that are redistributed or injected into BGP.

switch(config-router-vrf-af)# **maximum-paths mixed** *<32>*

Step 17 Exits the command mode:

switch(config-router-vrf-af)# **exit**

Step 18 Configure the route-map:

switch(config)# **route-map passall permit** *<10>*

Step 19 Sets the route-map related to the additional-paths feature:

switch(config-route-map)# **set path-selection all advertise**

Step 20 Exit the route-map configuration mode.

switch(config-route-map)# **exit**

Step 21 Configure the unicast FIB load sharing algorithm for data traffic.

switch(config)# **ip load-sharing address source-destination rotate** *<32>* **universal-id** *<1>*

Note

- 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 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.
- With multi-tier Layer 3 topology, polarization is possible. To avoid polarization, use a different rotate bit at each tier of the topology.
- 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*.

Running Configuration

This example shows a running configuration for a Border Leaf.

```
configure terminal
  router bgp 500000
    address-family l2vpn evpn
      maximum-paths mixed 32
      additional paths send
      additional paths receive
      additional paths selection route-map passall
    vrf cust_1
      address-family l2vpn evpn
        export-gateway-ip
        maximum-paths mixed 32
  route-map passall permit 10
    set path-selection all advertise
  ip load-sharing address source-destination rotate 32 universal-id 1
```

Configuring a BGP Legacy Peer

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Assign an autonomous system (AS) number to a device and enter the BGP configuration mode:

switch(config)# **router bgp** *<500000>*

Step 3 Configure a BGP neighbor and enter the neighbor configuration mode:

switch(config-router)# **neighbor** *<102.102.102.102>*

Step 4 Specify autonomous system number of the neighbor:

switch(config-router-neighbor)# **remote-as** *<2000000>*

Step 5 Enter the address family configuration mode:

switch(config-router-neighbor)# **address-family l2vpn evpn**

Step 6 Disable the Proportional Multipath for VNF feature for legacy peers:

switch(config-router-neighbor-af)# **no advertise-gw-ip**

Note EVPN Type-5 routes will now be sent with 0 as the gateway IP.

Running Configuration

This example shows a running configuration for a route reflector device.

```
router bgp 500000
  neighbor 102.102.102.102
    remote-as 2000000
    address-family l2vpn evpn
      no advertise-gw-ip
```

Verifying Proportional Multipath for VNF

Command	Purpose
show bgp ipv4 unicast	Displays Border Gateway Protocol (BGP) information for the IPv4 unicast address family.
show bgp l2vpn evpn	Displays BGP information for the Layer-2 Virtual Private Network (L2VPN) Ethernet Virtual Private Network (EVPN) address family.
show ip route	Displays routes from the unicast RIB.

The following example shows how to display BGP information for the L2VPN EVPN address family:

```
switch# show bgp l2vpn evpn 11.1.1.0
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 13.13.13.13:3       //     Remote route
BGP routing table entry for [5]:[0]:[0]:[24]:[11.1.1.0]/224, version 1341
Paths: (3 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: eBGP
```

```
  Advertised path-id 1
  Path type: external, path is valid, is best path
             Imported to 2 destination(s)
  Gateway IP: 11.1.1.133
  AS-Path: 2000000 100000 , path sourced external to AS
    11.11.11.11 (metric 5) from 102.102.102.102 (102.102.102.102)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 22001
      Received path-id 3
      Extcommunity: RT:23456:22001 Route-Import:11.11.11.11:2001 ENCAP:8
          Router MAC:003a.7d7d.1dbd

  Path type: external, path is valid, not best reason: Neighbor Address, multipath
             Imported to 2 destination(s)
  Gateway IP: 11.1.1.233
  AS-Path: 2000000 100 , path sourced external to AS
    33.33.33.33 (metric 5) from 102.102.102.102 (102.102.102.102)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 22001
      Received path-id 2
      Extcommunity: RT:23456:22001 Route-Import:33.33.33.33:2001 ENCAP:8
          Router MAC:e00e.da4a.589d

  Path type: external, path is valid, not best reason: Neighbor Address, multipath
             Imported to 2 destination(s)
  Gateway IP: 11.1.1.100
  AS-Path: 2000000 500000 , path sourced external to AS
    22.22.22.22 (metric 5) from 102.102.102.102 (102.102.102.102)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 22001
      Received path-id 1
      Extcommunity: RT:23456:22001 Route-Import:22.22.22.22:2001 ENCAP:8
          Router MAC:e00e.da4a.62a5

  Path-id 1 not advertised to any peer

Route Distinguisher: 4.4.4.4:3    (L3VNI 22001)       //   Local L3VNI
BGP routing table entry for [5]:[0]:[0]:[24]:[11.1.1.0]/224, version 3465
Paths: (3 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: eBGP

  Advertised path-id 1
  Path type: external, path is valid, is best path
             Imported from 13.13.13.13:3:[5]:[0]:[0]:[24]:[11.1.1.0]/224
  Gateway IP: 11.1.1.100
  AS-Path: 2000000 500000 , path sourced external to AS
    22.22.22.22 (metric 5) from 102.102.102.102 (102.102.102.102)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 22001
      Received path-id 1
      Extcommunity: RT:23456:22001 Route-Import:22.22.22.22:2001 ENCAP:8
          Router MAC:e00e.da4a.62a5

  Path type: external, path is valid, not best reason: newer EBGP path, multipat
h
             Imported from 13.13.13.13:3:[5]:[0]:[0]:[24]:[11.1.1.0]/224
  Gateway IP: 11.1.1.233
  AS-Path: 2000000 100 , path sourced external to AS
    33.33.33.33 (metric 5) from 102.102.102.102 (102.102.102.102)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 22001
      Received path-id 2
```

```
      Extcommunity: RT:23456:22001 Route-Import:33.33.33.33:2001 ENCAP:8
          Router MAC:e00e.da4a.589d

  Path type: external, path is valid, not best reason: newer EBGP path, multipat
h
              Imported from 13.13.13.13:3:[5]:[0]:[0]:[24]:[11.1.1.0]/224
  Gateway IP: 11.1.1.133
  AS-Path: 2000000 100000 , path sourced external to AS
    11.11.11.11 (metric 5) from 102.102.102.102 (102.102.102.102)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 22001
      Received path-id 3
      Extcommunity: RT:23456:22001 Route-Import:11.11.11.11:2001 ENCAP:8
          Router MAC:003a.7d7d.1dbd

  Path-id 1 not advertised to any peer
```

