Cisco Nexus 3000 Series NX-OS VXLAN Configuration Guide, Release 9.3(x)

First Published: 2019-07-20

Last Modified: 2020-10-18

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 3000 Series Switches

The entire Cisco Nexus 3000 Series switch documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/tsd-products-support-series-home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com. We appreciate your feedback.

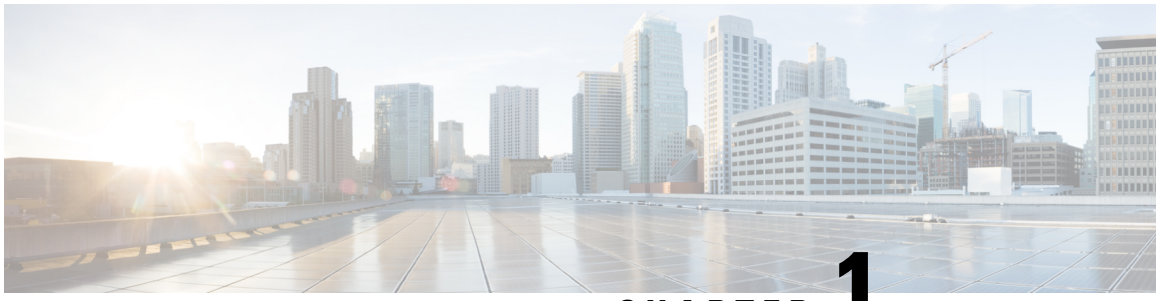
Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Preface



CHAPTER 1

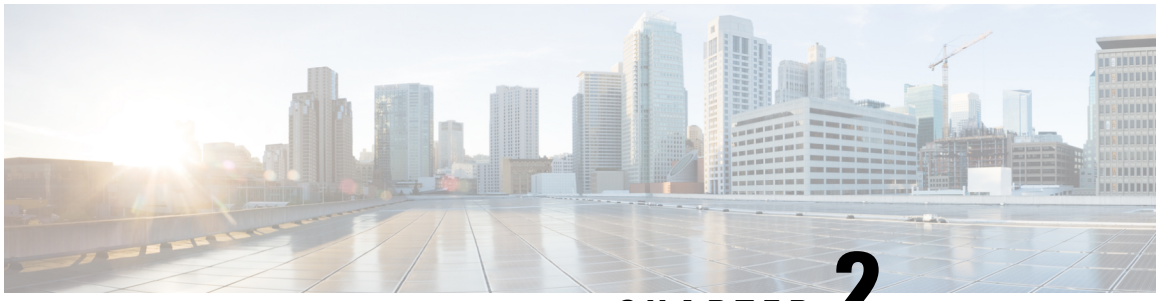
New and Changed Information

This chapter contains the following sections:

New and Changed Information in this Release

The following table provides an overview of the significant changes made to this configuration guide. The table does not provide an exhaustive list of all the changes made to this guide or all new features in a particular release.

Feature	Description	Added or Changed in Release	Where Documented
Tenant Routed Multicast	Added support for the TRM on Cisco Nexus 3132-Z switches	9.3(3)	Configuring Tenant Routed Multicast, on page 79
Initial Release		9.3(1)	



CHAPTER 2

Configuring VXLANs

Overview

VXLAN Overview

The Cisco Nexus 3100 Series switches are designed for a hardware-based Virtual Extensible LAN (VXLAN) function. These switches can extend Layer 2 connectivity across the Layer 3 boundary and integrate between VXLAN and non-VXLAN infrastructures. Virtualized and multitenant data center designs can be shared over a common physical infrastructure.

VXLANs enable you to extend Layer 2 networks across the Layer 3 infrastructure by using MAC-in-UDP encapsulation and tunneling. In addition, you can use a VXLAN to build a multitenant data center by decoupling tenant Layer 2 segments from the shared transport network.

When deployed as a VXLAN gateway, the Cisco Nexus 3100 Series switches can connect VXLAN and classic VLAN segments to create a common forwarding domain so that tenant devices can reside in both environments.

A VXLAN has the following benefits:

- Flexible placement of multitenant segments throughout the data center.

  It extends Layer 2 segments over the underlying shared network infrastructure so that tenant workloads can be placed across physical pods in the data center.

- Higher scalability to address more Layer 2 segments.

  A VXLAN uses a 24-bit segment ID called the VXLAN network identifier (VNID). The VNID allows a maximum of 16 million VXLAN segments to coexist in the same administrative domain. (In comparison, traditional VLANs use a 12-bit segment ID that can support a maximum of 4096 VLANs.)

- Utilization of available network paths in the underlying infrastructure.

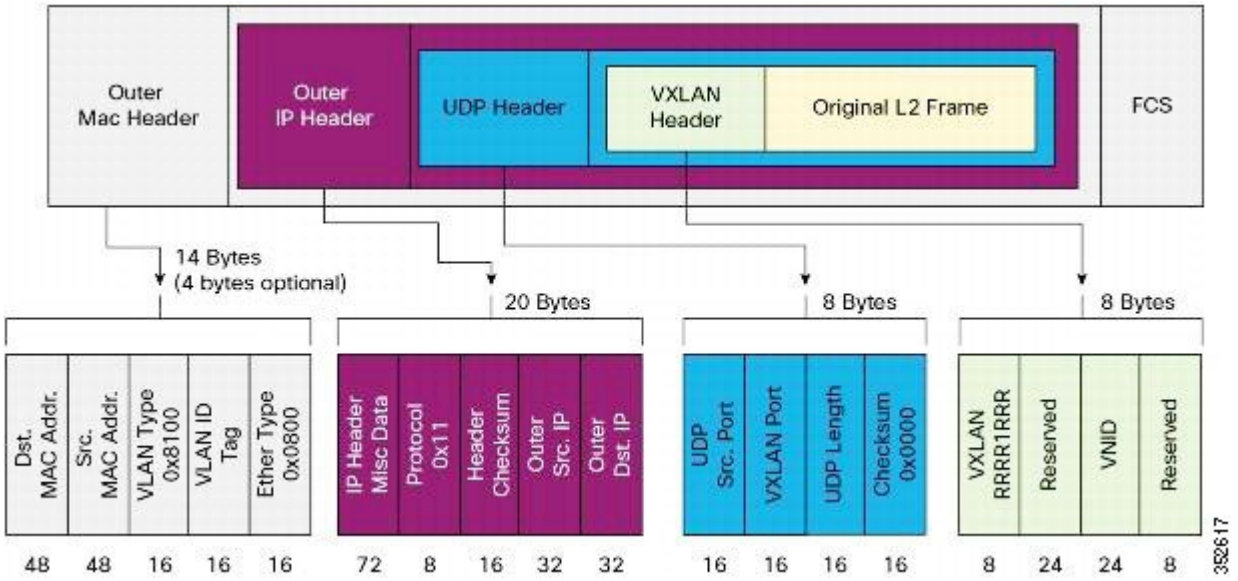
  VXLAN packets are transferred through the underlying network based on its Layer 3 header. It uses equal-cost multipath (ECMP) routing and link aggregation protocols to use all available paths.

VXLAN Encapsulation and Packet Format

A VXLAN is a Layer 2 overlay scheme over a Layer 3 network. It uses MAC-in-UDP encapsulation to extend Layer 2 segments across the data center network. The transport protocol over the physical data center network is IP plus UDP.

A VXLAN defines a MAC-in-UDP encapsulation scheme where the original Layer 2 frame has a VXLAN header added and is then placed in a UDP-IP packet. With this MAC-in-UDP encapsulation, VXLAN tunnels Layer 2 network over the Layer 3 network. The VXLAN packet format is shown in the following figure.

Figure 1: VXLAN Packet Format



A VXLAN uses an 8-byte VXLAN header that consists of a 24-bit VNID and a few reserved bits. The VXLAN header and the original Ethernet frame are in the UDP payload. The 24-bit VNID identifies the Layer 2 segments and maintains Layer 2 isolation between the segments. A VXLAN can support 16 million LAN segments.

VXLAN Tunnel Endpoints

A VXLAN uses VXLAN tunnel endpoint (VTEP) devices to map tenants' end devices to VXLAN segments and to perform VXLAN encapsulation and deencapsulation. Each VTEP device has two types of interfaces:

- Switch port interfaces on the local LAN segment to support local endpoint communication through bridging
- IP interfaces to the transport network where the VXLAN encapsulated frames will be sent

A VTEP device is identified in the IP transport network by using a unique IP address, which is a loopback interface IP address. The VTEP device uses this IP address to encapsulate Ethernet frames and transmits the encapsulated packets to the transport network through the IP interface. A VTEP device learns the remote VTEP IP addresses and the remote MAC address-to-VTEP IP mapping for the VXLAN traffic that it receives.

The VXLAN segments are independent of the underlying network topology; conversely, the underlying IP network between VTEPs is independent of the VXLAN overlay. The IP network routes the encapsulated

packets based on the outer IP address header, which has the initiating VTEP as the source IP address and the terminating VTEP or multicast group IP address as the destination IP address.

VXLAN Packet Forwarding Flow

A VXLAN uses stateless tunnels between VTEPs to transmit traffic of the overlay Layer 2 network through the Layer 3 transport network.

VXLAN Implementation on Cisco Nexus 3100 Platform Switches

The Cisco Nexus 3100 platform switches support the hardware-based VXLAN function that extends Layer 2 connectivity across the Layer 3 transport network and provides a high-performance gateway between VXLAN and non-VXLAN infrastructures.

Layer 2 Mechanisms for Broadcast, Unknown Unicast, and Multicast Traffic

A VXLAN on the Cisco Nexus 3100 platform switches uses flooding and dynamic MAC address learning to do the following:

- Transport broadcast, unknown unicast, and multicast traffic
- Discover remote VTEPs
- Learn remote host MAC addresses and MAC-to-VTEP mappings for each VXLAN segment

A VXLAN can forward these traffic types as follows:

- Using multicast in the core—IP multicast reduces the flooding of the set of hosts that are participating in the VXLAN segment. Each VXLAN segment, or VNID, is mapped to an IP multicast group in the transport IP network. The Layer 2 gateway uses Protocol Independent Multicast (PIM) to send and receive traffic from the rendezvous point (RP) for the IP multicast group. The multicast distribution tree for this group is built through the transport network based on the locations of participating VTEPs.
- Using ingress replication—Each VXLAN segment or VXLAN network identifier (VNI) is mapped to a remote unicast peer. The Layer 2 frame is VXLAN encapsulated with the destination IP address as the remote unicast peer IP address and is sent out to the IP transport network where it gets unicast routed or forwarded to the remote destination.

Layer 2 Mechanisms for Unicast-Learned Traffic

The Cisco Nexus 3100 platform switches perform MAC address lookup-based forwarding for VXLAN unicast-learned traffic.

When Layer 2 traffic is received on the access side, a MAC address lookup is performed for the destination MAC address in the frame. If the lookup is successful, VXLAN forwarding is done based on the information retrieved as a result of the lookup. The lookup result provides the IP address of the remote VTEP from which this MAC address is learned. This Layer 2 frame is then UDP/IP encapsulated with the destination IP address as the remote VTEP IP address and is forwarded out of the appropriate network interface. In the Layer 3 cloud, this IP packet is forwarded to the remote VTEP through the route to that IP address in the network.

For unicast-learned traffic, you must ensure the following:

- The route to the remote peer is known through a routing protocol or through static routes in the network.
- Adjacency is resolved.

VXLAN Layer 2 Gateway as a Transit Multicast Router

A VXLAN Layer 2 gateway must terminate VXLAN-multicast traffic that is headed to any of the groups to which VNIs are mapped. In a network, a VXLAN Layer 2 gateway can be a multicast transit router for the downstream multicast receivers that are interested in the group's traffic. A VXLAN Layer 2 gateway must do some additional processing to ensure that VXLAN multicast traffic that is received is both terminated and multicast routed. This traffic processing is done in two passes:

1. The VXLAN multicast traffic is multicast routed to all network receivers interested in that group's traffic.
2. The VXLAN multicast traffic is terminated, decapsulated, and forwarded to all VXLAN access side ports.

ECMP and LACP Load Sharing with VXLANs

Encapsulated VXLAN packets are forwarded between VTEPs based on the native forwarding decisions of the transport network. Most data center transport networks are designed and deployed with multiple redundant paths that take advantage of various multipath load-sharing technologies to distribute traffic loads on all available paths.

A typical VXLAN transport network is an IP-routing network that uses the standard IP equal cost multipath (ECMP) to balance the traffic load among multiple best paths. To avoid out-of-sequence packet forwarding, flow-based ECMP is commonly deployed. An ECMP flow is defined by the source and destination IP addresses and optionally, the source and destination TCP or UDP ports in the IP packet header.

All the VXLAN packet flows between a pair of VTEPs have the same outer source and destination IP addresses, and all VTEP devices must use one identical destination UDP port that can be either the Internet Allocated Numbers Authority (IANA)-allocated UDP port 4789 or a customer-configured port. The only variable element in the ECMP flow definition that can differentiate VXLAN flows from the transport network standpoint is the source UDP port. A similar situation for Link Aggregation Control Protocol (LACP) hashing occurs if the resolved egress interface that is based on the routing and ECMP decision is an LACP port channel. LACP uses the VXLAN outer-packet header for link load-share hashing, which results in the source UDP port being the only element that can uniquely identify a VXLAN flow.

In the Cisco Nexus 3100 platform switch implementation of VXLANs, a hash of the inner frame's header is used as the VXLAN source UDP port. As a result, a VXLAN flow can be unique. The IP address and UDP port combination is in its outer header while the packet traverses the underlay transport network.

Guidelines and Limitations for VXLANs

VXLAN has the following guidelines and limitations:

- The configuration of the multicast groups and Ingress Replication (IR) is not supported at the same time. You can configure and deploy either multicast groups or IR to deploy VXLAN.
- The **system vlan nve-overlay** command is not required for Cisco Nexus 3000 platform switches with certain types of BroadCom ASICs. Therefore, do not enable the **system vlan nve-overlay** command.

- In VXLAN on vPC configuration, the packets from North VTEP are decapped on the primary vPC switch and they are sent to all ports in the VLAN/VN-segment and they are also forwarded on the multicast link to the secondary vPC switch. Therefore, the NVE VNI counters are observed to increment for both Tx and Rx on the primary vPC switch, whereas the NVE VNI counters increment only for Rx on the secondary vPC switch.
- It is recommended that the summation of the number of the multicast groups and the OIFLs to be used in a scaled environment should not exclude 1024, which is the current range of the multicast VXLAN VP.
- Adjacencies are configured in different regions on an overlay or underlay network for different types of L3 interfaces based on whether or not the VXLAN, VNI or VFI are enabled on the interface. MAC rewrite does not happen if packets sent from a VFI enabled VLAN and hit an adjacency in an underlay network. So routing between VXLAN enabled VLANs and non-VXLAN enabled VLANs or L3 interfaces may fail.
- IGMP snooping is supported on VXLAN VLANs.
- VXLAN routing is supported for only the Cisco Nexus 3100-V platform switches. For other switches, the default Layer 3 gateway for VXLAN VLANs must be provisioned on a different device.
- Ensure that the network can accommodate an additional 50 bytes for the VXLAN header.
- Only one Network Virtualization Edge (NVE) interface is supported on a switch.
- Layer 3 VXLAN uplinks are not supported in a nondefault virtual and routing forwarding (VRF) instance.
- Only one VXLAN IP adjacency is possible per physical interface.
- SVIs over VXLAN VLAN for routing are supported for only the Cisco Nexus 3100-V platform switches.
- Switched Port Analyzer (SPAN) Tx for VXLAN-encapsulated traffic is not supported for the Layer 3 uplink interface.
- Access control lists (ACLs) and quality of service (QoS) for VXLAN traffic to access direction are not supported.
- SNMP is not supported on the NVE interface.
- Native VLANs for VXLAN are not supported.
- For ingress replication configurations, multiple VNIs can now have the same remote peer IP configured.
- The VXLAN source UDP port is determined based on the VNID and source and destination IP addresses.
- The UDP port configuration must be done before the NVE interface is enabled. If the UDP configuration must be changed while the NVE interface is enabled, you must shut down the NVE interface, make the UDP configuration change, and then reenable the NVE interface.

Note The VXLAN UDP port is not configurable on the Cisco Nexus 3100-V platform switches.

- Inter-VNI routing and IGMP snooping for VXLAN-enabled VLANs are not supported on Cisco Nexus 3232C and 3264Q platform switches.

- In a VXLAN EVPN setup that has 2K VNI scale configuration, the control plane downtime takes more than 200 seconds. You must configure the graceful restart time as 300 seconds to avoid BGP flap.
- When a VXLAN-encapsulated ping6 packet is received on a network port, two copies of the packet are sent to the host after decapsulation. This behavior applies to Cisco Nexus 3132Q, 3164Q, 3172PQ, 3172TQ, 3100-V, and 31128PQ platform switches.

FHRP Over VXLAN

Overview of FHRP over VXLAN

Overview of FHRP

Starting with Release 7.0(3)I7(1), you can configure First Hop Redundancy Protocol (FHRP) over VXLAN on Cisco Nexus 3000 Series switches. The FHRP provides a redundant Layer 3 traffic path. It provides fast failure detection and transparent switching of the traffic flow. The FHRP avoids the use of the routing protocols on all the devices. It also avoids the traffic loss that is associated with the routing or the discovery protocol convergence. It provides an election mechanism to determine the next best gateway. Current FHRP supports HSRPv1, HSRPv2, VRRPv2, and VRRPv3.

FHRP over VXLAN

The FHRP serves at the Layer 3 VXLAN redundant gateway for the hosts in the VXLAN. The Layer 3 VXLAN gateway provides routing between the VXLAN segments and routing between the VXLAN to the VLAN segments. Layer 3 VXLAN gateway also serves as a gateway for the external connectivity of the hosts.

Guidelines and Limitations for FHRP Over VXLAN

See the following guidelines and limitations for configuring FHRP over VXLAN:

- When using FHRP with VXLAN, ARP-ETHER TCAM must be carved using the **arp-ether 256 double-wide** CLI command.
- Configuring FHRP over VXLAN is supported for both IR and multicast flooding of the FHRP packets. The FHRP protocol working does not change for configuring FHRP over VXLAN.
- The FHRP over VXLAN feature is supported for flood and learn only.
- For Layer 3 VTEPs in BGP EVPN, only anycast GW is supported.

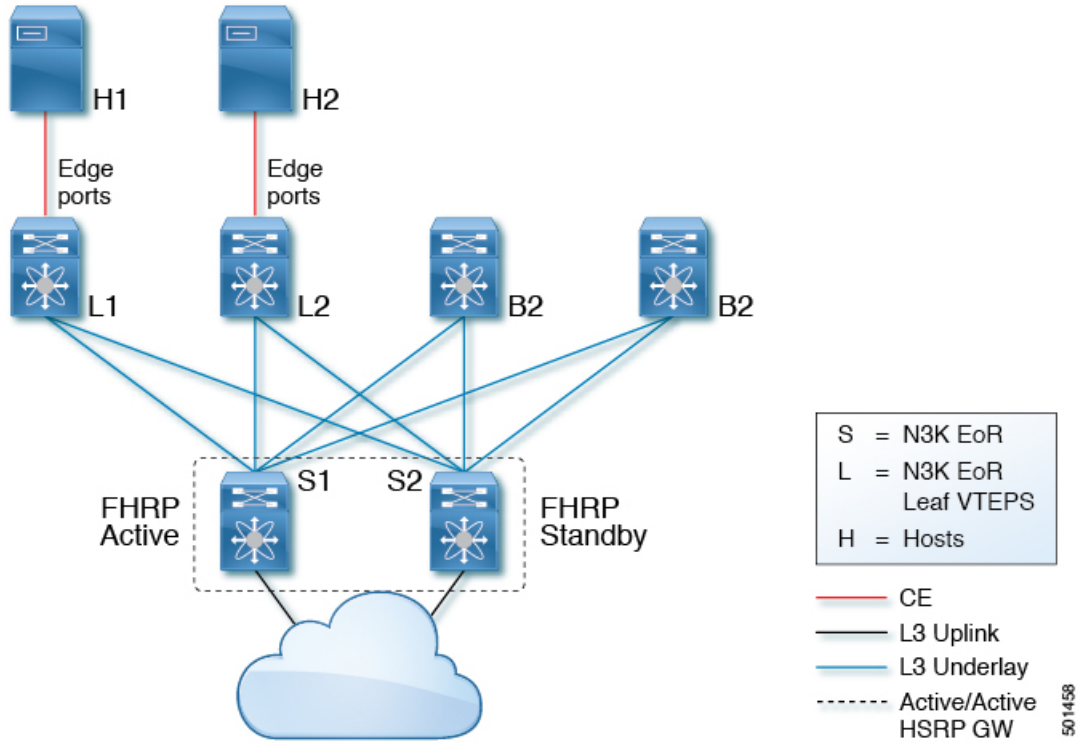
FHRP Over VXLAN Topology

In the following topology, the FHRP is configured on the Spine Layer. The FHRP protocols synchronize its state with the hellos that get flooded on the overlay without having a dedicated Layer 2 link in between the peers. The FHRP operates in an active/standby state as no vPC is being deployed.