The following example shows how to display BGP information for the IPv4 unicast address family:

```
switch# show bgp ipv4 unicast 11.1.1.0 vrf cust_1
BGP routing table information for VRF cust_1, address family IPv4 Unicast
BGP routing table entry for 11.1.1.0/24, version 4
Paths: (3 available, best #1)
Flags: (0x80080012) on xmit-list, is in urib, is backup urib route, is in HW
  vpn: version 1093, (0x100002) on xmit-list
Multipath: eBGP iBGP

  Advertised path-id 1, VPN AF advertised path-id 1
  Path type: external, path is valid, is best path, in rib
              Imported from 13.13.13.13:3:[5]:[0]:[0]:[24]:[11.1.1.0]/224
  AS-Path: 2000000 500000 , path sourced external to AS
    11.1.1.100 (metric 5) from 102.102.102.102 (102.102.102.102)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 22001
      Received path-id 1
      Extcommunity: RT:23456:22001 Route-Import:22.22.22.22:2001 ENCAP:8
          Router MAC:e00e.da4a.62a5

  Path type: external, path is valid, not best reason: Neighbor Address, multipath, in rib
              Imported from 13.13.13.13:3:[5]:[0]:[0]:[24]:[11.1.1.0]/224
  AS-Path: 2000000 100 , path sourced external to AS
    11.1.1.233 (metric 5) from 102.102.102.102 (102.102.102.102)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 22001
      Received path-id 2
      Extcommunity: RT:23456:22001 Route-Import:33.33.33.33:2001 ENCAP:8
          Router MAC:e00e.da4a.589d

  Path type: external, path is valid, not best reason: Neighbor Address, multipath, in rib
              Imported from 13.13.13.13:3:[5]:[0]:[0]:[24]:[11.1.1.0]/224
  AS-Path: 2000000 100000 , path sourced external to AS
    11.1.1.133 (metric 5) from 102.102.102.102 (102.102.102.102)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 22001
      Received path-id 3
      Extcommunity: RT:23456:22001 Route-Import:11.11.11.11:2001 ENCAP:8
          Router MAC:003a.7d7d.1dbd

  VRF advertise information:
  Path-id 1 not advertised to any peer

  VPN AF advertise information:
  Path-id 1 not advertised to any peer
```

The following example shows how to display routes from the unicast RIB after the Proportional Multipath for VNF feature has been configured:

```
switch# show ip route 1.1.1.0 vrf cust_1
IP Route Table for VRF "cust_1"
…
1.1.1.0/24, ubest/mbest: 22/0, all-best (0x300003d)
    *via 3.0.0.1, [1/0], 08:13:17, static
         recursive next hop: 3.0.0.1/32
    *via 3.0.0.2, [1/0], 08:13:17, static
         recursive next hop: 3.0.0.2/32
    *via 3.0.0.3, [1/0], 08:13:16, static
         recursive next hop: 3.0.0.3/32
    *via 3.0.0.4, [1/0], 08:13:16, static
         recursive next hop: 3.0.0.4/32
   *via 2.0.0.1, [200/0], 06:09:19, bgp-2, internal, tag 2 (evpn) segid: 3003802 tunnelid:
 0x300003e encap: VXLAN
          BGP-EVPN: VNI=3003802 (EVPN)
         client-specific data: 3b
         recursive next hop: 2.0.0.1/32
         extended route information: BGP origin AS 2 BGP peer AS 2
   *via 2.0.0.2, [200/0], 06:09:19, bgp-2, internal, tag 2 (evpn) segid: 3003802 tunnelid:
 0x300003e encap: VXLAN
          BGP-EVPN: VNI=3003802 (EVPN)
         client-specific data: 3b
         recursive next hop: 2.0.0.2/32
         extended route information: BGP origin AS 2 BGP peer AS 2
```

Refer the topologies in the figures given below for the sample outputs in this section.