Note Bi-Directional Forwarding (BFD) is not supported with HSRP in the new topology.

The following image illustrates the new topology that supports a FHRP over VXLAN configuration:

Figure 2: Configuring FHRP Over VXLAN



Following is the configuration example of the new topology:

```
S1 FHRP configuration with HSRP
# VLAN with VNI
vlan 10
  vn-segment 10000

# Layer-3 Interface with FHRP (HSRP)
interface vlan 10
 ip address 192.168.1.2
  hsrp 10
  ip 192.168.1.1

S2 FHRP configuration with HSRP
# VLAN with VNI
vlan 10
  vn-segment 10000

# Layer-3 Interface with FHRP (HSRP)
interface vlan 10
 ip address 192.168.1.3
  hsrp 10
  ip 192.168.1.1
```

Note The FHRP configuration can leverage HSRP or VRRP. No vPC peer-link is necessary and therefore no VLAN is allowed on the vPC peer-link. The VNI mapped to the VLAN must be configured on the NVE interface and it is associated with the used BUM replication mode (Multicast or Ingress Replication).

Considerations for VXLAN Deployment

The following are some of the considerations while deploying VXLANs:

- A loopback interface IP is used to uniquely identify a VTEP device in the transport network.
- To establish IP multicast routing in the core, an IP multicast configuration, PIM configuration, and Rendezvous Point (RP) configuration are required.
- You can configure VTEP-to-VTEP unicast reachability through any IGP protocol.
- You can configure a VXLAN UDP destination port as required. The default port is 4789.
- The default gateway for VXLAN VLANs should be provisioned on a different upstream router.
- VXLAN multicast traffic should always use the RPT shared tree.
- An RP for the multicast group on the VTEP is a supported configuration. However, you must configure the RP for the multicast group at the spine layer/upstream device. Because all multicast traffic traverses the RP, it is more efficient to have this traffic directed to a spine layer/upstream device.

vPC Guidelines and Limitations for VXLAN Deployment

- The VXLAN multicast encapsulation path has duplicate members of the VPC peer-link on the VPC peers. This design has been adopted to support anycast RP and the service orphan traffic. For all the access side traffic, now two copies of a packet are sent over the VPC peer-link on the multicast path, one native and one VXLAN header encapsulated.
- You must bind NVE to a loopback address that is separate form other loopback addresses required by Layer 3 protocols. Use a dedicated loopback address for VXLAN.
- Multicast traffic on a vPC that is hashed toward the non-DF switch traverses the multichassis EtherChannel trunk (MCT) and is encapsulated on the DF node.
- In a VXLAN vPC, consistency checks are performed to ensure that NVE configurations and VN-Segment configurations are identical across vPC peers.
- The router ID for unicast routing protocols must be different from the loopback IP address used for VTEP.
- Configure an SVI between vPC peers and advertise routes between the vPC peers by using a routing protocol with higher routing metric. This action ensures that the IP connectivity of the vPC node does not go down if one vPC node fails.

Configuration Guidelines for VXLAN VPC Setup and Expected Behaviors in Various Scenarios

- VPC peers must have identical configurations:
  - Consistent VLAN to VN-segment mapping.
  - Consistent NVE1 binding to the same loopback interface.
  - Using the same secondary IP address.
  - Using different primary IP addresses.

- Consistent VNI to group mapping.

- For multicast, the VPC node that receives the (S, G) join from the RP (rendezvous point) becomes the DF (Designated Forwarder). On the DF node, the encapsulation routes are installed for multicast.
- The decap routes are installed based on the election of a decapper from between the VPC primary node and the VPC secondary node. The winner of the decap election is the node with the least cost to the RP.
- However, if the cost to the RP is the same for both nodes, the VPC primary node is elected. The winner of the decap election has the decap mroute installed. The other node does not have a decap route installed.
- On a VPC device, the BUM traffic (broadcast, unknown-unicast, and multicast traffic) from hosts is replicated on the the peer-link. A copy is made of every native packet and each native packet is sent across the peer-link to service the orphan-ports connected to the peer VPC switch.
- To prevent traffic loops in VXLAN networks, native packets ingressing the peer-link cannot be sent to an uplink. However, if the peer switch is the encapper, the copied packet traverses the peer-link and it is sent to the uplink.
- When the peer-link is shut, the loopback address on the VPC secondary is brought down and the status is Admin Shut. This is done so that the route to the loopback is withdrawn on the upstream and that the upstream can divert all the traffic to the VPC primary.

Note Orphans that are connected to the secondary vPC experience a loss of traffic when the MCT is shut down. This situation is similar to Layer 2 orphans in a secondary vPC of a traditional vPC setup.

- When the peer-link is no-shut, the NVE loopback address is brought up again and the route is advertised upstream attracting the traffic.
- For VPC,
  - The loopback interface has 2 IP addresses: the primary IP address and the secondary IP address.
  - The primary IP address is unique and is used by Layer 3 protocols.
  - The secondary IP address on loopback is necessary because the interface NVE uses it for the VTEP IP address.
  - The secondary IP address must be same on both vPC peers.
  - The VPC peer-gateway feature must be enabled on both peers.
- As a best practice, use peer-switch, peer gateway, ip arp sync, ipv6 nd sync configurations for improved convergence in VPC topologies.
- When the NVE or loopback is shut in VPC configurations:
  - If the NVE or loopback is shut only on the primary VPC switch, the global VxLAN VPC consistency checker fails. Then the NVE, loopback, and VPCs are taken down on the secondary VPC switch.
  - If the NVE or loopback is shut only on the secondary VPC switch, the global VXLAN VPC consistency checker fails. Then the NVE, loopback, and secondary VPC are brought down on the secondary. The traffic continues to flow through the primary VPC switch.

- As a best practice, you should keep both the NVE and loopback up on both the primary and secondary VPC switches.
- Redundant anycast RPs configured in the network for multicast load-balancing and RP redundancy are supported on VPC VTEP topologies.
- Enabling vpc peer-switch configuration is mandatory. For peer-switch functionality, at least one SVI is required to be enabled across the peer-link and also configured with PIM. This provides a backup path in the case when VTEP loses complete connectivity to the spine. Remote peer reachability is re-routed over the peer-link in this case.

Configuring VXLAN Traffic Forwarding

There are two options for forwarding broadcast, unknown unicast and multicast traffic on a VXLAN Layer 2 gateway. Layer 2 Mechanisms for Broadcast, Unknown Unicast, and Multicast Traffic, on page 5 provides more information about these two options.

Before you enable and configure VXLANs, ensure that the following configurations are complete:

- For IP multicast in the core, ensure that the IP multicast configuration, the PIM configuration, and the RP configuration are complete, and that a routing protocol exists.
- For ingress replication, ensure that a routing protocol exists for reaching unicast addresses.

Note On a Cisco Nexus 3100 Series switch that functions as a VXLAN Layer 2 gateway, note that traffic that is received on the access side cannot trigger an ARP on the network side. ARP for network side interfaces should be resolved either by using a routing protocol such as BGP, or by using static ARP. This requirement is applicable for ingress replication cases alone, not for multicast replication cases.

Enabling and Configuring the PIM Feature

Before you can access the PIM commands, you must enable the PIM feature.

This is a prerequisite only for multicast replication.

Before you begin

Ensure that you have installed the LAN Base Services license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature pim**	Enables PIM. By default, PIM is disabled.
Step 3	(Optional) switch(config)# **show running-config pim**	Shows the running-configuration information for PIM, including the **feature** command.

	Command or Action	Purpose
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable the PIM feature:

```
switch# configure terminal
switch(config)# feature pim
switch(config)# ip pim spt-threshold infinity group-list rp_name
switch(config)# show running-config pim

!Command: show running-config pim
!Time: Wed Mar 26 08:04:23 2014

version 6.0(2)U3(1)
feature pim

ip pim spt-threshold infinity group-list rp_name
```

Configuring a Rendezvous Point

You can configure a rendezvous point (RP) by configuring the RP address on every router that will participate in the PIM domain.

This is a prerequisite only for multicast replication.

Before you begin

Ensure that you have installed the LAN Base Services license and enabled PIM.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip pim rp-address** *rp-address* [**group-list** *ip-prefix* | **route-map** *policy-name*]	Configures a PIM RP address for a multicast group range. You can specify a route-map policy name that lists the group prefixes to use with the **match ip multicast** command. The default mode is ASM. The default group range is 224.0.0.0 through 239.255.255.255.
Step 3	(Optional) switch(config)# **show ip pim group-range** [*ip-prefix*] [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays PIM modes and group ranges.

	Command or Action	Purpose
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure an RP:

```
switch# configure terminal
switch(config)# ip pim rp-address 111.1.1.1 group-list 224.0.0.0/4
```

Enabling a VXLAN

Enabling VXLANs involves the following:

- Enabling the VXLAN feature
- Enabling VLAN to VN-Segment mapping

Before you begin

Ensure that you have installed the VXLAN Enterprise license.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] feature nv overlay**	Enables the VXLAN feature.
Step 3	switch (config)# **[no] feature vn-segment-vlan-based**	Configures the global mode for all VXLAN bridge domains. Enables VLAN to VN-Segment mapping. VLAN to VN-Segment mapping is always one-to-one.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable a VXLAN and configure VLAN to VN-Segment mapping:

```
switch# configure terminal
switch(config)# feature nv overlay
switch(config)# feature vn-segment-vlan-based
switch(config)# copy running-config startup-config
```

Mapping a VLAN to a VXLAN VNI

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** *vlan-id*	Specifies a VLAN.
Step 3	switch(config-vlan)# **vn-segment** *vnid*	Specifies the VXLAN virtual network identifier (VNID). The range of values for vnid is 1 to 16777214.

Example

This example shows how to map a VLAN to a VXLAN VNI:

```
switch# configure terminal
switch(config)# vlan 3100
switch(config-vlan)# vn-segment 5000
```

Configuring a Routing Protocol for NVE Unicast Addresses

Configuring a routing protocol for unicast addresses involves the following:

- Configuring a dedicated loopback interface for NVE reachability.
- Configuring the routing protocol network type.
- Specifying the routing protocol instance and area for an interface.
- Enabling PIM sparse mode in case of multicast replication.

Note Open shortest path first (OSPF) is used as the routing protocol in the examples.

This is a prerequisite for both multicast and ingress replication.

Guidelines for configuring a routing protocol for unicast addresses are as follows:

- For ingress replication, you can use a routing protocol that can resolve adjacency, such as BGP.
- When using unicast routing protocols in a vPC topology, explicitly configure a unique router ID for the vPC peers to avoid the VTEP loopback IP address (which is the same on the vPC peers) being used as the router ID.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **interface loopback** *instance*	Creates a dedicated loopback interface for the NVE interface. The instance range is from 0 to 1023.
Step 3	switch(config-if)# **ip address** *ip-address/length*	Configures an IP address for this interface.
Step 4	switch(config-if)# **ip ospf network** {**broadcast** | **point-to-point**}	Configures the OSPF network type to a type other than the default for an interface.
Step 5	switch(config-if)# **ip router ospf** *instance-tag* **area** *area-id*	Specifies the OSPF instance and area for an interface.
Step 6	switch(config-if)# **ip pim sparse-mode**	Enables PIM sparse mode on this interface. The default is disabled. Enable the PIM sparse mode in case of multicast replication.

Example

This example shows how to configure a routing protocol for NVE unicast addresses:

```
switch# configure terminal
switch(config)# interface loopback 10
switch(config-if)# ip address 222.2.2.1/32
switch(config-if)# ip ospf network point-to-point
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# ip pim sparse-mode
```

Creating a VXLAN Destination UDP Port

The UDP port configuration should be done before the NVE interface is enabled.

Note If the configuration must be changed while the NVE interface is enabled, ensure that you shut down the NVE interface, make the UDP configuration change, and then reenable the NVE interface.

Ensure that the UDP port configuration is done network-wide before the NVE interface is enabled on the network.

The VXLAN UDP source port is determined based on the VNID and source and destination IP addresses.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **vxlan udp port** *number*	Specifies the destination UDP port number for VXLAN encapsulated packets. The default destination UDP port number is 4789.

Example

This example shows how to create a VXLAN destination UDP port:

```
switch# configure terminal
switch(config)# vxlan udp port 4789
```

Creating and Configuring an NVE Interface

An NVE interface is the overlay interface that initiates and terminates VXLAN tunnels. You can create and configure an NVE (overlay) interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface nve** *instance*	Creates a VXLAN overlay interface that initiates and terminates VXLAN tunnels. **Note** Only one NVE interface is allowed on the switch.
Step 3	switch(config-if-nve)# **source-interface loopback** *instance*	Specifies a source interface. The source interface must be a loopback interface that is configured on the switch with a valid /32 IP address. This /32 IP address must be known by the transit routers in the transport network and the remote VTEPs.

Example

This example shows how to create and configure an NVE interface:

```
switch# configure terminal
switch(config)# interface nve 1
switch(config-if-nve)# source-interface loopback 10
```

Configuring Replication for a VNI

Replication for VXLAN network identifier (VNI) can be configured in one of two ways:

- Multicast replication

- Ingress replication

Configuring Multicast Replication

Before you begin

- Ensure that the NVE interface is created and configured.
- Ensure that the source interface is specified.

Procedure

	Command or Action	Purpose
Step 1	switch(config-if-nve)# **member vni** {*vnid* **mcast-group** *multicast-group-addr* | *vnid-range* **mcast-group** *start-addr* [*end-addr*]}	Maps VXLAN VNIs to the NVE interface and assigns a multicast group to the VNIs.

Example

This example shows how to map a VNI to an NVE interface and assign it to a multicast group:

```
switch(config-if-nve)# member vni 5000 mcast-group 225.1.1.1
```

Configuring Ingress Replication

Before you begin

- Ensure that the NVE interface is created and configured.
- Ensure that the source interface is specified.

Procedure

	Command or Action	Purpose
Step 1	switch(config-if-nve)# **member vni** *vnid*	Maps VXLAN VNIs to the NVE interface.
Step 2	switch(config-if-nve-vni)# **ingress-replication protocol static**	Enables static ingress replication for the VNI.
Step 3	switch(config-if-nve-vni)# **peer-ip** *ip-address*	Enables the peer IP. **Note** • A VNI can be associated only with a single IP address. • An IP address can be associated only with a single VNI.

Example

This example shows how to map a VNI to an NVE interface and create a unicast tunnel:

```
switch(config-if-nve)# member vni 5001
switch(config-if-nve-vni)# ingress-replication protocol static
switch(config-if-nve-vni)# peer-ip 111.1.1.1
```

Configuring Q-in-VNI

Using Q-in-VNI provides a way for you to segregate traffic by mapping to a specific port. In a multi-tenant environment, you can specify a port to a tenant and send/receive packets over the VXLAN overlay.

Notes about configuring Q-in-VNI:

- Q-in-VNI is supported only for the Cisco Nexus 3100-V and 3132C-Z platform switches.
- The dot1q mode is not supported for 40G ports.
- Q-in-Q to Q-in-VNI interworking is supported.
- Q-in-VNI only supports VXLAN bridging. It does not support VXLAN routing.
- Q-in-VNI does not support FEX.
- When configuring access ports and trunk ports:
  - You can have access ports, trunk ports and dot1q ports on different interfaces on the same switch.
  - You cannot have the same VLAN configured for both dot1q and trunk ports/access ports.

Before you begin

Configuring the Q-in-VNI feature requires:

- The base port mode must be a dot1q tunnel port with an access VLAN configured.
- VNI mapping is required for the access VLAN on the port.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *type port*	Enters interface configuration mode.
Step 3	**switchport mode dot1q-tunnel**	Creates a 802.1Q tunnel on the port.
Step 4	**switchport access vlan** *vlan-id*	Specifies the port assigned to a VLAN.
Step 5	**spanning-tree bpdufilter enable**	Enables BPDU Filtering for the specified spanning tree edge interface. By default, BPDU Filtering is disabled.

Example

The following example shows how to configure Q-in-VNI:

```
switch# config terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree bpdufilter enable
switch(config-if)#
```

Verifying the VXLAN Configuration

Use one of the following commands to verify the VXLAN configuration, to display the MAC addresses, and to clear the MAC addresses:

Command	Purpose
show nve interface nve *id*	Displays the configuration of an NVE interface.
show nve vni	Displays the VNI that is mapped to an NVE interface.
show nve peers	Displays peers of the NVE interface.
show interface nve *id* **counters**	Displays all the counters for an NVE interface.
show nve vxlan-params	Displays the VXLAN UDP port configured.
show mac address-table	Displays both VLAN and VXLAN MAC addresses.
clear mac address-table dynamic	Clears all MAC address entries in the MAC address table.

Example

This example shows how to display the configuration of an NVE interface:

```
switch# show nve interface nve 1
Interface: nve1, State: up, encapsulation: VXLAN
Source-interface: loopback10 (primary: 111.1.1.1, secondary: 0.0.0.0)
```

This example shows how to display the VNI that is mapped to an NVE interface for multicast replication:

```
switch# show nve vni
Interface          VNI        Multicast-group   VNI State
-----------------  --------   ---------------   ---------
nve1               5000       225.1.1.1          Up
```

This example shows how to display the VNI that is mapped to an NVE interface for ingress replication:

```
switch# show nve vni
Interface          VNI        Multicast-group   VNI State
-----------------  --------   ---------------   ---------
nve1               5000       0.0.0.0           Up
```

This example shows how to display the peers of an NVE interface:

```
switch# show nve peers
Interface          Peer-IP           Peer-State
-----------------  ---------------   -------------
nve1               111.1.1.1          Up
```

This example shows how to display the counters of an NVE interface:

```
switch# show interface nv 1 counter

--------------------------------------------------------------------------------
Port                                InOctets                    InUcastPkts
--------------------------------------------------------------------------------
nve1                                       0                              0

--------------------------------------------------------------------------------
Port                              InMcastPkts                    InBcastPkts
--------------------------------------------------------------------------------
nve1                                       0                              0

--------------------------------------------------------------------------------
Port                               OutOctets                   OutUcastPkts
--------------------------------------------------------------------------------
nve1                                       0                              0

--------------------------------------------------------------------------------
Port                             OutMcastPkts                   OutBcastPkts
--------------------------------------------------------------------------------
nve1                                       0                              0
```

This example shows how to display the VXLAN UDP port configured:

```
switch# show nve vxlan-params
VxLAN Dest. UDP Port: 4789
```

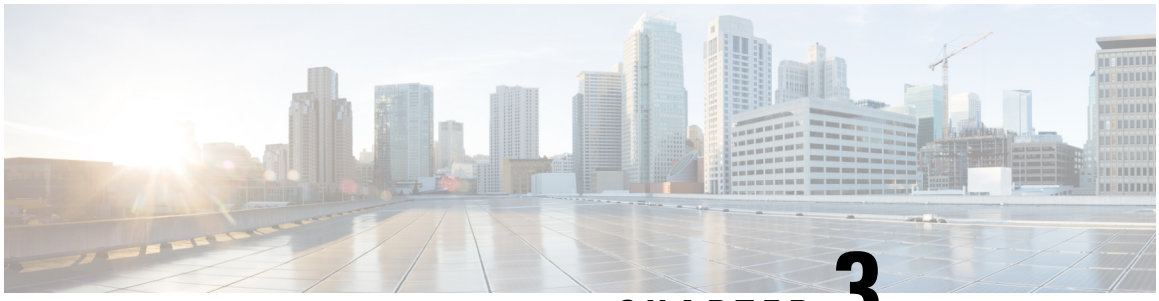
This example shows how to display both VLAN and VXLAN MAC addresses:

```
switch# show mac address-table
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since first seen,+ - primary entry using vPC Peer-Link
   VLAN     MAC Address      Type      age     Secure NTFY   Ports/SWID.SSID.LID
---------+-----------------+--------+---------+------+----+------------------
* 109      0000.0410.0902    dynamic   470        F    F  Po2233
* 109      0000.0410.0912    dynamic   470        F    F  Po2233
* 109      0000.0410.0912    dynamic   470        F    F  nve1(1.1.1.200)
* 108      0000.0410.0802    dynamic   470        F    F  Po2233
* 108      0000.0410.0812    dynamic   470        F    F  Po2233
* 107      0000.0410.0702    dynamic   470        F    F  Po2233
* 107      0000.0410.0712    dynamic   470        F    F  Po2233
* 107      0000.0410.0712    dynamic   470        F    F  nve1(1.1.1.200)
* 106      0000.0410.0602    dynamic   470        F    F  Po2233
* 106      0000.0410.0612    dynamic   470        F    F  Po2233
* 105      0000.0410.0502    dynamic   470        F    F  Po2233
* 105      0000.0410.0512    dynamic   470        F    F  Po2233
* 105      0000.0410.0512    dynamic   470        F    F  nve1(1.1.1.200)
```

```
* 104      0000.0410.0402    dynamic   470        F    F  Po2233
* 104      0000.0410.0412    dynamic   470        F    F  Po2233
```

This example shows how to clear all MAC address entries in the MAC address table:

```
switch# clear mac address-table dynamic
switch#
```



CHAPTER 3

IGMP Snooping Over VXLAN

Overview of IGMP Snooping Over VXLAN

The configuration of IGMP snooping is same in VXLAN as in configuration of IGMP snooping in regular VLAN domain. All the configuration CLIs remain the same. For more information on IGMP snooping, see the *Configuring IGMP Snooping* section in *Cisco Nexus 3000 Series NX-OS Multicast Routing Configuration Guide*.

Guidelines and Limitations for IGMP Snooping Over VXLAN

See the following guidelines and limitations for IGMP snooping over VXLAN:

- IGMP snooping over VXLAN is supported.
- IGMP snooping on VXLAN VLAN is disabled by default.
- For IGMP snooping over VXLAN, all the guidelines and limitations of VXLAN apply.
- IGMP snooping over VXLAN is not supported on any FEX enabled platforms and FEX ports.
- IGMP snooping over VXLAN VLAN is supported for Cisco Nexus 3100-V and 3172 platform switches in N9K mode only.

Configuring IGMP Snooping Over VXLAN

Before you begin

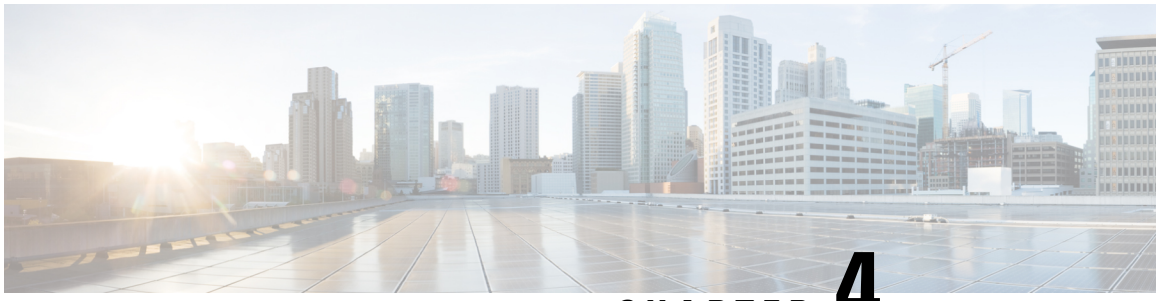
For VXLAN IGMP snooping functionality, the ARP-ETHER TCAM must be configured in the double-wide mode using the CLI command, switch# **hardware access-list tcam region arp-ether 256 double wide**.

Procedure

	Command or Action	Purpose
Step 1	switch(config)#**ip igmp snooping vxlan**	Enables IGMP snooping for VXLAN VLANs. You have to explicitly configure this command to enable snooping for VXLAN VLANs.
Step 2	switch(config)#**ip igmp snooping disable-nve-static-router-port**	Configures IGMP snooping over VXLAN to not include NVE as static mrouter port using this global CLI command. IGMP snooping over VXLAN has the NVE interface as mrouter port by default.

Step 3	switch(config)#**system nve ipmc global index-size ?**
	Example:
	`switch(config)# system nve ipmc global index-size ?`
	`<1000-7000> Ipmc allowed size`
		Configures the VXLAN global IPMC index size. IGMP snooping over VXLAN uses the IPMC indexes from the NVE global range on the Cisco Nexus 3000 Series switches with Network Forwarding Engine (NFE). You need to reconfigure the VXLAN global IPMC index size according to the scale using this command. Cisco recommends to reserve 6000 IPMC indexes using this CLI command. The default IPMC index size is 3000.

Step 4	switch(config)# **ip igmp snooping vxlan-umc drop vlan ?**
	Example:
	`switch(config)# ip igmp snooping vxlan-umc drop vlan ?`
	`<1-3863>  VLAN IDs for which unknown multicast traffic is dropped`
		Configures IGMP snooping over VXLAN to drop all the unknown multicast traffic on per VLAN basis using this global CLI command. On Cisco Nexus 3000 Series switches with Network Forwarding Engine (NFE), the default behavior of all unknown multicast traffic is to flood to the bridge domain.
		Note IPv6 neighbor solicitation packets are dropped when this command is enabled. Therefore, IPv6 hosts are not resolved.



CHAPTER 4

Configuring VXLAN BGP EVPN

This chapter contains the following sections:

Information About VXLAN BGP EVPN

Guidelines and Limitations for VXLAN BGP EVPN

VXLAN BGP EVPN has the following guidelines and limitations:

- Routing between VXLAN VLANs and non-VXLAN VLANs, and Layer 3 interfaces, is not supported on Cisco Nexus 3100-V platform switches. Hence, Cisco Nexus 3100-V platform switches cannot be a border leaf VTEP in a VXLAN EVPN setup.
- You can configure EVPN over segment routing or MPLS. See the Cisco Nexus 3000 Series NX-OS Label Switching Configuration Guide for more information.
- You can use MPLS tunnel encapsulation using the new CLI encapsulation **mpls** command. You can configure the label allocation mode for the EVPN address family. See the Cisco Nexus 3000 Series NX-OS Label Switching Configuration Guide for more information.
- In a VXLAN EVPN setup that has a 2K VNI scale configuration, the control plane down time takes more than 200 seconds. To avoid BGP flap, configure the graceful restart time to 300 seconds.
- SVI and sub-interfaces as core links are not supported in multisite EVPN.
- In a VXLAN EVPN setup, border leaves must use unique route distinguishers, preferably using **auto rd** command. It is not supported to have same route distinguishers in different border leaves.
- ARP suppression is only supported for a VNI if the VTEP hosts the First-Hop Gateway (Distributed Anycast Gateway) for this VNI. The VTEP and the SVI for this VLAN have to be properly configured for the distributed Anycast Gateway operation, for example, global Anycast Gateway MAC address configured and Anycast Gateway feature with the virtual IP address on the SVI.

- When Layer 3 EVPN is configured in Cisco Nexus 3000 platform switches that are based on Broadcom ASIC and these switches are added in the topology with Layer 2 EVPN, the routing for this scenario is not supported. When you configure SVI and Layer 3 EVPN on Cisco Nexus 3000 platform switches based on Broadcom ASIC with Anycast Gateway and when you send the ARP requests from a Layer 2 EVPN device (for example, Cisco Nexus 3000 platform switches, based on a Broadcom ASIC), the Cisco Nexus 3000 platform switches can not be used as a gateway for the ARP requests received on the network ports.
- The **show** commands with the **internal** keyword are not supported.
- DHCP snooping (Dynamic Host Configuration Protocol snooping) is not supported on VXLAN VLANs.
- SPAN TX for VXLAN encapsulated traffic is not supported for the Layer 3 uplink interface.
- RACLs are not supported on Layer 3 uplinks for VXLAN traffic. Egress VACLs support is not available for de-capsulated packets in the network to access direction on the inner payload.

  As a best practice, use PACLs/VACLs for the access to the network direction.
- QoS classification is not supported for VXLAN traffic in the network to access direction on the Layer 3 uplink interface.
- The QoS buffer-boost feature is not applicable for VXLAN traffic.
- VTEP does not support Layer 3 subinterface uplinks that carry VXLAN encapsulated traffic.
- Layer 3 interface uplinks that carry VXLAN encapsulated traffic do not support subinterfaces for non-VxLAN encapsulated traffic.
- Non-VXLAN sub-interface VLANs cannot be shared with VXLAN VLANs.
- Subinterfaces on 40G (ALE) uplink ports are not supported on VXLAN VTEPs.
- Point to multipoint Layer 3 and SVI uplinks are not supported. Since both uplink types can only be enabled point-to-point, they cannot span across more than two switches.
- For EBGP, it is recommended to use a single overlay EBGP EVPN session between loopbacks.
- Bind NVE to a loopback address that is separate from other loopback addresses that are required by Layer 3 protocols. A best practice is to use a dedicated loopback address for VXLAN.
- VXLAN BGP EVPN does not support an NVE interface in a non-default VRF.
- It is recommended to configure a single BGP session over the loopback for an overlay BGP session.
- The VXLAN UDP port number is used for VXLAN encapsulation. For Cisco Nexus NX-OS, the UDP port number is 4789. It complies with IETF standards and is not configurable.
- VXLAN supports In Service Software Upgrade (ISSU).
- VTEP connected to FEX host interface ports is not supported.
- Resilient hashing (port-channel load-balancing resiliency) and VXLAN configurations are not compatible with VTEPs using ALE uplink ports.

Note Resilient hashing is disabled by default.

Note For information about VXLAN BGP EVPN scalability, see the Verified Scalability Guide for your platform.

Notes for EVPN Convergence

The following are notes about EVPN Convergence (7.0(3)I3(1) and later):

- As a best practice, the NVE source loopback should be dedicated to NVE. so that NVE can bring the loopback up or down as needed.
- When vPC has been configured, the loopback stays down until the MCT link comes up.

  Note When **feature vpc** is enabled and there is no VPC configured, the NVE source loopback is in "shutdown" state after an upgrade. In this case, removing **feature vpc** restores the interface to "up" state."

- The NVE underlay (through the source loopback) is kept down until the overlay has converged.
  - When MCT comes up, the source loopback is kept down for an amount of time that is configurable. This approach prevents north-south traffic from coming in until the overlay has converged.
  - When MCT goes down, NVE is kept up for 30 seconds in the event that there is still south-north traffic from vPC legs which have not yet gone down.
- BGP ignores routes from vPC peer. This reduces the number of routes in BGP.

Considerations for VXLAN BGP EVPN Deployment

- A loopback address is required when using the **source-interface config** command. The loopback address represents the local VTEP IP.
- During boot-up of a switch (7.0(3)I2(2) and later), you can use the **source-interface hold-down-time** *hold-down-time* command to suppress advertisement of the NVE loopback address until the overlay has converged. The range for the *hold-down-time* is 0 - 2147483647 seconds. The default is 300 seconds.
- To establish IP multicast routing in the core, IP multicast configuration, PIM configuration, and RP configuration is required.
- VTEP to VTEP unicast reachability can be configured through any IGP/BGP protocol.
- If the anycast gateway feature is enabled for a specific VNI, then the anyway gateway feature must be enabled on all VTEPs that have that VNI configured. Having the anycast gateway feature configured on only some of the VTEPs enabled for a specific VNI is not supported.
- It is a requirement when changing the primary or secondary IP address of the NVE source interfaces to shut the NVE interface before changing the IP address.
- As a best practice, the RP for the multicast group should be configured only on the spine layer. Use the anycast RP for RP load balancing and redundancy.
- Every tenant VRF needs a VRF overlay VLAN and SVI for VXLAN routing.

- When configuring ARP suppression with BGP-EVPN, use the **hardware access-list tcam region arp-ether** *size* **double-wide** command to accommodate ARP in this region. (You must decrease the size of an existing TCAM region before using this command.)

VPC Considerations for VXLAN BGP EVPN Deployment

- The loopback address used by NVE needs to be configured to have a primary IP address and a secondary IP address.

  The secondary IP address is used for all VxLAN traffic that includes multicast and unicast encapsulated traffic.

- Each VPC peer needs to have separate BGP sessions to the spine.
- VPC peers must have identical configurations.
  - Consistent VLAN to VN-segment mapping.
  - Consistent NVE1 binding to the same loopback interface
    - Using the same secondary IP address.
    - Using different primary IP addresses.
  - Consistent VNI to group mapping.
  - The VRF overlay VLAN should be a member of the peer-link port-channel.
- For multicast, the VPC node that receives the (S, G) join from the RP (rendezvous point) becomes the DF (designated forwarder). On the DF node, encap routes are installed for multicast.

  Decap routes are installed based on the election of a decapper from between the VPC primary node and the VPC secondary node. The winner of the decap election is the node with the least cost to the RP. However, if the cost to the RP is the same for both nodes, the VPC primary node is elected.

  The winner of the decap election has the decap mroute installed. The other node does not have a decap route installed.

- On a VPC device, BUM traffic (broadcast, unknown-unicast, and multicast traffic) from hosts is replicated on the peer-link. A copy is made of every native packet and each native packet is sent across the peer-link to service orphan-ports connected to the peer VPC switch.

  To prevent traffic loops in VXLAN networks, native packets ingressing the peer-link cannot be sent to an uplink. However, if the peer switch is the encapper, the copied packet traverses the peer-link and is sent to the uplink.

  Note Each copied packet is sent on a special internal VLAN (VLAN 4041).

- When peer-link is shut, the loopback interface used by NVE on the VPC secondary is brought down and the status is **Admin Shut**. This is done so that the route to the loopback is withdrawn on the upstream and that the upstream can divert all traffic to the VPC primary.

Note Orphans connected to the VPC secondary will experience loss of traffic for the period that the peer-link is shut. This is similar to Layer 2 orphans in a VPC secondary of a traditional VPC setup.

- When peer-link is no-shut, the NVE loopback address is brought up again and the route is advertised upstream, attracting traffic.
- For VPC, the loopback interface has 2 IP addresses: the primary IP address and the secondary IP address.

  The primary IP address is unique and is used by Layer 3 protocols.

  The secondary IP address on loopback is necessary because the interface NVE uses it for the VTEP IP address. The secondary IP address must be same on both vPC peers.
- The VPC peer-gateway feature must be enabled on both peers.

  As a best practice, use peer-switch, peer gateway, ip arp sync, ipv6 nd sync configurations for improved convergence in VPC topologies.

  In addition, increase the STP hello timer to 4 seconds to avoid unnecessary TCN generations when VPC role changes occur.

  The following is an example (best practice) of a VPC configuration:

```
switch# sh ru vpc

version 6.1(2)I3(1)
feature vpc
vpc domain 2
  peer-switch
  peer-keepalive destination 172.29.206.65 source 172.29.206.64
  peer-gateway
  ipv6 nd synchronize
  ip arp synchronize
```

- On a VPC pair, shutting down NVE or NVE loopback on one of the VPC nodes is not a supported configuration. This means that traffic failover on one-side NVE shut or one-side loopback shut is not supported.
- Redundant anycast RPs configured in the network for multicast load-balancing and RP redundancy are supported on VPC VTEP topologies.
- Enabling vpc peer-gateway configuration is mandatory. For peer-gateway functionality, at least one backup routing SVI is required to be enabled across peer-link and also configured with PIM. This provides a backup routing path in the case when VTEP loses complete connectivity to the spine. Remote peer reachability is re-routed over the peer-link in this case.

  The following is an example of SVI with PIM enabled:

```
swithch# sh ru int vlan 2

interface Vlan2
  description special_svi_over_peer-link
  no shutdown
  ip address 30.2.1.1/30
```

```
ip pim sparse-mode
```

Note The SVI must be configured on both VPC peers and requires PIM to be enabled.

- As a best practice when changing the secondary IP address of an anycast VPC VTEP, the NVE interfaces on both the VPC primary and the VPC secondary should be shut before the IP changes are made.
- To provide redundancy and failover of VXLAN traffic when a VTEP loses all of its uplinks to the spine, it is recommended to run a Layer 3 link or an SVI link over the peer-link between VPC peers.
- If DHCP Relay is required in VRF for DHCP clients or if loopback in VRF is required for reachability test on a VPC pair, it is necessary to create a backup SVI per VRF with PIM enabled.

```
switchh# sh ru int vlan 20

interface Vlan20
description backup routing svi for VRF Green
vrf member GREEN
no shutdown
ip address 30.2.10.1/30
```

Network Considerations for VXLAN Deployments

- MTU Size in the Transport Network

  Due to the MAC-to-UDP encapsulation, VXLAN introduces 50-byte overhead to the original frames. Therefore, the maximum transmission unit (MTU) in the transport network needs to be increased by 50 bytes. If the overlays use a 1500-byte MTU, the transport network needs to be configured to accommodate 1550-byte packets at a minimum. Jumbo-frame support in the transport network is required if the overlay applications tend to use larger frame sizes than 1500 bytes.

- ECMP and LACP Hashing Algorithms in the Transport Network

  As described in a previous section, Cisco Nexus 3000 Series Switches introduce a level of entropy in the source UDP port for ECMP and LACP hashing in the transport network. As a way to augment this implementation, the transport network uses an ECMP or LACP hashing algorithm that takes the UDP source port as an input for hashing, which achieves the best load-sharing results for VXLAN encapsulated traffic.

- Multicast Group Scaling

  The VXLAN implementation on Cisco Nexus 3000 Series Switches uses multicast tunnels for broadcast, unknown unicast, and multicast traffic forwarding. Ideally, one VXLAN segment mapping to one IP multicast group is the way to provide the optimal multicast forwarding. It is possible, however, to have multiple VXLAN segments share a single IP multicast group in the core network. VXLAN can support up to 16 million logical Layer 2 segments, using the 24-bit VNID field in the header. With one-to-one mapping between VXLAN segments and IP multicast groups, an increase in the number of VXLAN segments causes a parallel increase in the required multicast address space and the amount of forwarding states on the core network devices. At some point, multicast scalability in the transport network can become a concern. In this case, mapping multiple VXLAN segments to a single multicast group can help conserve multicast control plane resources on the core devices and achieve the desired VXLAN scalability. However, this mapping comes at the cost of suboptimal multicast forwarding. Packets forwarded to the

multicast group for one tenant are now sent to the VTEPs of other tenants that are sharing the same multicast group. This causes inefficient utilization of multicast data plane resources. Therefore, this solution is a trade-off between control plane scalability and data plane efficiency.

Despite the suboptimal multicast replication and forwarding, having multiple-tenant VXLAN networks to share a multicast group does not bring any implications to the Layer 2 isolation between the tenant networks. After receiving an encapsulated packet from the multicast group, a VTEP checks and validates the VNID in the VXLAN header of the packet. The VTEP discards the packet if the VNID is unknown to it. Only when the VNID matches one of the VTEP's local VXLAN VNIDs, does it forward the packet to that VXLAN segment. Other tenant networks will not receive the packet. Thus, the segregation between VXLAN segments is not compromised.

Considerations for the Transport Network

The following are considerations for the configuration of the transport network:

- On the VTEP device:
  - Enable and configure IP multicast.*
  - Create and configure a loopback interface with a /32 IP address.

    (For vPC VTEPs, you must configure primary and secondary /32 IP addresses.)
  - Enable IP multicast on the loopback interface.*
  - Advertise the loopback interface /32 addresses through the routing protocol (static route) that runs in the transport network.
  - Enable IP multicast on the uplink outgoing physical interface.*
- Throughout the transport network:
  - Enable and configure IP multicast.*
- When using SVI uplinks with VXLAN enabled on Cisco Nexus 9200 and 9300-EX platform switches, use the **system nve infra-vlans** command to specify the VLANs that are used for uplink SVI. Failing to specify the VLANs results in traffic loss.

Note

- The **system nve infra-vlans** command specifies VLANs used by all SVI interfaces for uplink and vPC peer-links in VXLAN as infra-VLANs.
- You should not configure certain combinations of infra-VLANs. For example, 2 and 514, 10 and 522, which are 512 apart.

Note * Not required for static ingress replication or BGP EVPN ingress replication.

BGP EVPN Considerations for VXLAN Deployment

Commands for BGP EVPN

The following describes commands to support BGP EVPN VXLAN control planes.

Command	Description
member vni *range* [**associate-vrf**]	Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface The attribute **associate- vrf** is used to identify and separate processing VNIs that are associated with a VRF and used for routing. **Note** The VRF and VNI specified with this command must match the configuration of the VNI under the VRF.
show nve vni **show nve vni summary**	Displays information that determine if the VNI is configured for peer and host learning via the control plane or data plane.
show bgp l2vpn evpn **show bgp l2vpn evpn summary**	Displays the Layer 2 VPN EVPN address family.
host-reachability protocol bgp	Specifies BGP as the mechanism for host reachability advertisement.
suppress-arp	Suppresses ARP under Layer 2 VNI.
fabric forwarding anycast-gateway-mac	Configures anycast gateway MAC of the switch.
vrf context	Creates the VRF and enter the VRF mode.
nv overlay evpn	Enables/Disables the Ethernet VPN (EVPN).
router bgp	Configures the Border Gateway Protocol (BGP).