Figure 4: Sample Topology - IPv4

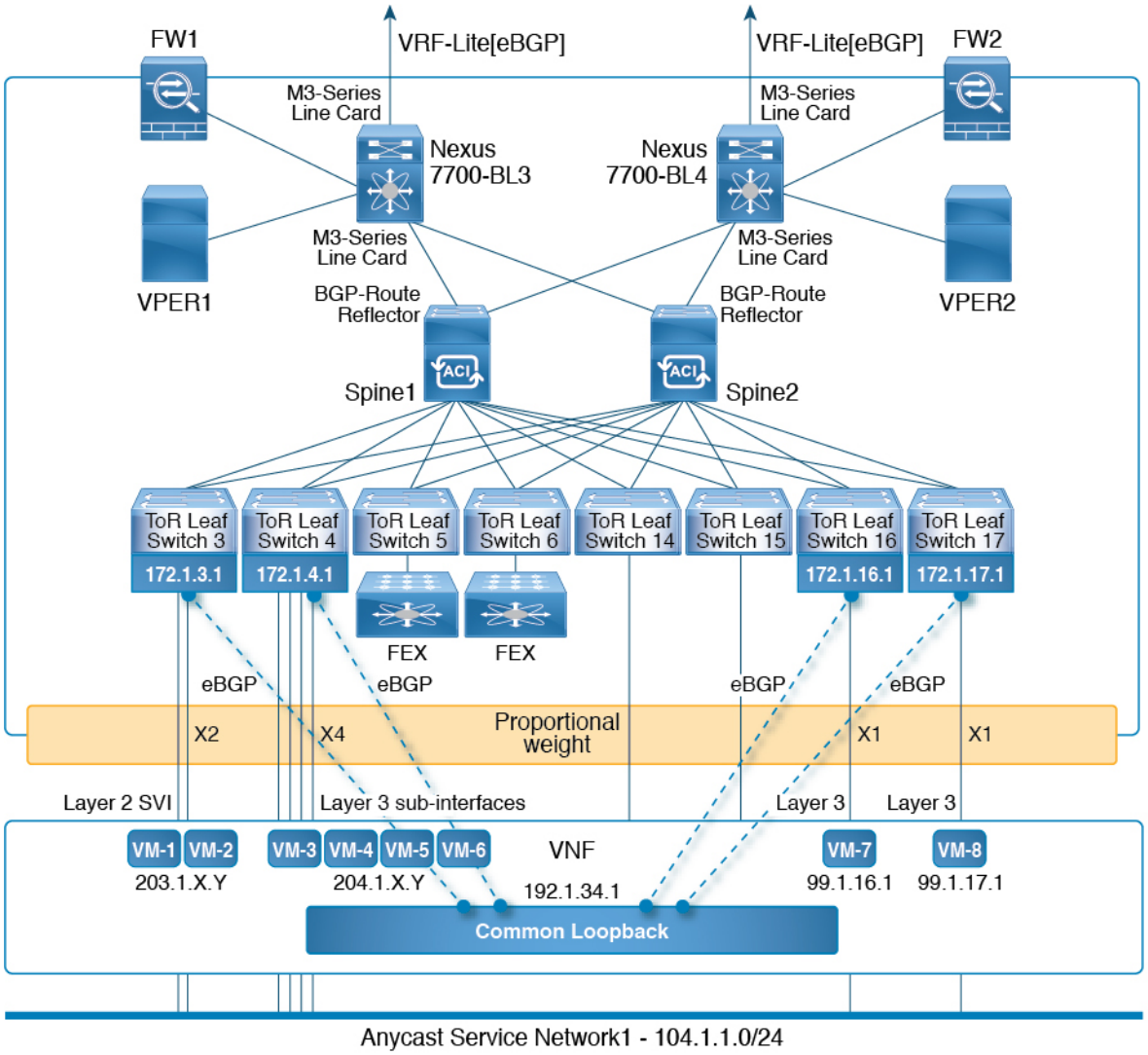
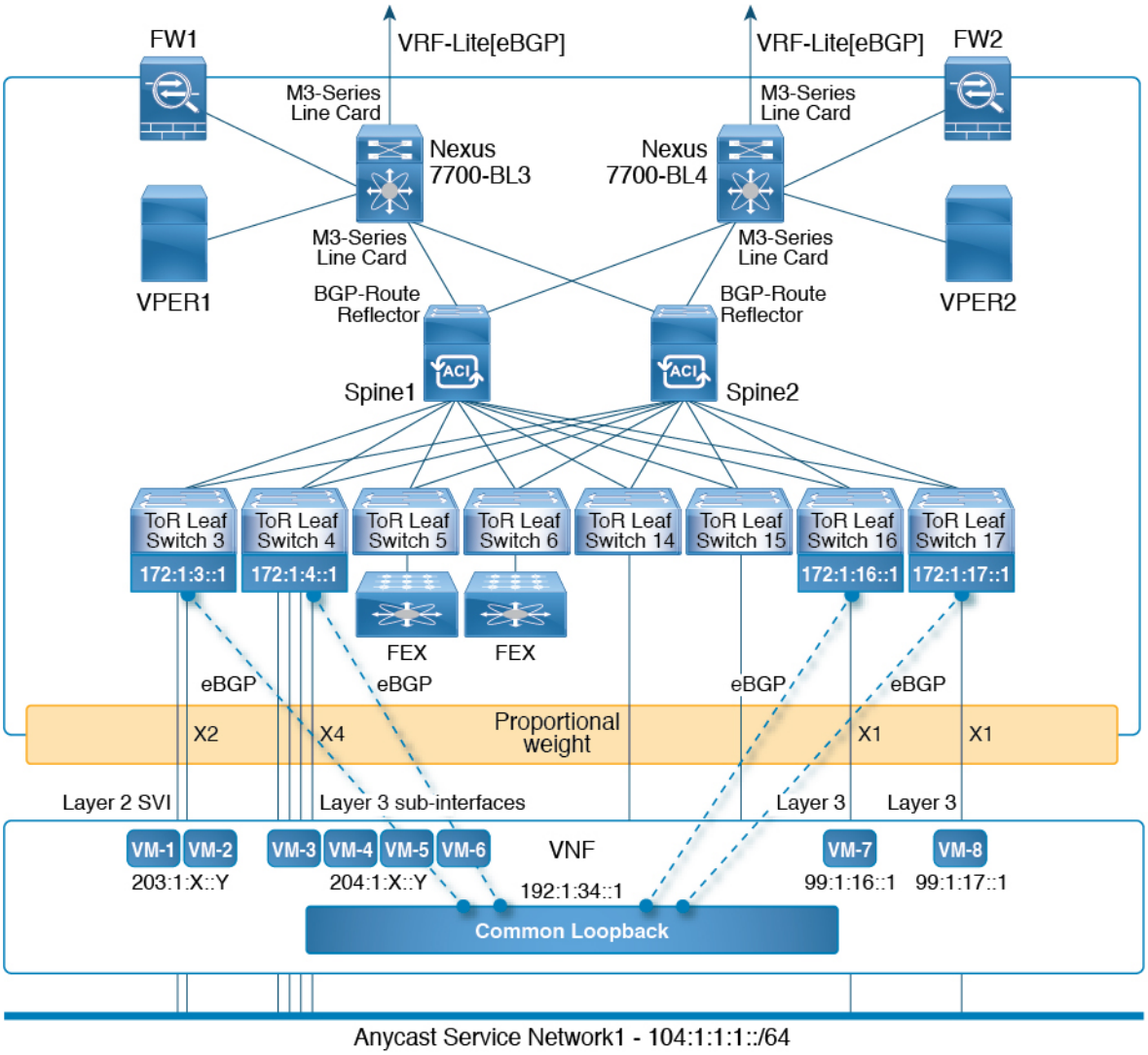