Command	Description
suppress mac-route	Suppresses the BGP MAC route so that BGP only sends the MAC/IP route for a host. Under NVE, the MAC updates for all VNIs are suppressed. **Note** • Receive-side — Suppressing the MAC route depends upon the capability of the remote EVPN peer to derive a MAC route from the MAC/IP route (7.0(3)I2(2) and later). Avoid using the "suppress mac-route" command if devices in the network are running an earlier NX-OS release. • Send-side — Suppressing the MAC route means that the sender has a MAC/IP route. If your configuration has pure-Layer 2 VNIs (such as no corresponding VRF or Layer3-VNI), then there is no corresponding MAC/IP and you should avoid using the "suppress mac-route" command.

Configuring VXLAN BGP EVPN

Enabling VXLAN

Enable VXLAN and the EVPN.

Procedure

	Command or Action	Purpose
Step 1	**feature vn-segment**	Enable VLAN-based VXLAN
Step 2	**feature nv overlay**	Enable VXLAN
Step 3	**nv overlay evpn**	Enable the EVPN control plane for VXLAN.

Configuring VLAN and VXLAN VNI

Procedure

	Command or Action	Purpose
Step 1	**vlan** *number*	Specify VLAN.
Step 2	**vn-segment** *number*	Map VLAN to VXLAN VNI to configure Layer 2 VNI under VXLAN VLAN.

Configuring VRF for VXLAN Routing

Configure the tenant VRF.

Procedure

	Command or Action	Purpose
Step 1	**vrf context** *vxlan*	Configure the VRF.
Step 2	**vni** *number*	Specify VNI.
Step 3	**rd auto**	Specify VRF RD (route distinguisher).
Step 4	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 5	**route-target both auto**	**Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP.
Step 6	**route-target both auto evpn**	**Note** Specifying the **auto** option is applicable only for IBGP. The **auto** option is available beginning with Cisco NX-OS Release 7.0(3)I7(1). Manually configured route targets are required for EBGP.
Step 7	**address-family ipv6 unicast**	Configure address family for IPv6.
Step 8	**route-target both auto**	**Note** Specifying the **auto** option is applicable only for IBGP. The **auto** option is available beginning with Cisco NX-OS Release 7.0(3)I7(1). Manually configured route targets are required for EBGP.

	Command or Action	Purpose
Step 9	**route-target both auto evpn**	**Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP.

Configuring SVI for Hosts for VXLAN Routing

Configure the SVI for hosts.

Procedure

	Command or Action	Purpose
Step 1	**vlan** *number*	Specify VLAN
Step 2	**interface** *vlan-number*	Specify VLAN interface.
Step 3	**vrf member** *vxlan-number*	Configure SVI for host.
Step 4	**ip address** *address*	Specify IP address.

Configuring VRF Overlay VLAN for VXLAN Routing

Procedure

	Command or Action	Purpose
Step 1	**vlan** *number*	Specify VLAN.
Step 2	**vn-segment** *number*	Specify vn-segment.

Configuring VNI Under VRF for VXLAN Routing

Configures a Layer 3 VNI under a VRF overlay VLAN. (A VRF overlay VLAN is a VLAN that is not associated with any server facing ports. All VXLAN VNIs that are mapped to a VRF, need to have their own internal VLANs allocated to it.)

Procedure

	Command or Action	Purpose
Step 1	**vrf context** *vxlan*	Create a VXLAN Tenant VRF
Step 2	**vni** *number*	Configure Layer 3 VNI under VRF.

Configuring Anycast Gateway for VXLAN Routing

Procedure

	Command or Action	Purpose
Step 1	**fabric forwarding anycast-gateway-mac** *address*	Configure distributed gateway virtual MAC address **Note** One virtual MAC per VTEP **Note** All VTEPs should have the same virtual MAC address
Step 2	**fabric forwarding mode anycast-gateway**	Associate SVI with anycast gateway under VLAN configuration mode.

Configuring the NVE Interface and VNIs

Procedure

	Command or Action	Purpose
Step 1	**interface** *nve-interface*	Configure the NVE interface.
Step 2	**host-reachability protocol bgp**	This defines BGP as the mechanism for host reachability advertisement
Step 3	**member vni** *vni* **associate-vrf**	Add Layer-3 VNIs, one per tenant VRF, to the overlay. **Note** Required for VXLAN routing only.
Step 4	**global mcast-group** *ip-address*	
Step 5	**member vni** *vni*	Add Layer 2 VNIs to the tunnel interface.
Step 6	**mcast-group** *address*	Configure the mcast group on a per-VNI basis

Configuring BGP on the VTEP

Procedure

	Command or Action	Purpose
Step 1	**router bgp** *number*	Configure BGP.
Step 2	**router-id** *address*	Specify router address.

	Command or Action	Purpose
Step 3	**neighbor** *address* **remote-as** *number*	Define MP-BGP neighbors. Under each neighbor define l2vpn evpn.
Step 4	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 5	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor. **Note** Address-family ipv4 evpn for vxlan host-based routing
Step 6	(Optional) **Allowas-in**	Allows duplicate AS numbers in the AS path. Configure this parameter on the leaf for eBGP when all leafs are using the same AS, but the spines have a different AS than leafs.
Step 7	**send-community extended**	Configures community for BGP neighbors.
Step 8	**vrf** *vrf-name*	Specify VRF.
Step 9	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 10	**advertise** *l2vpn* **evpn**	Enable advertising EVPN routes. **Note** Beginning with Cisco NX-OS Release 9.2(1), the **advertise l2vpn evpn** command no longer takes effect. To disable advertisement for a VRF toward the EVPN, disable the VNI in NVE by entering the **no member vni** *vni* **associate-vrf** command in interface nve1. The *vni* is the VNI associated with that particular VRF.
Step 11	**address-family ipv6 unicast**	Configure address family for IPv6.
Step 12	**advertise** *l2vpn* **evpn**	Enable advertising EVPN routes.

Configuring RD and Route Targets for VXLAN Bridging

Procedure

	Command or Action	Purpose
Step 1	**evpn**	Configure VRF.
Step 2	**vni** *number* **l2**	**Note** Only Layer 2 VNIs need to be specified.

	Command or Action	Purpose
Step 3	**rd auto**	Define VRF RD (route distinguisher) to configure VRF context.
Step 4	**route-target import auto**	Define VRF Route Target and import policies.
Step 5	**route-target export auto**	Define VRF Route Target and export policies.

Configuring BGP for EVPN on the Spine

Procedure

	Command or Action	Purpose
Step 1	**route-map permitall permit 10**	Configure route-map. **Note** The route-map keeps the next-hop unchanged for EVPN routes. • Required for eBGP. • Optional for iBGP.
Step 2	**set ip next-hop unchanged**	Set next-hop address. **Note** The route-map keeps the next-hop unchanged for EVPN routes. • Required for eBGP. • Optional for iBGP. **Note** When two next hops are enabled, next hop ordering is not maintained. If one of the next hops is a VXLAN next hop and the other next hop is local reachable via FIB/AM/Hmm, the local next hop reachable via FIB/AM/Hmm is always taken irrespective of the order. Directly/locally connected next hops are always given priority over remotely connected next hops.
Step 3	**router bgp** *autonomous system number*	Specify BGP.
Step 4	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 5	**retain route-target all**	Configure retain route-target all under address-family Layer 2 VPN EVPN [global].

	Command or Action	Purpose
		Note Required for eBGP. Allows the spine to retain and advertise all EVPN routes when there are no local VNI configured with matching import route targets.
Step 6	**neighbor** *address* **remote-as** *number*	Define neighbor.
Step 7	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 8	**disable-peer-as-check**	Disables checking the peer AS number during route advertisement. Configure this parameter on the spine for eBGP when all leafs are using the same AS but the spines have a different AS than leafs. **Note** Required for eBGP.
Step 9	**send-community extended**	Configures community for BGP neighbors.
Step 10	**route-map permitall out**	Applies route-map to keep the next-hop unchanged. **Note** Required for eBGP.

Suppressing ARP

Suppressing ARP includes changing the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

Procedure

	Command or Action	Purpose
Step 1	**hardware access-list tcam region arp-ether** *size* **double-wide**	Configure TCAM region to suppress ARP. *tcam-size*—TCAM size. The size has to be a multiple of 256. If the size is more than 256, it has to be a multiple of 512. **Note** Reload is required for the TCAM configuration to be in effect.
Step 2	**interface nve 1**	Create the network virtualization endpoint (NVE) interface.
Step 3	**member vni** *vni-id*	Specify VNI ID.
Step 4	**suppress-arp**	Configure to suppress ARP under Layer 2 VNI.

	Command or Action	Purpose
Step 5	**copy running-config start-up-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Disabling VXLANs

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**no nv overlay evpn**	Disables EVPN control plane.
Step 3	**no feature vn-segment-vlan-based**	Disables the global mode for all VXLAN bridge domains
Step 4	**no feature nv overlay**	Disables the VXLAN feature.
Step 5	(Optional) **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Duplicate Detection for IP and MAC Addresses

Cisco NX-OS supports duplicate detection for IP and MAC addresses. This enables the detection of duplicate IP or MAC addresses based on the number of moves in a given time-interval (seconds).

The default is 5 moves in 180 seconds. (Default number of moves is 5 moves. Default time-interval is 180 seconds.)

- For IP addresses:
  - After the 5th move within 180 seconds, the switch starts a 30 second lock (hold down timer) before checking to see if the duplication still exists (an effort to prevent an increment of the sequence bit). This 30 second lock can occur 5 times within 24 hours (this means 5 moves in 180 seconds for 5 times) before the switch permanently locks or freezes the duplicate entry. (**show fabric forwarding ip local-host-db vrf abc**)
- For MAC addresses:
  - After the 5th move within 180 seconds, the switch starts a 30 second lock (hold down timer) before checking to see if the duplication still exists (an effort to prevent an increment of the sequence bit). This 30 second lock can occur 3 times within 24 hours (this means 5 moves in 180 seconds for 3 times) before the switch permanently locks or freezes the duplicate entry. (**show l2rib internal permanently-frozen-list**)
- Wherever a MAC address is permanently frozen, a syslog message with written by L2RIB.

```
2017 Jul  5 10:27:34 leaf %$ VDC-1 %$  %USER-2-SYSTEM_MSG: Unfreeze limit (3) hit, MAC
```

```
 0000.0033.3333in topo: 200 is permanently frozen - l2rib
2017 Jul  5 10:27:34 leaf %$ VDC-1 %$  %USER-2-SYSTEM_MSG: Detected duplicate host
0000.0033.3333, topology 200, during Local update, with host located at remote VTEP
1.2.3.4, VNI 2 - l2rib
2017 Jul  5 10:27:34 leaf %$ VDC-1 %$  %USER-2-SYSTEM_MSG: Unfreeze limit (3) hit, MAC
 0000.0033.3334in topo: 200 is permanently frozen - l2rib
2017 Jul  5 10:27:34 leaf %$ VDC-1 %$  %USER-2-SYSTEM_MSG: Detected duplicate host
0000.0033.3334, topology 200, during Local update, with host l
```

The following are example commands to help the configuration of the number of VM moves in a specific time interval (seconds) for duplicate IP-detection:

Command	Description

`switch(config)# fabric forwarding ?`
`anycast-gateway-mac`
`dup-host-ip-addr-detection`
	Available sub-commands:
	• Anycast gateway MAC of the switch.
	• To detect duplicate host addresses in n seconds.

`switch(config)# fabric forwarding dup-host-ip-addr-detection ?` `<1-1000>`	The number of host moves allowed in n seconds. The range is 1 to 1000 moves; default is 5 moves.
`switch(config)# fabric forwarding dup-host-ip-addr-detection 100 ?` `<2-36000>`	The duplicate detection timeout in seconds for the number of host moves. The range is 2 to 36000 seconds; default is 180 seconds.
`switch(config)# fabric forwarding dup-host-ip-addr-detection 100 10`	Detects duplicate host addresses (limited to 100 moves) in a period of 10 seconds.

The following are example commands to help the configuration of the number of VM moves in a specific time interval (seconds) for duplicate MAC-detection:

Command	Description

`switch(config)# l2rib dup-host-mac-detection ?`
`<1-1000>`
`default`
	Available sub-commands for L2RIB:
	• The number of host moves allowed in n seconds. The range is 1 to 1000 moves.
	• Default setting (5 moves in 180 in seconds).

`switch(config)# l2rib dup-host-mac-detection 100 ?` `<2-36000>`	The duplicate detection timeout in seconds for the number of host moves. The range is 2 to 36000 seconds; default is 180 seconds.

Command	Description
`switch(config)# l2rib dup-host-mac-detection 100 10`	Detects duplicate host addresses (limited to 100 moves) in a period of 10 seconds.

Enabling Nuage Controller Interoperability

The following steps enable Nuage controller interoperability.

Procedure

	Command or Action	Purpose
Step 1	**nuage controller interop**	Global command to enable interoperability mode.
Step 2	**router bgp** *number*	Configure BGP.
Step 3	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 4	**advertise-system-mac**	Enable Nuage interoperability mode for BGP.
Step 5	**allow-vni-in-ethertag**	Enable Nuage interoperability mode for BGP.
Step 6	**route-map permitall permit 10**	Configure route-map to permit all.
Step 7	**router bgp** *number*	Configure BGP.
Step 8	**vrf** *vrf-name*	Specify tenant VRF.
Step 9	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 10	**advertise l2vpn evpn**	Enable advertising EVPN routes.
Step 11	**redistribute hmm route-map permitall**	Enables advertise host tenant routes as evpn type-5 routes for interoperability.

Example

The following is an example to enable Nuage controller interoperability:

```
/*** Enable interoperability mode at global level. ***/
switch(config)# nuage controller interop

/*** Configure BGP to enable interoperability mode. ***/
switch(config)# router bgp 1001
switch(config-router)# address-family l2vpn evpn
switch(config-router-af)# advertise-system-mac
switch(config-router-af)# allow-vni-in-ethertag

/*** Advertise host tenant routes as evpn type-5 routes for interoperability. ***/
switch(config)# route-map permitall permit 10
switch(config)# router bgp 1001
```

```
switch(config-router)# vrf vni-491830
switch(config-router-vrf)# address-family ipv4 unicast
switch(config-router-vrf-af)# advertise l2vpn evpn
switch(config-router-vrf-af)# redistribute hmm route-map permitall
```

Verifying the VXLAN BGP EVPN Configuration

To display the VXLAN BGP EVPN configuration information, enter one of the following commands:

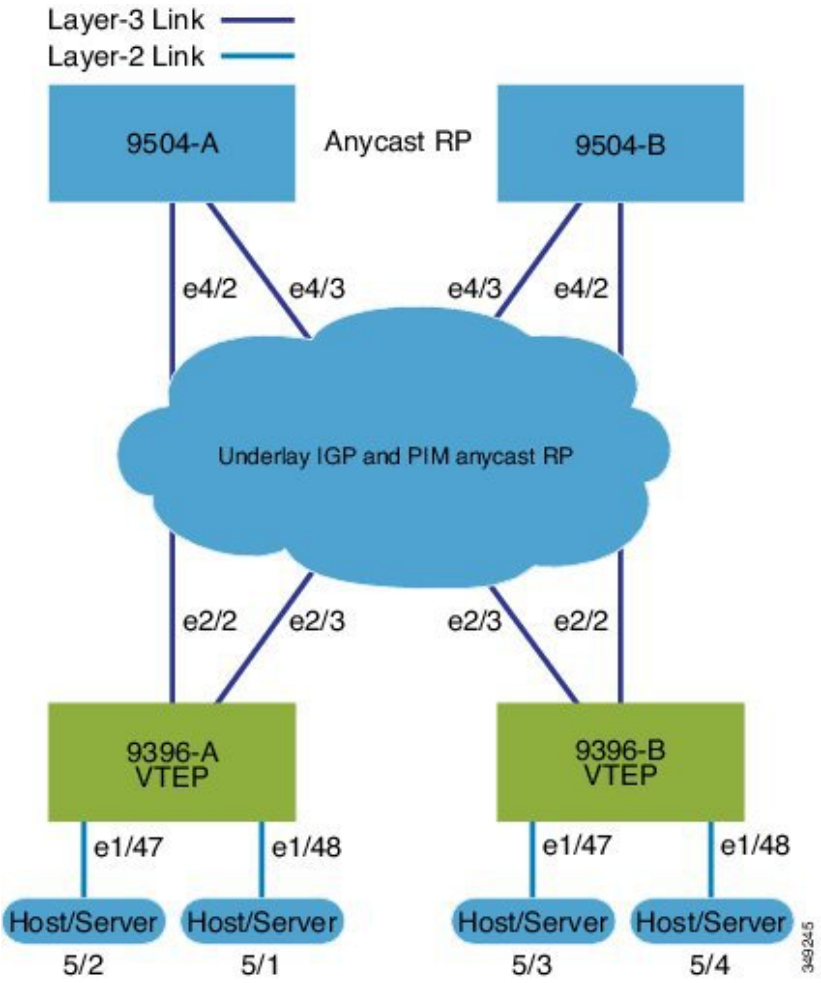
Command	Purpose
show nve vrf	Displays VRFs and associated VNIs
show bgp l2vpn evpn	Displays routing table information.
show ip arp suppression-cache [**detail** | **summary** | **vlan** *vlan* | **statistics**]	Displays ARP suppression information.
show vxlan interface	Displays VXLAN interface status.
show vxlan interface | count	Displays VXLAN VLAN logical port VP count. **Note** A VP is allocated on a per-port per-VLAN basis. The sum of all VPs across all VXLAN-enabled Layer 2 ports gives the total logical port VP count. For example, if there are 10 Layer 2 trunk interfaces, each with 10 VXLAN VLANs, then the total VXLAN VLAN logical port VP count is 10*10 = 100.
show l2route evpn mac [**all** | **evi** *evi* [**bgp** | **local** | **static** | **vxlan** | **arp**]]	Displays Layer 2 route information.
show l2route evpn fl all	Displays all fl routes.
show l2route evpn imet all	Displays all imet routes.
show l2route evpn mac-ip all **show l2route evpn mac-ip all detail**	Displays all MAC IP routes.
show l2route topology	Displays Layer 2 route topology.

Note Although the **show ip bgp** command is available for verifying a BGP configuration, as a best practice, it is preferable to use the **show bgp** command instead.

Example of VXLAN BGP EVPN (EBGP)

An example of a VXLAN BGP EVPN (EBGP):

Figure 3: VXLAN BGP EVPN Topology (EBGP)



EBGP between Spine and Leaf

- Spine (9504-A)
  - Enable the EVPN control plane

    ```
    nv overlay evpn
    ```

  - Enable the relevant protocols

    ```
    feature bgp
    feature pim
    ```

  - Configure Loopback for local VTEP IP, and BGP

```
interface loopback0
  ip address 10.1.1.1/32
  ip pim sparse-mode
```

- Configure Loopback for Anycast RP

```
interface loopback1
  ip address 100.1.1.1/32
  ip pim sparse-mode
```

- Configure Anycast RP

```
ip pim rp-address 100.1.1.1 group-list 225.0.0.0/8
ip pim rp-candidate loopback1 group-list 225.0.0.0/8
ip pim log-neighbor-changes
ip pim ssm range 232.0.0.0/8
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
```

- Configure route-map used by EBGP for Spine

```
route-map permitall permit 10
  set ip next-hop unchanged
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.1.42/24
  ip pim sparse-mode
  no shutdown

interface Ethernet4/3
  ip address 192.168.2.43/24
  ip pim sparse-mode
  no shutdown
```

- Configure the BGP overlay for the EVPN address family.

```
router bgp 100
  router-id 10.1.1.1
  address-family l2vpn evpn
    nexthop route-map permitall
    retain route-target all
  neighbor 30.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      disable-peer-as-check
      send-community extended
      route-map permitall out
  neighbor 40.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      disable-peer-as-check
      send-community extended
      route-map permitall out
```

- Configure the BGP underlay.

```
neighbor 192.168.1.43 remote-as 200
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
```

- Spine (9504-B)
  - Enable the EVPN control plane and the relevant protocols

```
feature telnet
feature nxapi
feature bash-shell
feature scp-server
nv overlay evpn
feature bgp
feature pim
feature lldp
```

  - Configure Anycast RP

```
ip pim rp-address 100.1.1.1 group-list 225.0.0.0/8
ip pim rp-candidate loopback1 group-list 225.0.0.0/8
ip pim log-neighbor-changes
ip pim ssm range 232.0.0.0/8
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
vlan 1-1002
route-map permitall permit 10
  set ip next-hop unchanged
```

  - Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.4.42/24
  ip pim sparse-mode
  no shutdown

interface Ethernet4/3
  ip address 192.168.3.43/24
  ip pim sparse-mode
  no shutdown
```

  - Configure Loopback for local VTEP IP, and BGP

```
interface loopback0
  ip address 20.1.1.1/32
  ip pim sparse-mode
```

  - Configure Loopback for Anycast RP

```
interface loopback1
  ip address 100.1.1.1/32
  ip pim sparse-mode
```

  - Configure the BGP overlay for the EVPN address family.

```
router bgp 100
```

```
  router-id 20.1.1.1
  address-family l2vpn evpn
    retain route-target all
  neighbor 30.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      disable-peer-as-check
      send-community extended
      route-map permitall out
  neighbor 40.1.1.1 remote-as 200
    ebgp-multihop 3
    address-family l2vpn evpn
      disable-peer-as-check
      send-community extended
      route-map permitall out
```

- Configure the BGP underlay.

```
neighbor 192.168.1.43 remote-as 200
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
```

- Leaf (9396-A)
  - Enable the EVPN control plane

```
nv overlay evpn
```

  - Enable the relevant protocols

```
feature bgp
feature pim
feature interface-vlan
feature dhcp
```

  - Configure DHCP relay for Tenant VRFs

```
service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay sub-option type cisco
ip dhcp relay information option vpn
```

  - Enable VXLAN with distributed anycast-gateway using BGP EVPN

```
feature vn-segment-vlan-based
feature nv overlay
fabric forwarding anycast-gateway-mac 0000.2222.3333
```

  - Enable PIM RP

```
ip pim rp-address 100.1.1.1 group-list 225.0.0.0/8
```

  - Configure Loopback for BGP

```
interface loopback0
```

```
  ip address 30.1.1.1/32
  ip pim sparse-mode
```

- Configure Loopback for local VTEP IP

```
interface loopback1
  ip address 50.1.1.1/32
  ip pim sparse-mode
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet2/2
  no switchport
  load-interval counter 1 5
  ip address 192.168.1.22/24
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  no switchport
  load-interval counter 1 5
  ip address 192.168.3.23/24
  ip pim sparse-mode
  no shutdown
```

- Create the VRF overlay VLAN and configure the vn-segment.

```
vlan 101
  vn-segment 900001
```

- Configure VRF overlay VLAN/SVI for the VRF

```
interface Vlan101
  no shutdown
  vrf member vxlan-900001
  ip forward
```

- Create VLAN and provide mapping to VXLAN

```
vlan 1001
  vn-segment 2001001
vlan 1002
  vn-segment 2001002
```

- Create VRF and configure VNI

```
vrf context vxlan-900001
  vni 900001
```

Note The **rd auto** and **route-target** commands are automatically configured unless one or more are entered as overrides.

```
  rd auto
  address-family ipv4 unicast
    route-target import 65535:101 evpn
```

```
    route-target export 65535:101 evpn
    route-target import 65535:101
    route-target export 65535:101
  address-family ipv6 unicast
    route-target import 65535:101 evpn
    route-target export 65535:101 evpn
    route-target import 65535:101
    route-target export 65535:101
```

- Create server facing SVI and enable distributed anycast-gateway

```
interface Vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 4.1.1.1/24
  ipv6 address 4:1:0:1::1/64
  fabric forwarding mode anycast-gateway
  ip dhcp relay address 192.168.100.1 use-vrf default

interface Vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24
  ipv6 address 4:2:0:1::1/64
  fabric forwarding mode anycast-gateway
```

- Configure ACL TCAM region for ARP suppression

```
hardware access-list tcam region arp-ether 256 double-wide
```

- 

Note You can choose either of the following two options for creating the NVE interface. Use the first option for a small number of VNIs. Use the second option to configure a large number of VNIs.

Create the network virtualization endpoint (NVE) interface

Option 1

```
interface nve1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  member vni 900001 associate-vrf
  member vni 2001001
    suppress-arp
    mcast-group 225.4.0.1
  member vni 2001002
    suppress-arp
    mcast-group 225.4.0.1
```

Option 2

```
interface nve1
  no shutdown
  source-interface loopback 1
  host-reachibility protocol bgp
  global suppress-arp
```

```
  global mcast-group 224.1.1.1 L3
  global mcast-group 255.1.1.1 L2
  member vni 10000 associate-vrf
  member vni 10001 associate-vrf
  member vni 10002 associate-vrf
  member vni 10003 associate-vrf
  member vni 10004 associate-vrf
  member vni 10005 associate-vrf
  member vni 20000
  member vni 20001
  member vni 20002
  member vni 20003
  member vni 20004
  member vni 20005
```

- Configure interfaces for hosts/servers.

```
interface Ethernet1/47
  switchport access vlan 1002
interface Ethernet1/48
  switchport access vlan 1001
```

- Configure BGP

```
router bgp 200
router-id 30.1.1.1
  neighbor 10.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
      allowas-in
      send-community extended
    address-family l2vpn evpn
      allowas-in
      send-community extended
  neighbor 20.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
      allowas-in
      send-community extended
    address-family l2vpn evpn
      allowas-in
      send-community extended
  vrf vxlan-900001

      advertise l2vpn evpn
```

Note The following commands in EVPN mode do not need to be entered.

```
evpn
  vni 2001001 l2
  vni 2001002 l2
```

Note The **rd auto** and **route-target auto** commands are automatically configured unless one or more are entered as overrides.

```
rd auto
    route-target import auto
    route-target export auto


router bgp 200
router-id 30.1.1.1
  neighbor 10.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
      allowas-in
      send-community extended
    address-family l2vpn evpn
      allowas-in
      send-community extended
  neighbor 20.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
      allowas-in
      send-community extended
    address-family l2vpn evpn
      allowas-in
      send-community extended
  vrf vxlan-900001

      advertise l2vpn evpn
```

Note The following **advertise** command is optional.

```
advertise l2vpn evpn


evpn
  vni 2001001 l2
  vni 2001002 l2
```

Note The following **rd auto** and **route-target** commands are automatically configured unless one or more are entered as overrides.

Note The following EVPN mode commands are optional.

```
             evpn
  vni 2001001 l2
```

```
    rd auto
    route-target import auto
    route-target export auto
  vni 2001002 l2
    rd auto
    route-target import auto
    route-target export auto
```

- Leaf (9396-B)
  - Enable the EVPN control plane functionality and the relevant protocols

```
feature telnet
feature nxapi
feature bash-shell
feature scp-server
nv overlay evpn
feature bgp
feature pim
feature interface-vlan
feature vn-segment-vlan-based
feature lldp
feature nv overlay
```

  - Enable VxLAN with distributed anycast-gateway using BGP EVPN

```
fabric forwarding anycast-gateway-mac 0000.2222.3333
```

  - Create the VRF overlay VLAN and configure the vn-segment

```
vlan 1-1002
vlan 101
  vn-segment 900001
```

  - Create VLAN and provide mapping to VXLAN

```
vlan 1001
  vn-segment 2001001
vlan 1002
  vn-segment 2001002
```

  - Create VRF and configure VNI

```
vrf context vxlan-900001
  vni 900001
```

Note The following commands are automatically configured unless one or more are entered as overrides.

```
rd auto
  address-family ipv4 unicast
    route-target import 65535:101 evpn
    route-target export 65535:101 evpn
    route-target import 65535:101
    route-target export 65535:101
```

```
  address-family ipv6 unicast
    route-target import 65535:101 evpn
    route-target export 65535:101 evpn
    route-target import 65535:101 evpn
    route-target export 65535:101 evpn
```

- Configure ACL TCAM region for ARP suppression

```
hardware access-list tcam region arp-ether 256 double-wide
```

- Configure internal control VLAN/SVI for the VRF

```
interface Vlan1

interface Vlan101
  no shutdown
  vrf member vxlan-900001
```

- Create server facing SVI and enable distributed anycast-gateway

```
interface Vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 4.1.1.1/24
  ipv6 address 4:1:0:1::1/64
  fabric forwarding mode anycast-gateway

interface Vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24
  ipv6 address 4:2:0:1::1/64
  fabric forwarding mode anycast-gateway
```

- Create the network virtualization endpoint (NVE) interface

Note You can choose either of the following two procedures for creating the NVE interface. Use Option 1 for a small number of VNIs. Use Option 2 to configure a large number of VNIs.

Option 1

```
interface nve1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  member vni 10000 associate-vrf
  mcast-group 224.1.1.1
  member vni 10001 associate-vrf
  mcast-group 224.1.1.1
  member vni20000
  suppress-arp
  mcast-group 225.1.1.1
  member vni 20001
  suppress-arp
  mcast-group 225.1.1.1
```

Option 2

```
interface nve1
  no shutdown
  source-interface loopback 1
  host-reachibility protocol bgp
  global suppress-arp
  global mcast-group 224.1.1.1 L3
  global mcast-group 255.1.1.1 L2
  member vni 10000 associate-vrf
  member vni 10001 associate-vrf
  member vni 10002 associate-vrf
  member vni 10003 associate-vrf
  member vni 10004 associate-vrf
  member vni 10005 associate-vrf
  member vni 20000
  member vni 20001
  member vni 20002
  member vni 20003
  member vni 20004
  member vni 20005
```

- Configure interfaces for hosts/servers

```
interface Ethernet1/47
  switchport access vlan 1002

interface Ethernet1/48
  switchport access vlan 1001
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet2/1

interface Ethernet2/2
  no switchport
  load-interval counter 1 5
  ip address 192.168.4.22/24
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  no switchport
  load-interval counter 1 5
  ip address 192.168.2.23/24
  ip pim sparse-mode
  no shutdown
```

- Configure Loopback for BGP

```
interface loopback0
  ip address 40.1.1.1/32
  ip pim sparse-mode
```

- Configure Loopback for local VTEP IP

```
interface loopback1
  ip address 51.1.1.1/32
  ip pim sparse-mode
```

• Configure BGP

```
router bgp 200
router-id 40.1.1.1
  neighbor 10.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
      allowas-in
      send-community extended
    address-family l2vpn
      allowas-in
      send-community extended
  neighbor 20.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
      allowas-in
      send-community extended
    address-family l2vpn
      allowas-in
      send-community extended
  vrf vxlan-900001
```

Note The following **advertise** command is optional.

```
advertise l2vpn evpn
```

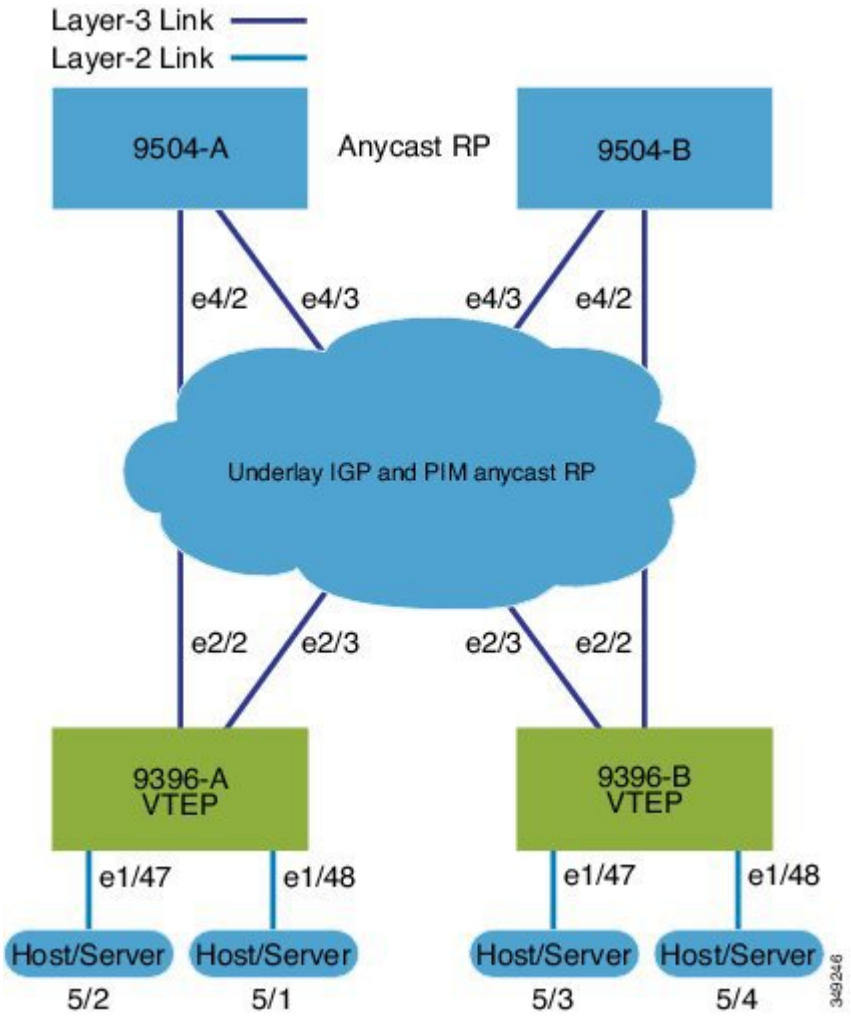
Note The **rd auto** and **route-target** commands are optional unless you want to use them to override the **import** or **export** options.

```
            evpn
  vni 2001001 l2
    rd auto
    route-target import auto
    route-target export auto
  vni 2001002 l2
    rd auto
    route-target import auto
    route-target export auto
```

Example of VXLAN BGP EVPN (IBGP)

An example of a VXLAN BGP EVPN (IBGP):

Figure 4: VXLAN BGP EVPN Topology (IBGP)



IBGP between Spine and Leaf

- Spine (9504-A)
  - Enable the EVPN control plane

    ```
    nv overlay evpn
    ```

  - Enable the relevant protocols

    ```
    feature ospf
    feature bgp
    feature pim
    ```

  - Configure Loopback for local VTEP IP, and BGP

    ```
    interface loopback0
      ip address 10.1.1.1/32
      ip router ospf 1 area 0.0.0.0
      ip pim sparse-mode
    ```

- Configure Loopback for Anycast RP

```
interface loopback1
  ip address 100.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

- Configure Anycast RP

```
ip pim rp-address 100.1.1.1 group-list 225.0.0.0/8
ip pim rp-candidate loopback1 group-list 225.0.0.0/8
ip pim ssm range 232.0.0.0/8
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
```

- Enable OSPF for underlay routing

```
router ospf 1
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.1.42/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet4/3
  ip address 192.168.2.43/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown
```

- Configure BGP

```
router bgp 65535
router-id 10.1.1.1
  neighbor 30.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-reflector-client
  neighbor 40.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-reflector-client
```

- Spine (9504-B)
  - Enable the EVPN control plane and the relevant protocols

```
feature telnet
feature nxapi
feature bash-shell
feature scp-server
nv overlay evpn
feature ospf
feature bgp
```

```
feature pim
feature lldp
```

- Configure Anycast RP

```
ip pim rp-address 100.1.1.1 group-list 225.0.0.0/8
ip pim rp-candidate loopback1 group-list 225.0.0.0/8
ip pim ssm range 232.0.0.0/8
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
vlan 1-1002
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.4.42/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet4/3
  ip address 192.168.3.43/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown
```

- Configure Loopback for local VTEP IP, and BGP

```
interface loopback0
  ip address 20.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

- Configure Loopback for Anycast RP

```
interface loopback1
  ip address 100.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

- Enable OSPF for underlay routing

```
router ospf 1
```

- Configure BGP

```
router bgp 65535
router-id 20.1.1.1
  neighbor 30.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-reflector-client
  neighbor 40.1.1.1 remote-as 65535
    update-source loopback0
     address-family l2vpn evpn
      send-community both
      route-reflector-client
```

- Leaf (9396-A)
  - Enable the EVPN control plane

```
nv overlay evpn
```

  - Enable the relevant protocols

```
feature ospf
feature bgp
feature pim
feature interface-vlan
```

  - Enable VxLAN with distributed anycast-gateway using BGP EVPN

```
feature vn-segment-vlan-based
feature nv overlay
fabric forwarding anycast-gateway-mac 0000.2222.3333
```

  - Enabling OSPF for underlay routing

```
router ospf 1
```

  - Configure Loopback for local VTEP IP, and BGP

```
interface loopback0
  ip address 30.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

  - Configure interfaces for Spine-leaf interconnect

```
interface Ethernet2/2
  no switchport
  ip address 192.168.1.22/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  no switchport
  ip address 192.168.3.23/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown
```

  - Configure PIM RP

```
ip pim rp-address 100.1.1.1 group-list 225.0.0.0/8
ip pim ssm range 232.0.0.0/8
```

  - Create overlay VRF VLAN and configure vn-segment

```
vlan 101
  vn-segment 900001
```

  - Configure VRF overlay VLAN/SVI for the VRF

```
interface Vlan101
  no shutdown
  vrf member vxlan-900001
```

- Create VLAN and provide mapping to VXLAN

```
vlan 1001
  vn-segment 2001001
vlan 1002
  vn-segment 2001002
```

- Create VRF and configure VNI

```
vrf context vxlan-900001
  vni 900001
```

Note The **rd auto** and **route-target** commands are automatically configured unless one or more are entered as overrides.

```
rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
  address-family ipv6 unicast
    route-target both auto
    route-target both auto evpn
```

- Create server facing SVI and enable distributed anycast-gateway

```
interface Vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 4.1.1.1/24
  ipv6 address 4:1:0:1::1/64
  fabric forwarding mode anycast-gateway

interface Vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24
  ipv6 address 4:2:0:1::1/64
  fabric forwarding mode anycast-gateway
```

- Configure ACL TCAM region for ARP suppression

```
hardware access-list tcam region arp-ether 256 double-wide
```

- 

Note You can choose either of the following two procedures for creating the NVE interfaces. Use the first one for a small number of VNIs. Use the second procedure to configure a large number of VNIs.

Create the network virtualization endpoint (NVE) interface

Option 1

```
interface nve1
  no shutdown
  source-interface loopback0
  host-reachability protocol bgp
  member vni 900001 associate-vrf
  member vni 2001001
    suppress-arp
    mcast-group 225.4.0.1
  member vni 2001002
    suppress-arp
    mcast-group 225.4.0.1
```

Option 2

```
Interface nve1
  source-interface loopback 1
  host-reachability protocol bgp
  global suppress-arp
  global mcast-group 255.1.1.1 L2
  global mcast-group 255.1.1.2 L3
  member vni 10000
  member vni 20000
  member vni 30000
```

- Configure interfaces for hosts/servers

```
interface Ethernet1/47
  switchport access vlan 1002

interface Ethernet1/48
  switchport access vlan 1001
```

- Configure BGP

```
router bgp 65535
router-id 30.1.1.1
  neighbor 10.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
  neighbor 20.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
  vrf vxlan-900001
    address-family ipv4 unicast
      advertise l2vpn evpn
```

Note The following commands in EVPN mode do not need to be entered.

```
evpn
  vni 2001001 l2
  vni 2001002 l2
```

Note The **rd auto** and **route-target auto** commands are automatically configured unless one or more are entered as overrides.

```
rd auto
    route-target import auto
    route-target export auto
```

Note The **rd auto** and **route-target** commands are automatically configured unless you want to use them to override the **import** or **export** options.

Note The following EVPN mode commands are optional.

```
            evpn
  vni 2001001 l2
    rd auto
    route-target import auto
    route-target export auto
  vni 2001002 l2
    rd auto
    route-target import auto
    route-target export auto
```

- Leaf (9396-B)
  - Enable the EVPN control plane functionality and the relevant protocols

```
feature telnet
feature nxapi
feature bash-shell
feature scp-server
nv overlay evpn
feature ospf
feature bgp
feature pim
feature interface-vlan
feature vn-segment-vlan-based
feature lldp
feature nv overlay
```

  - Enable VxLAN with distributed anycast-gateway using BGP EVPN

```
fabric forwarding anycast-gateway-mac 0000.2222.3333
```

  - Configure PIM RP

```
ip pim rp-address 100.1.1.1 group-list 225.0.0.0/8
ip pim ssm range 232.0.0.0/8
```

- Create overlay VRF VLAN and configure vn-segment

```
vlan 1-1002
vlan 101
  vn-segment 900001
```

- Create VLAN and provide mapping to VXLAN

```
vlan 1001
  vn-segment 2001001
vlan 1002
  vn-segment 2001002
```

- Create VRF and configure VNI

```
vrf context vxlan-900001
  vni 900001
```

Note The **rd auto** and **route-target** commands are automatically configured unless you want to use them to override the **import** or **export** options.