Figure 5: Sample Topology - IPv6



The following example shows how to display the common loopback path for IPv4 addressing scenarios:

```
Nexus-7700-BL3# show ip route 104.1.1.0 vrf os-vrf100
104.1.1.0/24, ubest/mbest: 1/0 time, all-best (0xb011801)
*via 192.1.34.1, [200/0], 12:09:06, bgp-65000, internal, tag 99 (evpn), segid: 10002
tunnelid: 0xb010301 encap: VXLAN
```

The following example shows how to display the total number of recursive next hop (RNH) ECMP paths that point to the gateway IPv4 address in the unicast routing information base (URIB):

```
Nexus-7700-BL3# show ip route 192.1.34.1 vrf os-vrf100
192.1.34.1/32, ubest/mbest: 8/0 time, all-best (0xb011801)
*via 203.1.3.2, [200/0], 12:11:39, bgp-65000, internal, tag 65000 (evpn), segid:
10002 tunnelid: 0xb010301 encap: VXLAN
*via 203.1.4.2, [200/0], 12:11:39, bgp-65000, internal, tag 65000 (evpn), segid:
10002 tunnelid: 0xb010301 encap: VXLAN
*via 204.1.5.2, [200/0], 12:11:39, bgp-65000, internal, tag 65000 (evpn), segid:
10002 tunnelid: 0xb010401 encap: VXLAN
*via 204.1.6.2, [200/0], 12:11:39, bgp-65000, internal, tag 65000 (evpn), segid:
```

```
10002 tunnelid: 0xb010401 encap: VXLAN
*via 204.1.7.2, [200/0], 12:11:39, bgp-65000, internal, tag 65000 (evpn), segid:
10002 tunnelid: 0xb010401 encap: VXLAN
*via 204.1.8.2, [200/0], 12:11:39, bgp-65000, internal, tag 65000 (evpn), segid:
10002 tunnelid: 0xb010401 encap: VXLAN
*via 99.1.16.1, [200/0], 12:11:39, bgp-65000, internal, tag 65000 (evpn), segid:
10002 tunnelid: 0xb011001 encap: VXLAN
*via 99.1.17.1, [200/0], 12:11:39, bgp-65000, internal, tag 65000 (evpn), segid:
10002 tunnelid: 0xb011101 encap: VXLAN
```

The following example shows how to display the total number of RNH ECMP paths that point to the gateway IPv4 address and are flattened in the unicast forwarding information base (UFIB):

```
Nexus-7700-BL3# show forwarding route 104.1.1.0 vrf os-vrf100
slot  9
=======


IPv4 routes for table os-vrf100/base

'*' denotes recursive route
---------------+---------------------------------------+---------------------+-----------------
Prefix         | Next-hop                              | Interface           | Labels
---------------+---------------------------------------+---------------------+-----------------
104.1.1.0/24         11.1.3.1          nve1               (tunnel 0xb010301)
                     11.1.3.1          nve1               (tunnel 0xb010301)
                     11.1.4.1          nve1               (tunnel 0xb010401)
                     11.1.4.1          nve1               (tunnel 0xb010401)
                     11.1.4.1          nve1               (tunnel 0xb010401)
                     11.1.4.1          nve1               (tunnel 0xb010401)
                     11.1.16.1         nve1               (tunnel 0xb011001)
                     11.1.17.1         nve1               (tunnel 0xb011101)
```

The following example shows how to display the common loopback path for IPv6 addressing scenarios:

```
Nexus-7700-BL3# show ipv6 route 104:1:1:1:: vrf os-vrf100
104:1:1:1::/64, ubest/mbest: 1/0, all-best (0xb011801)
*via 192:1:34::1/128, [200/0], 12:19:00, bgp-65000, internal, tag 99 (evpn), segid
10002 tunnel: 0xb010301 encap: VXLAN
```

The following example shows how to display the total number of RNH ECMP paths that point to the gateway IPv6 address in the URIB:

```
Nexus-7700-BL3# show ipv6 route 192:1:34::1 vrf os-vrf100
192:1:34::1/128, ubest/mbest: 8/0, all-best (0xb011801)
*via 203:1:3::2/128, [200/0], 12:20:06, bgp-65000, internal, tag 65000 (evpn),
segid 10002 tunnel: 0xb010301 encap: VXLAN
*via 203:1:4::2/128, [200/0], 12:20:06, bgp-65000, internal, tag 65000 (evpn),
segid 10002 tunnel: 0xb010301 encap: VXLAN
*via 204:1:5::2/128, [200/0], 12:40:59, bgp-65000, internal, tag 65000 (evpn),
segid 10002 tunnel: 0xb010401 encap: VXLAN
*via 204:1:6::2/128, [200/0], 12:40:59, bgp-65000, internal, tag 65000 (evpn),
segid 10002 tunnel: 0xb010401 encap: VXLAN
*via 204:1:7::2/128, [200/0], 12:40:59, bgp-65000, internal, tag 65000 (evpn),
segid 10002 tunnel: 0xb010401 encap: VXLAN
*via 204:1:8::2/128, [200/0], 12:40:59, bgp-65000, internal, tag 65000 (evpn),
segid 10002 tunnel: 0xb010401 encap: VXLAN
*via 99:1:16::1/128, [200/0], 12:40:38, bgp-65000, internal, tag 65000 (evpn),
segid 10002 tunnel: 0xb011001 encap: VXLAN
*via 99:1:17::1/128, [200/0], 12:40:45, bgp-65000, internal, tag 65000 (evpn),
segid 10002 tunnel: 0xb011101 encap: VXLAN
```

The following example shows how to display the total number of RNH ECMP paths that point to the gateway IPv6 address and are flattened in the unicast IPv6 forwarding information base (U6FIB):

```
Nexus-7700-BL3# show forwarding ipv6 route 104:1:1:1:: vrf os-vrf100

slot  9
=======


IPv6 routes for table os-vrf100/base

'*' denotes recursive route
-------------------+---------------------------------------+---------------------+----------------
Prefix             | Next-hop                              | Interface           | Labels
-------------------+---------------------------------------+---------------------+----------------
104:1:1:1::/64       11.1.3.1          nve1                (tunnel 0xb010301)
                     11.1.3.1          nve1                (tunnel 0xb010301)
                     11.1.4.1          nve1                (tunnel 0xb010401)
                     11.1.4.1          nve1                (tunnel 0xb010401)
                     11.1.4.1          nve1                (tunnel 0xb010401)
                     11.1.4.1          nve1                (tunnel 0xb010401)
                     11.1.16.1         nve1                (tunnel 0xb011001)
                     11.1.17.1         nve1                (tunnel 0xb011101)
```

The following example shows how to display the IPv4 BGP EVPN route type5 information in a VRF. The sample output shows that the multipath mode is mixed along with the Gateway IP address that is used.

```
Nexus-7700-BL3# show bgp l2vpn evpn 104.1.1.0
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 11.0.3.1:29
BGP routing table entry for [5]:[0]:[0]:[24]:[104.1.1.0]/224, version 1715444
Paths: (2 available, best #2)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: Mixed

  Path type: internal, path is valid, not best reason: Neighbor Address, no labeled nexthop

  Gateway IP: 192.1.34.1
  AS-Path: 99 98 , path sourced external to AS
    11.1.3.1 (metric 3) from 10.1.2.1 (10.1.2.1)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10002
      Received path-id 1
      Extcommunity: RT:65000:51300 ENCAP:8 Router MAC:2cd0.2d56.931f
      Originator: 11.0.3.1 Cluster list: 10.1.2.1

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
             Imported to 2 destination(s)
             Imported paths list: issu300 default

  Gateway IP: 192.1.34.1
  AS-Path: 99 98 , path sourced external to AS
    11.1.3.1 (metric 3) from 10.1.1.1 (10.1.1.1)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10002
      Received path-id 1
      Extcommunity: RT:65000:51300 ENCAP:8 Router MAC:2cd0.2d56.931f
      Originator: 11.0.3.1 Cluster list: 10.1.1.1
```

The following example shows how to display the IPv6 BGP EVPN route type5 information in a VRF. The sample output shows that the multipath mode is mixed along with the Gateway IPv6 address that is used.

```
Nexus-7700-BL3# show bgp l2vpn evpn 104:1:1:1::
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 11.0.3.1:29
BGP routing table entry for [5]:[0]:[0]:[64]:[104:1:1:1::]/416, version 1712190
Paths: (2 available, best #2)
```

```
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: Mixed

  Path type: internal, path is valid, not best reason: Neighbor Address, no labeled nexthop

  Gateway IP: 192:1:34::1
  AS-Path: 99 98 , path sourced external to AS
    11.1.3.1 (metric 3) from 10.1.2.1 (10.1.2.1)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10002
      Received path-id 1
      Extcommunity: RT:65000:51300 ENCAP:8 Router MAC:2cd0.2d56.931f
      Originator: 11.0.3.1 Cluster list: 10.1.2.1

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
             Imported to 2 destination(s)
             Imported paths list: issu300 default

  Gateway IP: 192:1:34::1
  AS-Path: 99 98 , path sourced external to AS
    11.1.3.1 (metric 3) from 10.1.1.1 (10.1.1.1)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10002
      Received path-id 1
      Extcommunity: RT:65000:51300 ENCAP:8 Router MAC:2cd0.2d56.931f
      Originator: 11.0.3.1 Cluster list: 10.1.1.1
```

The following example displays IPv4 Multipath hashing information. This will display the route selected for a particular source and destination address along with the VTEP and the egress interface.

```
Nexus-7700-BL3# show routing hash 200.1.1.1 104.1.1.1 vrf os-vrf100
Load-share parameters used for software forwarding:
load-share mode: address source-destination
Universal-id seed: 0xffffffff
Hash for VRF "os-vrf100"
Hash Type is 1
Hashing to path *11.1.4.1 Eth10/24%default
  MPLS[2]: Label=10002 E=0 TTL=0 S=0
  MPLS[1]: Label=10002 E=0 TTL=0 S=0
  MPLS[0]: Label=10002 E=0 TTL=0 S=0
For route:
104.1.1.0/24, ubest/mbest: 1/0 time, all-best (0xb011801)
    *via 192.1.34.1, [200/0], 12:41:23, bgp-65000, internal, tag 99 (evpn), segid: 10002
tunnelid: 0xb010301 encap: VXLAN
```