```
rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
  address-family ipv6 unicast
    route-target both auto
    route-target both auto evpn
```

- Configure ACL TCAM region for ARP suppression

```
hardware access-list tcam region arp-ether 256 double-wide
```

- Configure internal control VLAN/SVI for the VRF

```
interface Vlan101
  no shutdown
  vrf member vxlan-900001
```

- Create server facing SVI and enable distributed anycast-gateway

```
interface Vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 4.1.1.1/24
  ipv6 address 4:1:0:1::1/64
  fabric forwarding mode anycast-gateway

interface Vlan1002
  no shutdown
  vrf member vxlan-900001
```

```
  ip address 4.2.2.1/24
  ipv6 address 4:2:0:1::1/64
  fabric forwarding mode anycast-gateway
```

- 

Note You can choose either of the following two command procedures for creating the NVE interfaces. Use Option 1 for a small number of VNIs. Use Option 2 to configure a large number of VNIs.

Create the network virtualization endpoint (NVE) interface

Option 1

```
interface nve1
  no shutdown
  source-interface loopback0
  host-reachability protocol bgp
  member vni 900001 associate-vrf
  member vni 2001001
    suppress-arp
    mcast-group 225.4.0.1
  member vni 2001002
    suppress-arp
    mcast-group 225.4.0.1
```

Option 2

```
Interface nve1
  source-interface loopback0
  host-reachability protocol bgp
  global suppress-arp
  global mcast-group 255.4.0.1
  member vni 900001
  member vni 2001001
```

- Configure interfaces for hosts/servers

```
interface Ethernet1/47
  switchport access vlan 1002

interface Ethernet1/48
  switchport access vlan 1001
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet2/1

interface Ethernet2/2
  no switchport
  ip address 192.168.4.22/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  no switchport
  ip address 192.168.2.23/24
  ip router ospf 1 area 0.0.0.0
```

```
  ip pim sparse-mode
  no shutdown
```

- Configure Loopback for local VTEP IP, and BGP

```
interface loopback0
  ip address 40.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

- Enabling OSPF for underlay routing

```
router ospf 1
```

- Configure BGP

```
router bgp 65535
router-id 40.1.1.1
  neighbor 10.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
  neighbor 20.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
  vrf vxlan-900001
    address-family ipv4 unicast
      advertise l2vpn evpn
evpn
  vni 2001001 l2
    rd auto
    route-target import auto
    route-target export auto
  vni 2001002 l2
    rd auto
    route-target import auto
    route-target export auto
```

Note The **rd auto** and **route-target** commands are optional unless you want to use them to override the **import** or **export** options.

```
evpn
  vni 2001001 l2
    rd auto
    route-target import auto
    route-target export auto
  vni 2001002 l2
    rd auto
    route-target import auto
    route-target export auto
```

Example Show Commands

- **show nve peers**

```
9396-B# show nve peers
Interface Peer-IP          Peer-State
--------- ---------------  ----------
nve1      30.1.1.1         Up
```

- **show nve vni**

```
9396-B# show nve vni
Codes: CP - Control Plane        DP - Data Plane
       UC - Unconfigured         SA - Suppress ARP

Interface VNI      Multicast-group   State Mode Type [BD/VRF]      Flags
--------- -------- ----------------- ----- ---- ------------------ -----
nve1      900001   n/a               Up    CP   L3 [vxlan-900001]
nve1      2001001  225.4.0.1         Up    CP   L2 [1001]          SA
nve1      2001002  225.4.0.1         Up    CP   L2 [1002]          SA
```

- **show ip arp suppression-cache detail**

```
9396-B# show ip arp suppression-cache detail

Flags: + - Adjacencies synced via CFSoE
       L - Local Adjacency
       R - Remote Adjacency
       L2 - Learnt over L2 interface

Ip Address      Age      Mac Address    Vlan Physical-ifindex    Flags

4.1.1.54        00:06:41 0054.0000.0000 1001 Ethernet1/48        L
4.1.1.51        00:20:33 0051.0000.0000 1001 (null)              R
4.2.2.53        00:06:41 0053.0000.0000 1002 Ethernet1/47        L
4.2.2.52        00:20:33 0052.0000.0000 1002 (null)              R
```

- **show vxlan interface**

```
9396-B# show vxlan interface
Interface       Vlan    VPL Ifindex     LTL             HW VP
=========       ====    ===========     ===             =====
Eth1/47         1002    0x4c07d22e      0x10000         5697
Eth1/48         1001    0x4c07d02f      0x10001         5698
```

- **show bgp l2vpn evpn summary**

```
9396-B# show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 40.1.1.1, local AS number 65535
BGP table version is 27, L2VPN EVPN config peers 2, capable peers 2
14 network entries and 18 paths using 2984 bytes of memory
BGP attribute entries [14/2240], BGP AS path entries [0/0]
```

```
BGP community entries [0/0], BGP clusterlist entries [2/8]

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.1.1.1        4 65535   30199   30194       27    0    0     2w6d 4
20.1.1.1        4 65535   30199   30194       27    0    0     2w6d 4
```

- **show bgp l2vpn evpn**

```
9396-B# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 27, Local Router ID is 40.1.1.1
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-i
njected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 30.1.1.1:33768
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[0]:[0.0.0.0]/216
                      30.1.1.1                           100          0 i
* i                   30.1.1.1                           100          0 i
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[32]:[4.1.1.12]/272
                      30.1.1.1                           100          0 i
* i                   30.1.1.1                           100          0 i

Route Distinguisher: 30.1.1.1:33769
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[0]:[0.0.0.0]/216
                      30.1.1.1                           100          0 i
* i                   30.1.1.1                           100          0 i
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[32]:[4.2.2.11]/272
                      30.1.1.1                           100          0 i
* i                   30.1.1.1                           100          0 i

Route Distinguisher: 40.1.1.1:33768    (L2VNI 2001001)
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[0]:[0.0.0.0]/216
                      30.1.1.1                           100          0 i
*>l[2]:[0]:[0]:[48]:[f8c2.8890.2a45]:[0]:[0.0.0.0]/216
                      40.1.1.1                           100      32768 i
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[32]:[4.1.1.12]/272
                      30.1.1.1                           100          0 i
*>l[2]:[0]:[0]:[48]:[f8c2.8890.2a45]:[32]:[4.1.1.122]/272
                      40.1.1.1                           100      32768 i

Route Distinguisher: 40.1.1.1:33769    (L2VNI 2001002)
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[0]:[0.0.0.0]/216
                      30.1.1.1                           100          0 i
*>l[2]:[0]:[0]:[48]:[f8c2.8890.2a45]:[0]:[0.0.0.0]/216
                      40.1.1.1                           100      32768 i
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[32]:[4.2.2.11]/272
                      30.1.1.1                           100          0 i
*>l[2]:[0]:[0]:[48]:[f8c2.8890.2a45]:[32]:[4.2.2.111]/272
                      40.1.1.1                           100      32768 i

Route Distinguisher: 40.1.1.1:3    (L3VNI 900001)
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[32]:[4.1.1.12]/272
                      30.1.1.1                           100          0 i
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[32]:[4.2.2.11]/272
                      30.1.1.1                           100          0 i
```

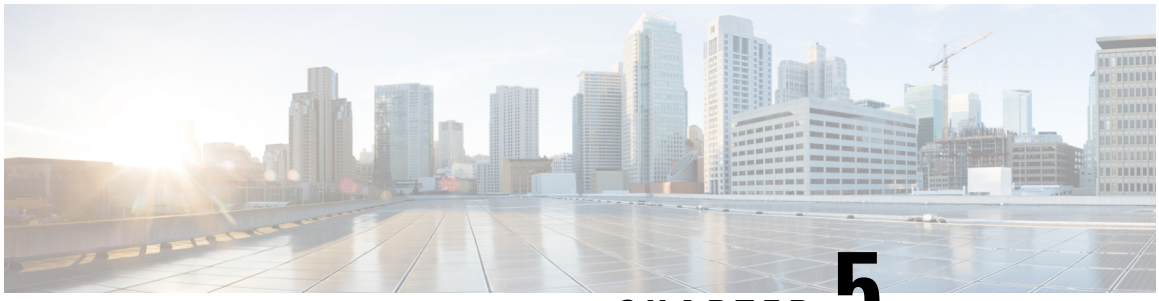
- **show l2route evpn mac all**

```
9396-B# show l2route evpn mac all
```

```
Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
(Dup):Duplicate (Spl):Split (Rcv):Recv (AD):Auto-Delete (D):Del Pending
(S):Stale (C):Clear, (Ps):Peer Sync (O):Re-Originated (Nho):NH-Override
(Pf):Permanently-Frozen

Topology    Mac Address    Prod   Flags         Seq No     Next-Hops
----------- -------------- ------ ------------- ---------- ----------------
101         6412.2574.9f27 VXLAN  Rmac          0          30.1.1.1
1001        d8b1.9071.e903 BGP    SplRcv        0          30.1.1.1
1001        f8c2.8890.2a45 Local  L,            0          Eth1/48
1002        d8b1.9071.e903 BGP    SplRcv        0          30.1.1.1
1002        f8c2.8890.2a45 Local  L,            0          Eth1/47
```

- **show l2route evpn mac-ip all**

```
9396-B# show l2route evpn mac-ip all
Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
(Dup):Duplicate (Spl):Split (Rcv):Recv(D):Del Pending (S):Stale (C):Clear
(Ps):Peer Sync (Ro):Re-Originated
Topology    Mac Address    Prod   Flags         Seq No     Host IP         Next-Hops
----------- -------------- ------ ---------- --------------- ---------------
1001        d8b1.9071.e903 BGP    --            0          4.1.1.12       30.1.1.1
1001        f8c2.8890.2a45 HMM    --            0          4.1.1.122      Local
1002        d8b1.9071.e903 BGP    --            0          4.2.2.11       30.1.1.1
1002        f8c2.8890.2a45 HMM    --            0          4.2.2.111      Local
```



CHAPTER 5

Configuring VXLAN OAM

This chapter contains the following sections:

VXLAN OAM Overview

The VXLAN operations, administration, and maintenance (OAM) protocol is a protocol for installing, monitoring, and troubleshooting Ethernet networks to enhance management in VXLAN based overlay networks.

Cisco Nexus 3500 Series switches do not support VXLAN OAM on Cisco NX-OS Release 7.0(3)I7(2) and the previous releases.

Similar to ping, traceroute, or pathtrace utilities that allow quick determination of the problems in the IP networks, equivalent troubleshooting tools have been introduced to diagnose the problems in the VXLAN networks. The VXLAN OAM tools, for example, ping, pathtrace, and traceroute provide the reachability information to the hosts and the VTEPs in a VXLAN network. The OAM channel is used to identify the type of the VXLAN payload that is present in these OAM packets.

There are two types of payloads supported:

- Conventional ICMP packet to the destination to be tracked
- Special NVO3 draft Tissa OAM header that carries useful information

The ICMP channel helps to reach the traditional hosts or switches that do not support the new OAM packet formats. The NVO3 draft Tissa channels helps to reach the supported hosts or switches and carries the important diagnostic information. The VXLAN NVO3 draft Tissa OAM messages may be identified via the reserved OAM EtherType or by using a well-known reserved source MAC address in the OAM packets depending on the implementation on different platforms. This constitutes a signature for recognition of the VXLAN OAM packets. The VXLAN OAM tools are categorized as shown in table below.

Table 1: VXLAN OAM Tools

Category	Tools
Fault Verification	Loopback Message
Fault Isolation	Path Trace Message
Performance	Delay Measurement, Loss Measurement
Auxiliary	Address Binding Verification, IP End Station Locator, Error Notification, OAM Command Messages, and Diagnostic Payload Discovery for ECMP Coverage

Loopback (Ping) Message

The loopback message (The ping and the loopback messages are the same and they are used interchangeably in this guide) is used for the fault verification. The loopback message utility is used to detect various errors and the path failures. Consider the topology in the following example where there are three core (spine) switches labeled Spine 1, Spine 2, and Spine 3 and five leaf switches connected in a Clos topology. The path of an example loopback message initiated from Leaf 1 for Leaf 5 is displayed when it traverses via Spine 3. When the loopback message initiated by Leaf 1 reaches Spine 3, it forwards it as VXLAN encapsulated data packet based on the outer header. The packet is not sent to the software on Spine 3. On Leaf 3, based on the appropriate loopback message signature, the packet is sent to the software VXLAN OAM module, that in turn, generates a loopback response that is sent back to the originator Leaf 1.

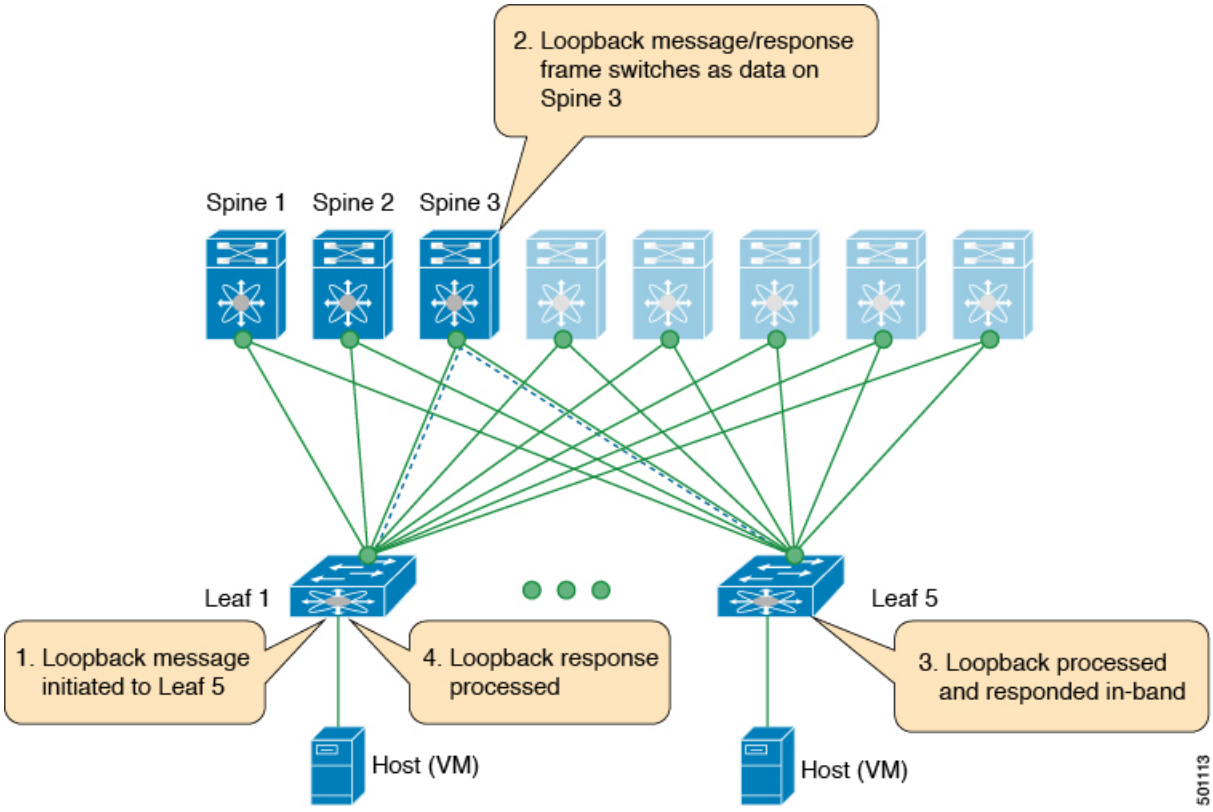
The loopback (ping) message can be destined to VM or to the (VTEP on) leaf switch. This ping message can use different OAM channels. If the ICMP channel is used, the loopback message can reach all the way to the VM if the VM's IP address is specified. If NVO3 draft Tissa channel is used, this loopback message is terminated on the leaf switch that is attached to the VM, as the VMs do not support the NVO3 draft Tissa headers in general. In that case, the leaf switch replies back to this message indicating the reachability of the VM. The ping message supports the following reachability options:

Ping

Check the network reachability (**Ping** command):

- From Leaf 1 (VTEP 1) to Leaf 2 (VTEP 2) (ICMP or NVO3 draft Tissa channel)
- From Leaf 1 (VTEP 1) to VM 2 (host attached to another VTEP) (ICMP or NVO3 draft Tissa channel)

Figure 5: Loopback Message



Traceroute or Pathtrace Message

The traceroute or pathtrace message is used for the fault isolation. In a VXLAN network, it may be desirable to find the list of switches that are traversed by a frame to reach the destination. When the loopback test from a source switch to a destination switch fails, the next step is to find out the offending switch in the path. The operation of the path trace message begins with the source switch transmitting a VXLAN OAM frame with a TTL value of 1. The next hop switch receives this frame, decrements the TTL, and on finding that the TTL is 0, it transmits a TTL expiry message to the sender switch. The sender switch records this message as an indication of success from the first hop switch. Then the source switch increases the TTL value by one in the next path trace message to find the second hop. At each new transmission, the sequence number in the message is incremented. Each intermediate switch along the path decrements the TTL value by 1 as is the case with regular VXLAN forwarding.

This process continues until a response is received from the destination switch, or the path trace process timeout occurs, or the hop count reaches a maximum configured value. The payload in the VXLAN OAM frames is referred to as the flow entropy. The flow entropy can be populated so as to choose a particular path among multiple ECMP paths between a source and destination switch. The TTL expiry message may also be generated by the intermediate switches for the actual data frames. The same payload of the original path trace request is preserved for the payload of the response.

The traceroute and pathtrace messages are similar, except that traceroute uses the ICMP channel, whereas pathtrace use the NVO3 draft Tissa channel. Pathtrace uses the NVO3 draft Tissa channel, carrying additional diagnostic information, for example, interface load and statistics of the hops taken by these messages. If an

intermediate device does not support the NVO3 draft Tissa channel, the pathtrace behaves as a simple traceroute and it provides only the hop information.

Traceroute

Trace the path that is traversed by the packet in the VXLAN overlay using **Traceroute** command:

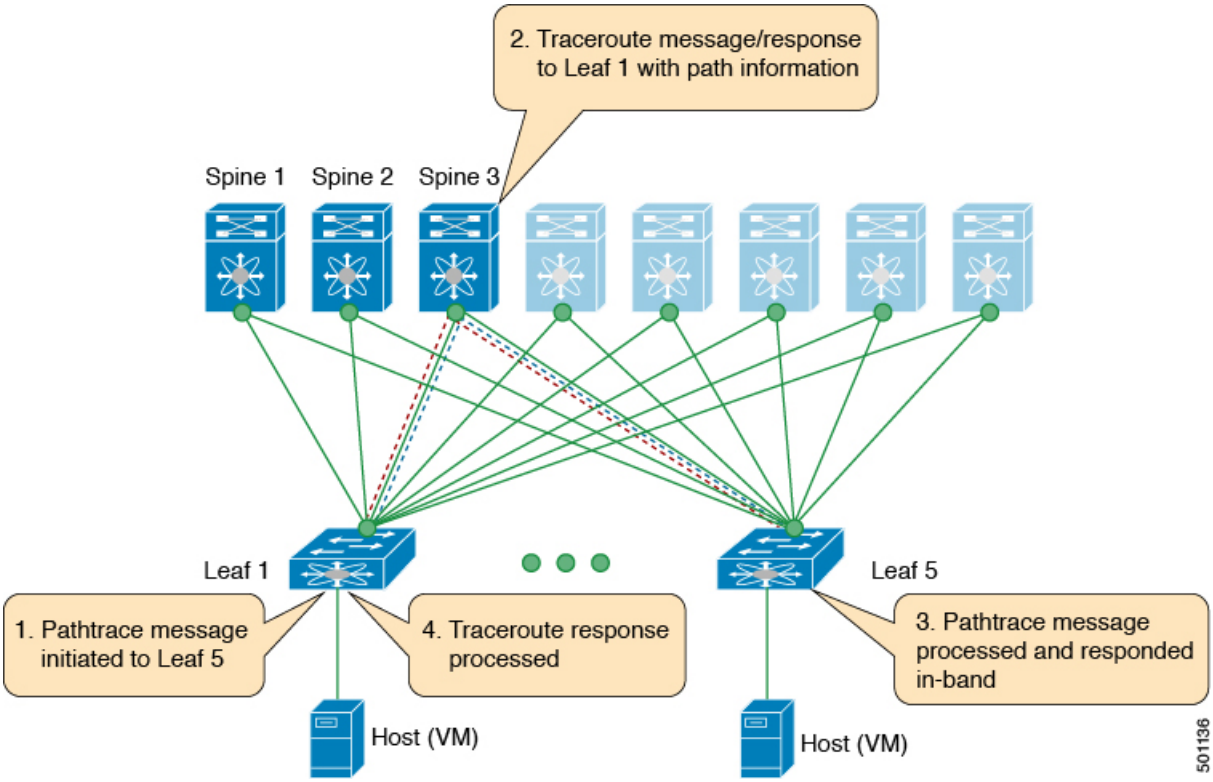
- Traceroute uses the ICMP packets (channel-1), encapsulated in the VXLAN encapsulation to reach the host

Pathtrace

Trace the path that is traversed by the packet in the VXLAN overlay using the NVO3 draft Tissa channel with **Pathtrace** command:

- Pathtrace uses special control packets like NVO3 draft Tissa or TISSA (channel-2) to provide additional information regarding the path (for example, ingress interface and egress interface). These packets terminate at VTEP and they does not reach the host. Therefore, only the VTEP responds.