The following example displays IPv6 Multipath hashing information. This will display the route selected for a particular source and destination address along with the VTEP and the egress interface.

```
Nexus-7700-BL3# show routing ipv6 hash 200:1:1:1::1 104:1:1:1::1 vrf os-vrf100
Load-share parameters used for software forwarding:
load-share mode: address source-destination
Universal-id seed: 0xffffffff
No IP protocol specified, defaulting to UDP
Hash for VRF "os-vrf100"
Hash Type is 1
Hashing to path *::ffff:11.1.14.1 Ethernet10/24
  MPLS[1]: Label=10002 E=0 TTL=0 S=0
  MPLS[0]: Label=10002 E=0 TTL=0 S=0
For route:
104:1:1:1::/64, ubest/mbest: 26/0, all-best (0xb011801)
 *via 192:1:34::1/128, [200/0], 00:01:32, bgp-65000, internal, tag 99 (evpn), segid 10002
tunnel: 0xb010301 encap: VXLAN
```

The following example shows how to display the common loopback path for IPv4 addessing scenarios:

```
ToR-Leaf-Switch-3# show ip route 104.1.1.0 vrf os-vrf100
104.1.1.0/24, ubest/mbest: 1/0, all-best (0xb010301)
*via 192.1.34.1, [20/0], 13:24:29, bgp-65000, external, tag 99
```

The following example shows how to display the total number of local and remote RNH ECMP paths that point to the gateway IPv4 address in the URIB:

```
ToR-Leaf-Switch-3# show ip route 192.1.34.1 vrf os-vrf100
192.1.34.1/32, ubest/mbest: 8/0, all-best (0xb010301)
    *via 203.1.3.2, Vlan3, [1/0], 14:46:05, static
    *via 203.1.4.2, Vlan4, [1/0], 14:46:04, static
    *via 204.1.5.2, [200/0], 01:53:23, bgp-65000, internal, tag 65000 (evpn) segid: 10002
tunnelid: 0xb010401 encap: VXLAN

    *via 204.1.6.2, [200/0], 01:52:08, bgp-65000, internal, tag 65000 (evpn) segid: 10002
tunnelid: 0xb010401 encap: VXLAN

    *via 204.1.7.2, [200/0], 01:53:23, bgp-65000, internal, tag 65000 (evpn) segid: 10002
tunnelid: 0xb010401 encap: VXLAN

    *via 204.1.8.2, [200/0], 01:52:08, bgp-65000, internal, tag 65000 (evpn) segid: 10002
tunnelid: 0xb010401 encap: VXLAN

    *via 99.1.16.1, [200/0], 14:45:14, bgp-65000, internal, tag 65000 (evpn) segid: 10002
tunnelid: 0xb011001 encap: VXLAN

    *via 99.1.17.1, [200/0], 14:45:14, bgp-65000, internal, tag 65000 (evpn) segid: 10002
tunnelid: 0xb011101 encap: VXLAN
```

The following example shows how to display the total number of local and remote RNH ECMP paths that point to the gateway IPv4 address in the UFIB:

```
ToR-Leaf-Switch-3# show forwarding route 104.1.1.0 vrf os-vrf100

slot  1
=======


IPv4 routes for table os-vrf100/base

------------------+-----------------------------------+--------------------+---------------+---------------
Prefix            | Next-hop                          | Interface          | Labels
          | Partial Install
------------------+-----------------------------------+--------------------+---------------+---------------
104.1.1.0/24
                     203.1.3.2                            Vlan3
                     203.1.4.2                            Vlan4
                     11.1.4.1          nve1
                     11.1.4.1          nve1
                     11.1.4.1          nve1
                     11.1.4.1          nve1
                     11.1.16.1         nve1
                     11.1.17.1         nve1
```

The following example shows how to display the common loopback path for IPv6 addressing scenarios:

```
ToR-Leaf-Switch-3# show ipv6 route 104:1:1:1:: vrf os-vrf100
104:1:1:1::/64, ubest/mbest: 1/0, all-best (0xb010301)
*via 192:1:34::1/128, [20/0], 02:04:49, bgp-65000, external, tag 99
```

The following example shows how to display the total number of local and remote RNH ECMP paths that point to the gateway IPv6 address in the URIB:

```
ToR-Leaf-Switch-3# show ipv6 route 192:1:34::1 vrf os-vrf100

192:1:34::1/128, ubest/mbest: 8/0, all-best (0xb010301)
```

```
    *via 203:1:3::2, Vlan3, [1/0], 15:00:43, static
    *via 203:1:4::2, Vlan4, [1/0], 15:00:42, static
    *via 204:1:8::2/128, [200/0], 02:06:54, bgp-65000, internal, tag 65000 (evpn) segid
10002 tunnel: 0xb010401 encap: VXLAN

    *via 204:1:7::2/128, [200/0], 02:06:54, bgp-65000, internal, tag 65000 (evpn) segid
10002 tunnel: 0xb010401 encap: VXLAN

    *via 204:1:5::2/128, [200/0], 02:08:09, bgp-65000, internal, tag 65000 (evpn) segid
10002 tunnel: 0xb010401 encap: VXLAN

    *via 204:1:6::2/128, [200/0], 02:08:09, bgp-65000, internal, tag 65000 (evpn) segid
10002 tunnel: 0xb010401 encap: VXLAN

    *via 99:1:16::1/128, [200/0], 02:08:09, bgp-65000, internal, tag 65000 (evpn) segid
10002 tunnel: 0xb011001 encap: VXLAN

    *via 99:1:17::1/128, [200/0], 02:08:09, bgp-65000, internal, tag 65000 (evpn) segid
10002 tunnel: 0xb011101 encap: VXLAN
```