Figure 6: Traceroute Message



Configuring VXLAN OAM

Before you begin

As a prerequisite, ensure that the VXLAN configuration is complete.

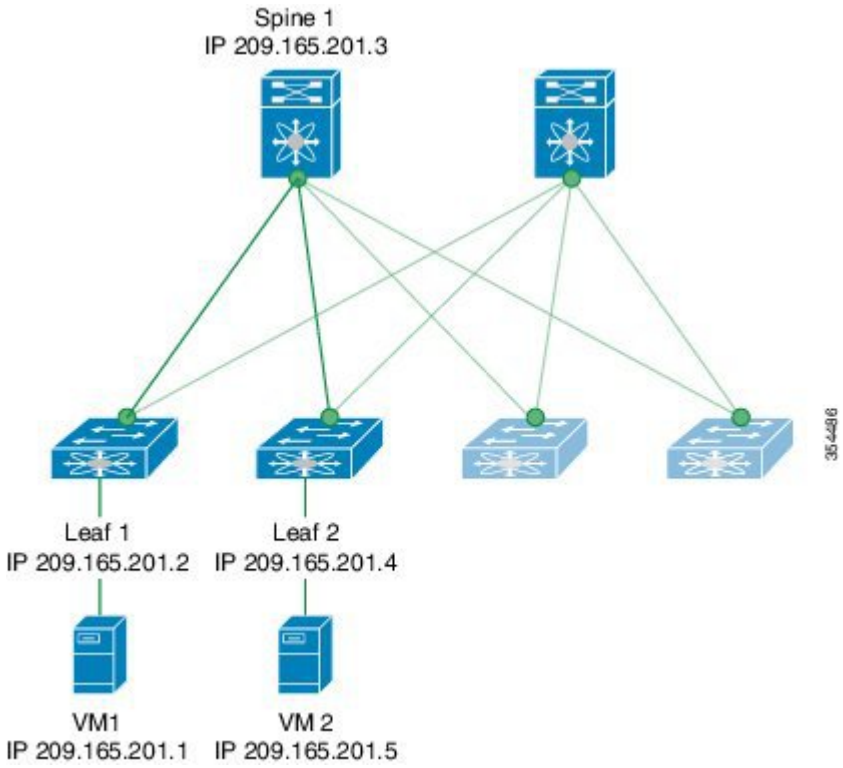
Procedure

	Command or Action	Purpose
Step 1	switch(config)# **feature ngoam**	Enters the NGOAM feature.
Step 2	switch(config)# **hardware access-list tcam region arp-ether 256 double-wide**	For Cisco Nexus 3000 Series switches with Network Forwarding Engine (NFE), configure the TCAM region for ARP-ETHER using this command. This step is essential to program the ACL rule in the hardware and it is a pre-requisite before installing the ACL rule. **Note** Configuring the TCAM region requires the node to be rebooted.
Step 3	switch(config)# **ngoam install acl**	Installs NGOAM Access Control List (ACL).
Step 4	(Optional) **#bcm-shell module 1 "fp show group 62"**	For Cisco Nexus 3000 Series switches with Network Forwarding Engine (NFE), complete this verification step. After entering the command, perform a lookup for entry/eid with data=0x8902 under EtherType.
Step 5	(Optional) **# show system internal access-list tcam ingress start-idx <hardware index> count 1**	

Example

See the following examples of the configuration topology.

Figure 7: VXLAN Network



VXLAN OAM provides the visibility of the host at the switch level, that allows a leaf to ping the host using the **ping nve** command.

The following example displays how to ping from Leaf 1 to VM2 via Spine 1.

```
switch# ping nve ip 209.165.201.5 vrf vni-31000 source 1.1.1.1 verbose

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
'c' - Corrupted Data/Test, '#' - Duplicate response


Sender handle: 34
! sport 40673 size 39,Reply from 209.165.201.5,time = 3 ms
! sport 40673 size 39,Reply from 209.165.201.5,time = 1 ms
! sport 40673 size 39,Reply from 209.165.201.5,time = 1 ms
! sport 40673 size 39,Reply from 209.165.201.5,time = 1 ms
! sport 40673 size 39,Reply from 209.165.201.5,time = 1 ms
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/18 ms
Total time elapsed 49 ms
```

Note The source ip-address 1.1.1.1 used in the above example is a loopback interface that is configured on Leaf 1 in the same VRF as the destination ip-address. For example, the VRF in this example is vni-31000.

The following example displays how to traceroute from Leaf 1 to VM 2 via Spine 1.

```
switch# traceroute nve ip 209.165.201.5 vrf vni-31000 source 1.1.1.1 verbose


Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
'c' - Corrupted Data/Test, '#' - Duplicate response


Traceroute request to peer ip 209.165.201.4 source ip 209.165.201.2
Sender handle: 36
  1 !Reply from 209.165.201.3,time = 1 ms
  2 !Reply from 209.165.201.4,time = 2 ms
  3 !Reply from 209.165.201.5,time = 1 ms
```

The following example displays how to pathtrace from Leaf 2 to Leaf 1.

```
switch# pathtrace nve ip 209.165.201.4 vni 31000 verbose

Path trace Request to peer ip 209.165.201.4 source ip 209.165.201.2

Sender handle: 42
TTL            Code        Reply
          IngressI/f               EgressI/f               State
=====================================================================
1     !Reply from 209.165.201.3,  Eth5/5/1      Eth5/5/2     UP/UP
2     !Reply from 209.165.201.4,  Eth1/3                    Unknown         UP/DOWN
```

The following example displays how to MAC ping from Leaf 2 to Leaf 1 using NVO3 draft Tissa channel:

```
switch# ping nve mac 0050.569a.7418 2901 ethernet 1/51 profile 4 verbose

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
'c' - Corrupted Data/Test, '#' - Duplicate response

Sender handle: 408
!!!!Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/5 ms
Total time elapsed 104 ms

switch# show run ngoam
feature ngoam
ngoam profile 4
oam-channel 2
ngoam install acl
```

The following example displays how to pathtrace based on a payload from Leaf 2 to Leaf 1:

```
switch# pathtrace nve ip unknown vrf vni-31000 payload mac-addr 0050.569a.d927 0050.569a.a4fa
ip 209.165.201.5 209.165.201.1 port 15334 12769 proto 17 payload-end

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
```

```
'c' - Corrupted Data/Test, '#' - Duplicate response

Path trace Request to peer ip 209.165.201.4 source ip 209.165.201.2
Sender handle: 46
TTL Code Reply IngressI/f EgressI/f State
=====================================================================
1 !Reply from 209.165.201.3, Eth5/5/1 Eth5/5/2 UP/UP
2 !Reply from 209.165.201.4, Eth1/3 Unknown UP/DOWN
```

Configuring NGOAM Profile

Complete the following steps to configure NGOAM profile.

Procedure

	Command or Action	Purpose
Step 1	switch(config)#**[no] feature ngoam**	Enables or disables NGOAM feature
Step 2	switch(config)#**[no] ngoam profile <profile-id>**	Configures OAM profile. The range for the profile-id is <1 – 1023>. This command does not have a default value. Enters the config-ngoam-profile submode to configure NGOAM specific commands. **Note** All profiles have default values and the **show run all** CLI command displays them. The default values are not visible through the **show run** CLI command.

Step 3	switch(config-ng-oam-profile)# **?**
	Example:
	`switch(config-ng-oam-profile)# ?`
	`  description  Configure description of the profile`
	`  dot1q  Encapsulation dot1q/bd`
	`  flow  Configure ngoam flow`
	`  hop  Configure ngoam hop count`
	`  interface  Configure ngoam egress interface`
	`  no  Negate a command or set its defaults`
	`  oam-channel  Oam-channel used`
	`  payload  Configure ngoam payload`
	`  sport  Configure ngoam Udp source port range`
		Displays the options for configuring NGOAM profile.

Example

See the following examples for configuring an NGOAM profile and for configuring NGOAM flow.

```
switch(config)#
ngoam profile 1
oam-channel 1
flow forward
payload pad 0x2
sport 12345, 54321

switch(config-ngoam-profile)#flow {forward }
Enters config-ngoam-profile-flow submode to configure forward flow entropy specific
information
```

NGOAM Authentication

NGOAM provides the interface statistics in the pathtrace response. Beginning with Cisco NX-OS Release 7.0(3)I6(1), NGOAM authenticates the pathtrace requests to provide the statistics by using the HMAC MD5 authentication mechanism.

NGOAM authentication validates the pathtrace requests before providing the interface statistics. NGOAM authentication takes effect only for the pathtrace requests with **req-stats** option. All the other commands are not affected with the authentication configuration. If NGOAM authentication key is configured on the requesting node, NGOAM runs the MD5 algorithm using this key to generate the 16-bit MD5 digest. This digest is encoded as type-length-value (TLV) in the pathtrace request messages.

When the pathtrace request is received, NGOAM checks for the **req-stats** option and the local NGOAM authentication key. If the local NGOAM authentication key is present, it runs MD5 using the local key on the request to generate the MD5 digest. If both digests match, it includes the interface statistics. If both digests do not match, it sends only the interface names. If an NGOAM request comes with the MD5 digest but no local authentication key is configured, it ignores the digest and sends all the interface statistics. To secure an entire network, configure the authentication key on all nodes.

To configure the NGOAM authentication key, use the **ngoam authentication-key** <*key*> CLI command. Use the **show running-config ngoam** CLI command to display the authentication key.

```
switch# show running-config ngoam
!Time: Tue Mar 28 18:21:50 2017
version 7.0(3)I6(1)
feature ngoam
ngoam profile 1
  oam-channel 2
ngoam profile 3
ngoam install acl
ngoam authentication-key 987601ABCDEF
```

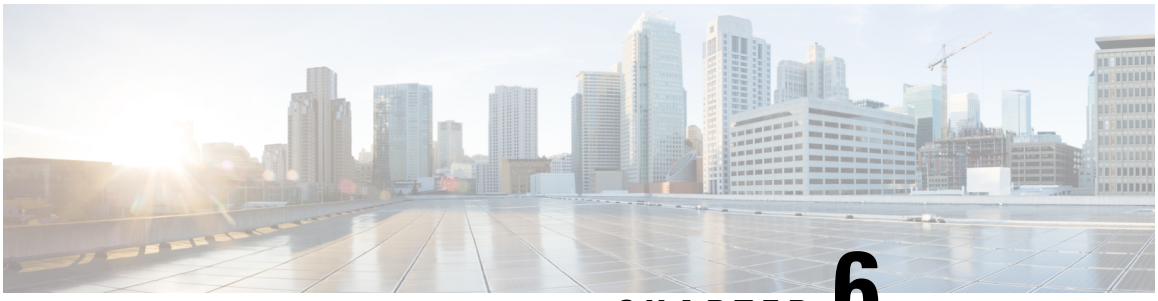
In the following example, the same authentication key is configured on the requesting switch and the responding switch.

```
switch# pathtrace nve ip 12.0.22.1 profile 1 vni 31000 req-stats ver
Path trace Request to peer ip 12.0.22.1 source ip 11.0.22.1
Hop   Code   ReplyIP   IngressI/f  EgressI/f   State
```

```
================================================
  1 !Reply from 55.55.55.2, Eth5/7/1  Eth5/7/2  UP / UP
  Input Stats: PktRate:0 ByteRate:0 Load:0 Bytes:339573434 unicast:14657 mcast:307581
bcast:67 discards:0 errors:3 unknown:0 bandwidth:42949672970000000
Output Stats: PktRate:0 ByteRate:0 load:0 bytes:237399176 unicast:2929 mcast:535710
bcast:10408 discards:0 errors:0 bandwidth:42949672970000000
  2 !Reply from 12.0.22.1, Eth1/7  Unknown  UP / DOWN
  Input Stats: PktRate:0 ByteRate:0 Load:0 Bytes:4213416 unicast:275 mcast:4366 bcast:3
discards:0 errors:0 unknown:0 bandwidth:42949672970000000
switch# conf t
switch(config)# no ngoam authentication-key 123456789
switch(config)# end
```

In the following example, an authentication key is not configured on the requesting switch. Therefore, the responding switch does not send any interface statistics. The intermediate node does not have any authentication key configured and it always replies with the interface statistics.

```
switch# pathtrace nve ip 12.0.22.1 profile 1 vni 31000 req-stats ver
Path trace Request to peer ip 12.0.22.1 source ip 11.0.22.1
Sender handle: 10
Hop   Code   ReplyIP   IngressI/f  EgressI/f   State
================================================
  1 !Reply from 55.55.55.2, Eth5/7/1  Eth5/7/2  UP / UP
  Input Stats: PktRate:0 ByteRate:0 Load:0 Bytes:339580108 unicast:14658 mcast:307587
bcast:67 discards:0 errors:3 unknown:0 bandwidth:42949672970000000
Output Stats: PktRate:0 ByteRate:0 load:0 bytes:237405790 unicast:2929 mcast:535716
bcast:10408 discards:0 errors:0 bandwidth:42949672970000000
  2 !Reply from 12.0.22.1, Eth1/17  Unknown  UP / DOWN
```



CHAPTER 6

Configuring Tenant Routed Multicast

This chapter contains the following sections:

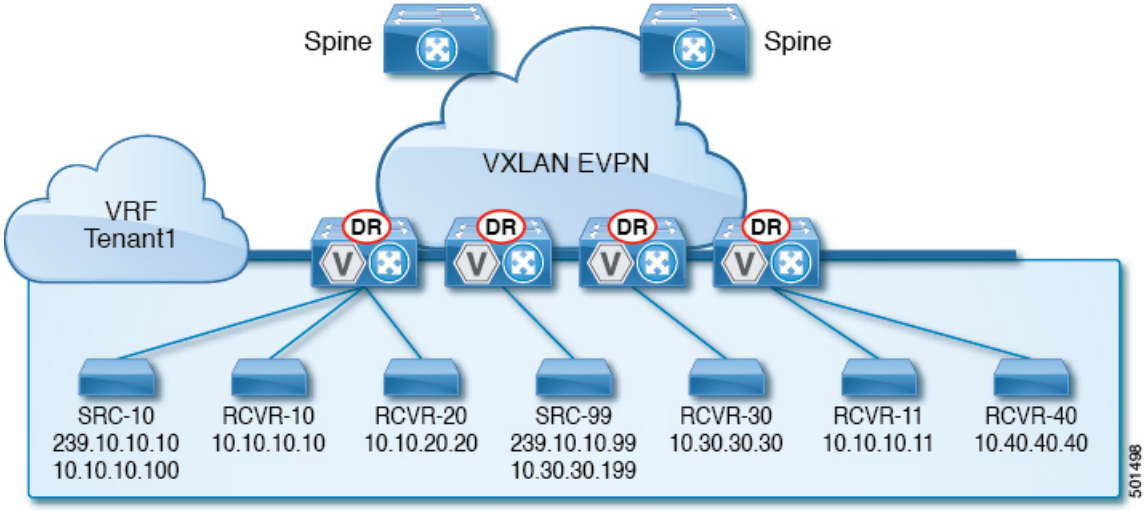
About Tenant Routed Multicast

Tenant Routed Multicast (TRM) enables multicast forwarding on the VXLAN fabric that uses a BGP-based EVPN control plane. TRM provides multi-tenancy aware multicast forwarding between senders and receivers within the same or different subnet local or across VTEPs.

This feature brings the efficiency of multicast delivery to VXLAN overlays. It is based on the standards-based next generation control plane (ngMVPN) described in IETF RFC 6513, 6514. TRM enables the delivery of customer IP multicast traffic in a multitenant fabric, and thus in an efficient and resilient manner. The delivery of TRM improves Layer-3 overlay multicast functionality in our networks.

While BGP EVPN provides the control plane for unicast routing, ngMVPN provides scalable multicast routing functionality. It follows an “always route” approach where every edge device (VTEP) with distributed IP Anycast Gateway for unicast becomes a Designated Router (DR) for Multicast. Bridged multicast forwarding is only present on the edge-devices (VTEP) where IGMP snooping optimizes the multicast forwarding to interested receivers. Every other multicast traffic beyond local delivery is efficiently routed.

Figure 8: VXLAN EVPN TRM



With TRM enabled, multicast forwarding in the underlay is leveraged to replicate VXLAN encapsulated routed multicast traffic. A Default Multicast Distribution Tree (Default-MDT) is built per-VRF. This is an addition to the existing multicast groups for Layer-2 VNI Broadcast, Unknown Unicast, and Layer-2 multicast replication group. The individual multicast group addresses in the overlay are mapped to the respective underlay multicast address for replication and transport. The advantage of using a BGP-based approach allows the VXLAN BGP EVPN fabric with TRM to operate as fully distributed Overlay Rendezvous-Point (RP), with the RP presence on every edge-device (VTEP).

A multicast-enabled data center fabric is typically part of an overall multicast network. Multicast sources, receivers, and multicast rendezvous points, might reside inside the data center but might also be inside the campus or externally reachable via the WAN. TRM allows a seamless integration with existing multicast networks. It can leverage multicast rendezvous points external to the fabric. Furthermore, TRM allows for tenant-aware external connectivity using Layer-3 physical interfaces or subinterfaces.

Guidelines and Limitations for Tenant Routed Multicast

Tenant Routed Multicast (TRM) has the following guidelines and limitations:

- The Guidelines and Limitations for VXLANs, on page 6 also apply to TRM.
- With TRM enabled, SVI as a core link is not supported.
- TRM supports IPv4 multicast only.
- TRM requires an IPv4 multicast-based underlay using PIM Any Source Multicast (ASM) which is also known as sparse mode.
- TRM supports overlay PIM ASM and PIM SSM only. PIM BiDir is not supported in the overlay.
- RP has to be configured either internal or external to the fabric.
- The internal RP must be configured on all TRM-enabled VTEPs including the border nodes.
- The external RP must be external to the border nodes.

- The RP must be configured within the VRF pointing to the external RP IP address (static RP). This ensures that unicast and multicast routing is enabled to reach the external RP in the given VRF.
- TRM supports multiple border nodes. Reachability to an external RP via multiple border leaf switches is supported (ECMP).

Guidelines and Limitations for Layer 3 Tenant Routed Multicast

Layer 3 Tenant Routed Multicast (TRM) has the following configuration guidelines and limitations:

- Beginning with Cisco NX-OS Release 9.3(3), the Cisco Nexus 3132-Z switch supports TRM VXLAN BGP EVPN.
- Beginning with Cisco NX-OS Release 9.3(3), the Cisco Nexus 3132-Z switch supports TRM in Layer 3 mode. For this platform, Layer 3 TRM is supported on IPv4 overlays only. Layer 2 mode and L2/L3 mixed mode are not supported.

  The Cisco Nexus 3132-Z switch supports TRM as a border leaf in non-vPC mode. TRM is not supported on vPC border leafs on Cisco Nexus 3132-Z switches.

  The Cisco Nexus 3132-Z switch can function as a border leaf for Layer 3 unicast traffic. For Anycast functionality, the RP can be internal, external, or RP everywhere.
- Beginning with Cisco NX-OS Release 9.3(3), the Cisco Nexus 3132-Z switch supports RP everywhere for PIM Anycast and RP everywhere for MSDP peering.
- Well-known local scope multicast (224.0.0.0/24) is excluded from TRM and is bridged.
- When an interface NVE is brought down on the border leaf, the internal overlay RP per VRF must be brought down.

Rendezvous Point for Tenant Routed Multicast

With TRM enabled Internal and External RP is supported. The following table displays the first release in which RP positioning is or is not supported.