The following example shows how to display the total number of local and remote RNH ECMP paths that point to the gateway IPv6 address in the U6FIB:

```
N93K-EX-LF3# show forwarding ipv6 route 104:1:1:1:: vrf os-vrf100

slot  1
=======


IPv6 routes for table os-vrf100/base

---------------+-----------------------------------+------------------+---------------+---------------
Prefix            | Next-hop                                  | Interface            | Labels
          | Partial Install
---------------+-----------------------------------+------------------+---------------+---------------
104:1:1:1::/64
                     203:1:3::2                                  Vlan3
                     203:1:4::2                                  Vlan4
                     11.1.4.1               nve1
                     11.1.4.1               nve1
                     11.1.4.1               nve1
                     11.1.4.1               nve1
                     11.1.16.1              nve1
                     11.1.17.1              nve1
```

The following example shows how to display the IPv4 BGP EVPN route type5 information in a VRF. The sample output shows that the multipath mode is mixed along with the Gateway IP address that is used.

```
ToR-Leaf-Switch-3# show bgp l2vpn evpn 104.1.1.0
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 11.0.4.1:28
BGP routing table entry for [5]:[0]:[0]:[24]:[104.1.1.0]/224, version 26579
Paths: (2 available, best #2)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: Mixed

  Path type: internal, path is valid, not best reason: Neighbor Address, no labeled nexthop

  Gateway IP: 192.1.34.1
  AS-Path: 99 98 , path sourced external to AS
    11.1.4.1 (metric 3) from 10.1.2.1 (10.1.2.1)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10002
      Received path-id 1
```

```
    Extcommunity: RT:65000:51300 ENCAP:8 Router MAC:2cd0.2d56.918f
    Originator: 0.0.0.0 Cluster list: 10.1.2.1

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
             Imported to 2 destination(s)
             Imported paths list: issu300 default
  Gateway IP: 192.1.34.1
  AS-Path: 99 98 , path sourced external to AS
    11.1.4.1 (metric 3) from 10.1.1.1 (10.1.1.1)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10002
      Received path-id 1
      Extcommunity: RT:65000:51300 ENCAP:8 Router MAC:2cd0.2d56.918f
      Originator: 0.0.0.0 Cluster list: 10.1.1.1
```

The following example shows how to display the IPv6 BGP EVPN route type5 information in a VRF. The sample output shows that the multipath mode is mixed along with the Gateway IP address that is used.

```
ToR-Leaf-Switch-3# show bgp l2vpn evpn 104:1:1:1::
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 11.0.4.1:28
BGP routing table entry for [5]:[0]:[0]:[64]:[104:1:1:1::]/416, version 21947
Paths: (2 available, best #1)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: Mixed

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
             Imported to 2 destination(s)
             Imported paths list: issu300 default
  Gateway IP: 192:1:34::1
  AS-Path: 99 98 , path sourced external to AS
    11.1.4.1 (metric 3) from 10.1.1.1 (10.1.1.1)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 51300
      Received path-id 1
      Extcommunity: RT:65000:51300 ENCAP:8 Router MAC:2cd0.2d56.918f
      Originator: 0.0.0.0 Cluster list: 10.1.1.1

  Path type: internal, path is valid, not best reason: Neighbor Address, no labeled nexthop

  Gateway IP: 192:1:34::1
  AS-Path: 99 98 , path sourced external to AS
    11.1.4.1 (metric 3) from 10.1.2.1 (10.1.2.1)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 51300
      Received path-id 1
      Extcommunity: RT:65000:51300 ENCAP:8 Router MAC:2cd0.2d56.918f
      Originator: 0.0.0.0 Cluster list: 10.1.2.1
```

The following example displays IPv4 Multipath hashing information. This will display the route selected for a particular source and destination address along with the VTEP and the egress interface. In this example, a local path is selected. This indicates that the border leaf and the access leaf have the same hash result.

```
ToR-Leaf-Switch-3# show routing hash 200.1.1.1 104.1.1.1 vrf os-vrf100
Load-share parameters used for software forwarding:
load-share mode: address source-destination gtpu-teid
No IPv4 protocol specified, defaulting to UDP
Hash for VRF "os-vrf100"
Hashing to path *203.1.3.2
    Out Interface: Vlan3

  MPLS[1]: Label=10002 E=0 TTL=0 S=0
  MPLS[0]: Label=10002 E=0 TTL=0 S=0
```

```
For route:
104.1.1.0/24, ubest/mbest: 1/0, all-best (0xb010301)
    *via 192.1.34.1, [20/0], 15:39:16, bgp-65000, external, tag 99
```

The following example also displays IPv4 Multipath hashing information. In this example, a remote path is selected. This indicates that the border leaf and the access leaf have different hash results.

```
ToR-Leaf-Switch-LF17# show routing hash 200.1.2.1 104.1.2.1 vrf os-vrf100
Load-share parameters used for software forwarding:
load-share mode: address source-destination gtpu-teid
No IPv4 protocol specified, defaulting to UDP
Hash for VRF "os-vrf100"
Hashing to path *11.1.16.1
    Out Interface: Eth1/50

  MPLS[1]: Label=10002 E=0 TTL=0 S=0
  MPLS[0]: Label=10002 E=0 TTL=0 S=0
For route:
104.1.2.0/24, ubest/mbest: 1/0, all-best (0xb011101)
    *via 192.1.34.1, [20/0], 15:54:05, bgp-65000, external, tag 99
```

Now, go to Leaf 16 and and use the **show routing hash** command. The sample output is as given below.

```
ToR-Leaf-Switch-LF16# show routing hash 200.1.2.1 104.1.2.1 vrf os-vrf100
Load-share parameters used for software forwarding:
load-share mode: address source-destination gtpu-teid
No IPv4 protocol specified, defaulting to UDP
Hash for VRF "os-vrf100"
Hashing to path *99.1.16.1
    Out Interface: Eth1/48

  MPLS[1]: Label=10002 E=0 TTL=0 S=0
  MPLS[0]: Label=10002 E=0 TTL=0 S=0
For route:
104.1.2.0/24, ubest/mbest: 1/0, all-best (0xb011001)
    *via 192.1.34.1, [20/0], 15:58:13, bgp-65000, external, tag 99
```

The following example displays IPv6 Multipath hashing information. This will display the route selected for a particular source and destination address along with the VTEP and the egress interface. In this example, a local path is selected. This indicates that the border leaf and the access leaf have the same hash result.

```
ToR-Leaf-Switch-3# show routing ipv6 hash 200:1:1:1::1 104:1:1:1::1 vrf os-vrf100
Load-share parameters used for software forwarding:
load-share mode: 6
No IP protocol specified, defaulting to UDP

Hash for VRF "os-vrf100"
Hash Type is 3
Hashing to path *203:1:3::2
Out Interface  Vlan3
For route:
104:1:1:1::/64, ubest/mbest: 1/0, all-best (0xb010301)
    *via 192:1:34::1/128, [20/0], 03:08:55, bgp-65000, external, tag 99
```

The following example also displays IPv6 Multipath hashing information. In this example, a remote path is selected. This indicates that the border leaf and the access leaf have different hash results.

```
ToR-Leaf-Switch-LF17#  show routing ipv6 hash 200:1:1:2::1 104:1:1:2::1 vrf os-vrf100
Load-share parameters used for software forwarding:
load-share mode: 6
No IP protocol specified, defaulting to UDP

Hash for VRF "os-vrf100"
Hash Type is 3
Hashing to path *::ffff:11.1.3.1
```

```
Out Interface  Ethernet1/50
For route:
104:1:1:2::/64, ubest/mbest: 1/0, all-best (0xb011101)
    *via 192:1:34::1/128, [20/0], 03:13:12, bgp-65000, external, tag 99
```

Now, go to Leaf 3 and and use the **show routing ipv6 hash** command. The sample output is as given below.

```
ToR-Leaf-Switch-LF3# show routing ipv6 hash 200:1:1:2::1 104:1:1:2::1 vrf os-vrf100

Load-share parameters used for software forwarding:
load-share mode: 6
No IP protocol specified, defaulting to UDP

Hash for VRF "os-vrf100"
Hash Type is 3
Hashing to path *203:1:3::2
Out Interface  Vlan3
For route:
104:1:1:2::/64, ubest/mbest: 1/0, all-best (0xb010301)
    *via 192:1:34::1/128, [20/0], 03:19:09, bgp-65000, external, tag 99
```

Configuration Examples for Proportional Multipath for VNF

This example shows a running BGP TOR configuration to enable BGP peering with the spine in a different autonomous system.

```
vrf context cust_1
  vni 22001
  rd auto
  address-family ipv4 unicast
    route-target import 23456:22001
    route-target import 23456:22001 evpn
    route-target export 23456:22001
    route-target export 23456:22001 evpn
    route-target both auto
    route-target both auto evpn

    ip route 200.1.1.1 via 10.1.1.1 tag 100
    ip route 200.1.1.1 via 10.1.2.1 tag 100

  route-map REDIST 10 permit 10
  route-map passall permit 10
    set path-selection all advertise

router bgp 500000
  address-family l2vpn evpn
    maximum-paths mixed 32
    additional-paths send
    additional-paths receive
    additional-paths selection route-map passall

  neighbor 102.102.102.102
    remote-as 2000000
    update-source loopback0
    ebgp-multihop 255
    address-family ipv4 unicast
      allowas-in 3
      send-community extended
    address-family l2vpn evpn
      allowas-in 3
```

```
      send-community extended
  neighbor 105.105.105.105
    remote-as 2000000
    update-source loopback0
    ebgp-multihop 255
    address-family ipv4 unicast
      allowas-in 3
      send-community extended
    address-family l2vpn evpn
      allowas-in 3
      send-community extended
      no advertise-gw-ip

vrf cust_1
  address-family ipv4 unicast
    advertise l2vpn evpn
    wait-igp-convergence
    redistribute direct route-map REDIST
    redistribute static route-map REDIST
    export-gateway-ip
    maximum-paths mixed 32
```

This example shows a running legacy peer configuration.

```
router bgp 2000000
  neighbor 4.4.4.4
    remote-as 500000
    update-source loopback0
    ebgp-multihop 255
    address-family ipv4 unicast
    disable-peer-as-check
    send-community extended

  address-family l2vpn evpn
    disable-peer-as-check
    send-community extended

  neighbor 6.6.6.6
    remote-as 100
    update-source loopback0
    ebgp-multihop 255
    address-family ipv4 unicast
      send-community extended
      address-family ipv4 mvpn
        disable-peer-as-check
        send-community extended
        route-map setnh_unchanged out

    address-family l2vpn evpn
      disable-peer-as-check
      send-community extended
      route-map setnh_unchanged out
      no advertise-gw-ip
```

Additional References for Proportional Multipath for VNF

This section describes additional information related to implementing Proportional Multipath for VNF.

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	Cisco NX-OS Licensing Guide
Verified Scalability	Cisco Nexus 7000 Series NX-OS Verified Scalability Guide

Feature History for Proportional Multipath for VNF

This table lists the release history for this feature.

Feature Name	Release	Information
Proportional Multipath for VNF	8.3(1)	In Network Function Virtualization Infrastructures (NFVi), anycast services networks are advertised from multiple Virtual Network Functions (VNFs). The Proportional Multipath for VNF feature enables advertising of all the available next hops to a given destination network. This feature enables the switch to consider all paths to a given route as equal cost multipath (ECMP) allowing the traffic to be forwarded using all the available links stretched across multiple ToRs.