	RP Internal	RP External	PIM-Based RP Everywhere
TRM L3 Mode	9.3(3) for Cisco Nexus 3132-Z switches	9.3(3) for Cisco Nexus 3132-Z switches	9.3(3) for Cisco Nexus 3132-Z switches

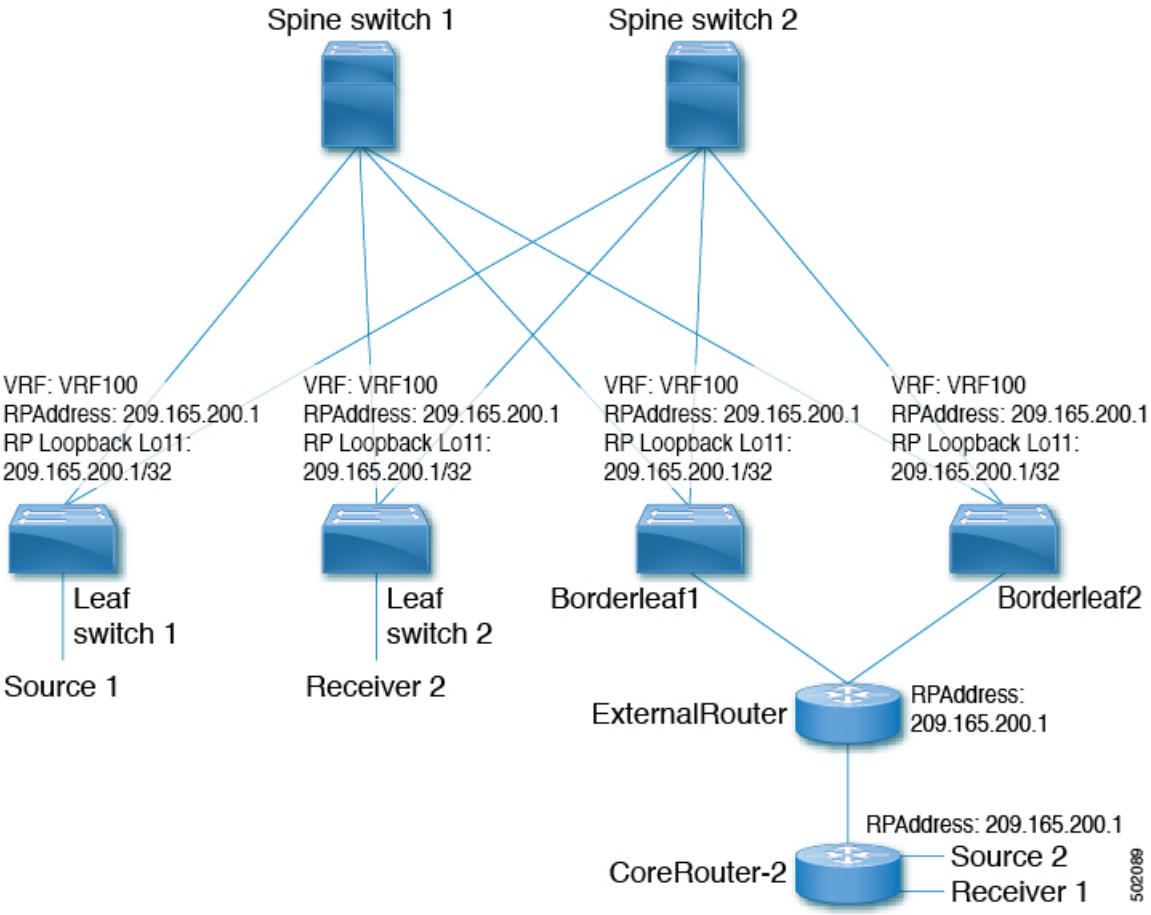
Configuring a Rendezvous Point for Tenant Routed Multicast

For Tenant Routed Multicast, the following rendezvous point options are supported:

- Configuring a Rendezvous Point Inside the VXLAN Fabric, on page 82
- Configuring an External Rendezvous Point, on page 83

Configuring a Rendezvous Point Inside the VXLAN Fabric

Configure the loopback for the TRM VRFs with the following commands on all devices (VTEP). Ensure it is reachable within EVPN (advertise/redistribute).



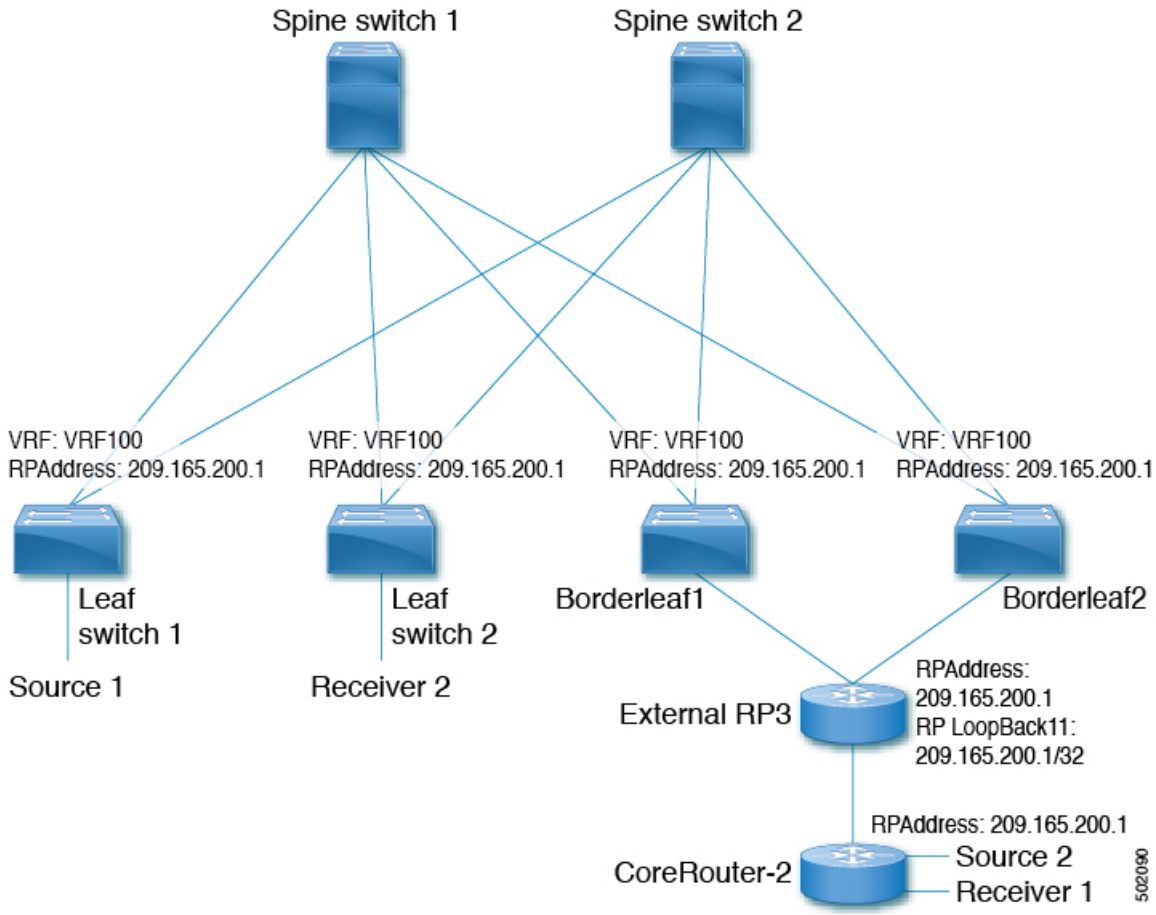
Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 11`	Configure the loopback interface on all TRM-enabled nodes. This enables the rendezvous point inside the fabric.
Step 3	**vrf member** *vxlan-number* **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 4	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.200.1/32`	Specify IP address.
Step 5	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.
Step 6	**vrf context** *vrf-name* **Example:** `switch(config-if)# vrf context vrf100`	Create a VXLAN tenant VRF.
Step 7	**ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix* **Example:** `switch(config-vrf# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4`	The value of the *ip-address-of-router* parameter is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.

Configuring an External Rendezvous Point

Configure the external rendezvous point (RP) IP address within the TRM VRFs on all devices (VTEP). In addition, ensure reachability of the external RP within the VRF via the border node.

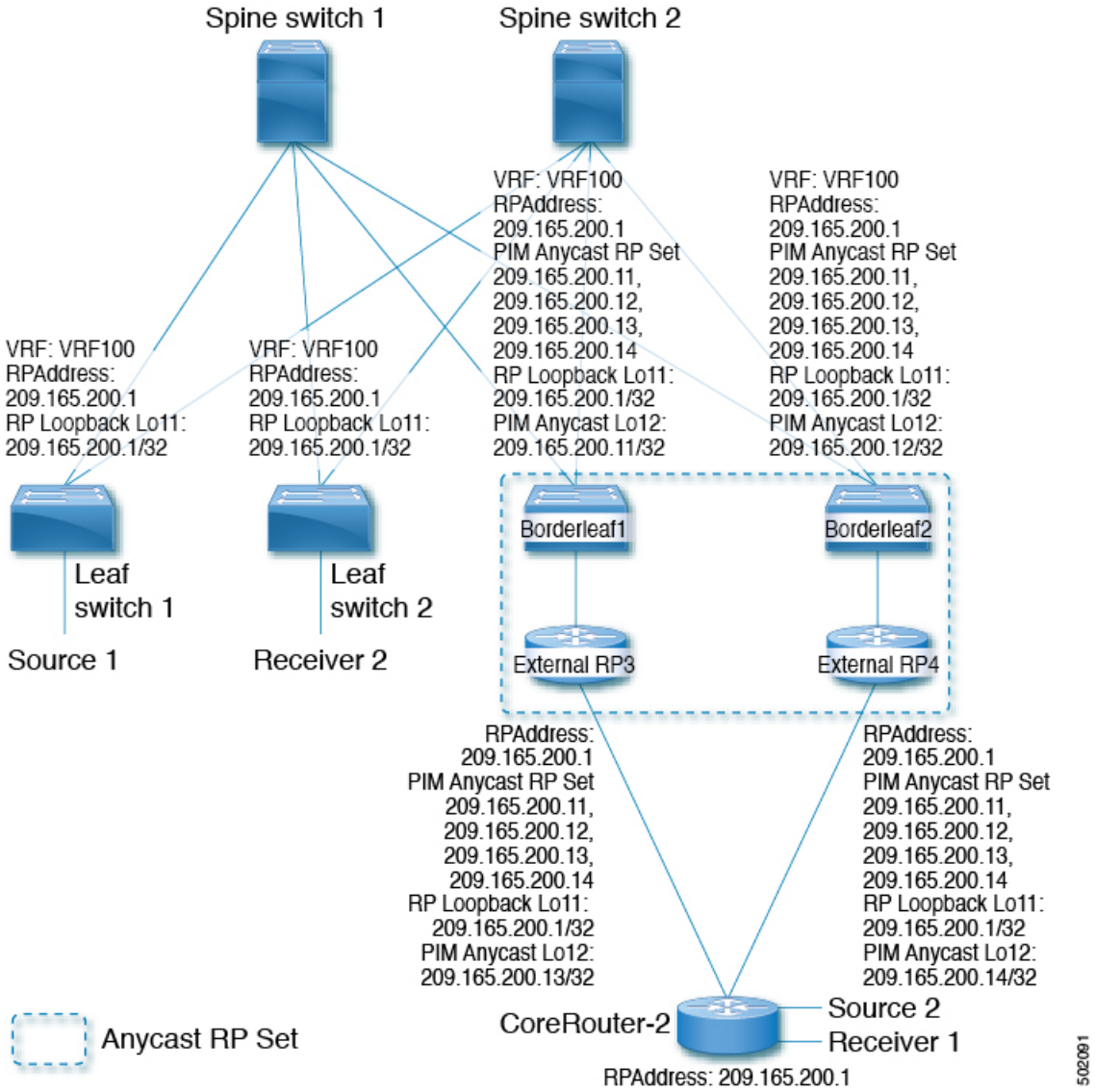


Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.
Step 2	**vrf context vrf100** **Example:** `switch(config)# vrf context vrf100`	Enter configuration mode.
Step 3	**ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix* **Example:** `switch(config-vrf)# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4`	The value of the *ip-address-of-router* parameter is that of the RP. The same IP address must be on all of the edge devices (VTEPs) for a fully distributed RP.

Configuring RP Everywhere with PIM Anycast

RP Everywhere configuration with PIM Anycast solution.



For information about configuring RP Everywhere with PIM Anycast, see:

- Configuring a TRM Leaf Node for RP Everywhere with PIM Anycast, on page 86
- Configuring a TRM Border Leaf Node for RP Everywhere with PIM Anycast, on page 86
- Configuring an External Router for RP Everywhere with PIM Anycast, on page 88

Configuring a TRM Leaf Node for RP Everywhere with PIM Anycast

Configuration of Tenant Routed Multicast (TRM) leaf node for RP Everywhere.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.
Step 2	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 11`	Configure the loopback interface on all VXLAN VTEP devices.
Step 3	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 4	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.200.1/32`	Specify IP address.
Step 5	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.
Step 6	**vrf context** *vxlan* **Example:** `switch(config-if)# vrf context vrf100`	Create a VXLAN tenant VRF.
Step 7	**ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix* **Example:** `switch(config-vrf# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4`	The value of the *ip-address-of-router* parameters is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.

Configuring a TRM Border Leaf Node for RP Everywhere with PIM Anycast

Configuring the TRM Border Leaf Node for RP Anywhere with PIM Anycast.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.
Step 2	**ip pim evpn-border-leaf** **Example:** `switch(config)# ip pim evpn-border-leaf`	Configure VXLAN VTEP as TRM border leaf node,
Step 3	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 11`	Configure the loopback interface on all VXLAN VTEP devices.
Step 4	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 5	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.200.1/32`	Specify IP address.
Step 6	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.
Step 7	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 12`	Configure the PIM Anycast set RP loopback interface.
Step 8	**vrf member** *vxlan-number* **Example:** `switch(config-if)# vrf member vxlan-number`	Configure VRF name.
Step 9	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.200.11/32`	Specify IP address.
Step 10	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.

	Command or Action	Purpose
Step 11	**vrf context** *vrf-name* **Example:** `switch(config-if)# vrf context vrf100`	Create a VXLAN tenant VRF.
Step 12	**ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix* **Example:** `switch(config-vrf)# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4`	The value of the *ip-address-of-router* parameters is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.
Step 13	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.11`	Configure PIM Anycast RP set.
Step 14	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.12`	Configure PIM Anycast RP set.
Step 15	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.13`	Configure PIM Anycast RP set.
Step 16	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.14`	Configure PIM Anycast RP set.

Configuring an External Router for RP Everywhere with PIM Anycast

Use this procedure to configure an external router for RP Everywhere.

Procedure

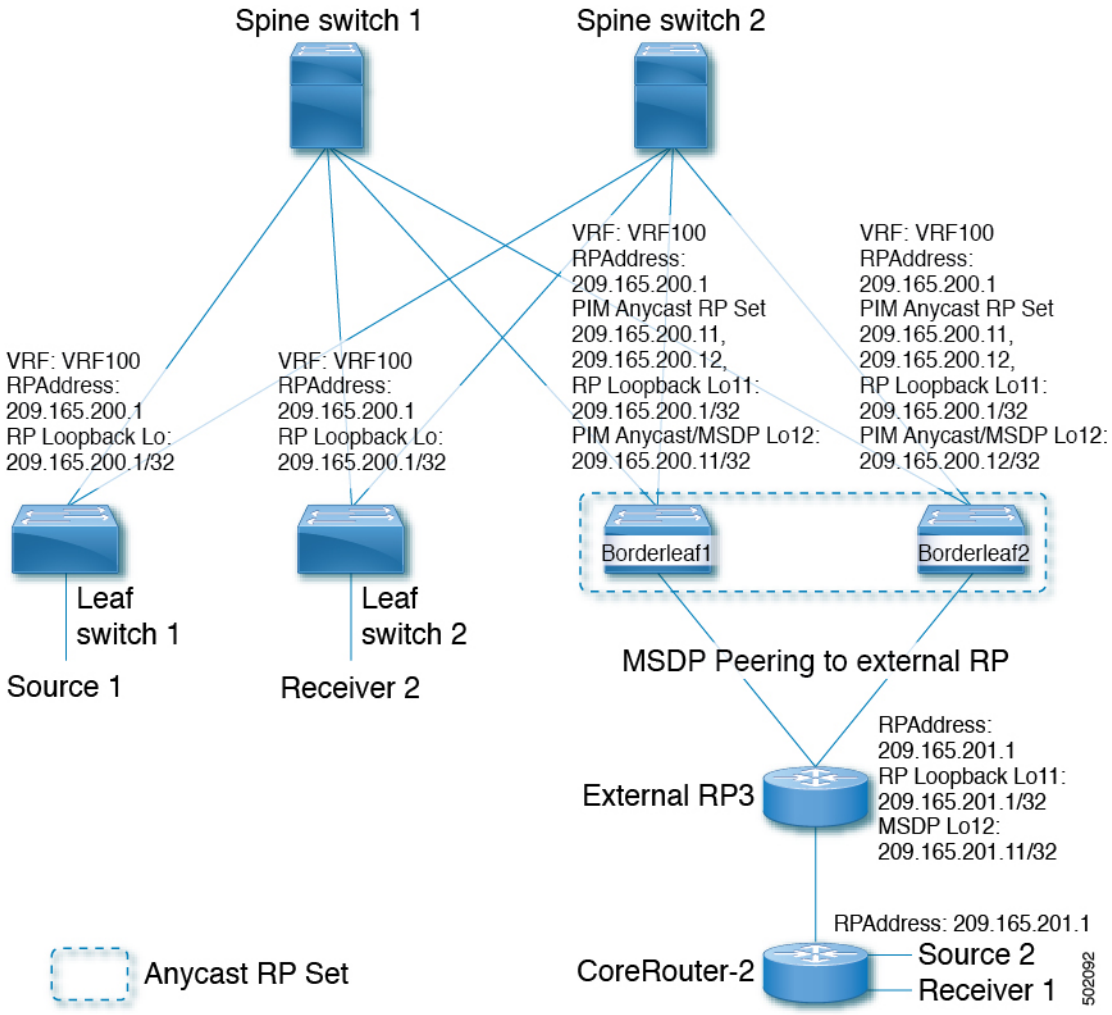
	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.

	Command or Action	Purpose
Step 2	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 11`	Configure the loopback interface on all VXLAN VTEP devices.
Step 3	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member vfr100`	Configure VRF name.
Step 4	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.200.1/32`	Specify IP address.
Step 5	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.
Step 6	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 12`	Configure the PIM Anycast set RP loopback interface.
Step 7	**vrf member** *vxlan-number* **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 8	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.200.13/32`	Specify IP address.
Step 9	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.
Step 10	**vrf context** *vxlan* **Example:** `switch(config-if)# vrf context vrf100`	Create a VXLAN tenant VRF.
Step 11	**ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix* **Example:** `switch(config-vrf)# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4`	The value of the *ip-address-of-router* parameters is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.

	Command or Action	Purpose
Step 12	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.11`	Configure PIM Anycast RP set.
Step 13	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.12`	Configure PIM Anycast RP set.
Step 14	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.13`	Configure PIM Anycast RP set.
Step 15	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.14`	Configure PIM Anycast RP set.

Configuring RP Everywhere with MSDP Peering

RP Everywhere configuration with MSDP RP solution.



For information about configuring RP Everywhere with MSDP Peering, see:

- Configuring a TRM Leaf Node for RP Everywhere with MSDP Peering, on page 91
- Configuring a TRM Border Leaf Node for RP Everywhere with MSDP Peering, on page 92
- Configuring an External Router for RP Everywhere with MSDP Peering, on page 94

Configuring a TRM Leaf Node for RP Everywhere with MSDP Peering

Configuring a TRM leaf node for RP Everywhere with MSDP peering.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.

	Command or Action	Purpose
Step 2	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 11`	Configure the loopback interface on all VXLAN VTEP devices.
Step 3	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 4	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.200.1/32`	Specify IP address.
Step 5	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.
Step 6	**vrf context** *vrf-name* **Example:** `switch(config-if)# vrf context vrf100`	Create a VXLAN tenant VRF.
Step 7	**ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix* **Example:** `switch(config-vrf# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4`	The value of the *ip-address-of-router* parameters is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.

Configuring a TRM Border Leaf Node for RP Everywhere with MSDP Peering

Use this procedure to configure a TRM border leaf for RP Everywhere with PIM Anycast.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.
Step 2	**feature msdp** **Example:** `switch(config)# feature msdp`	Enable feature MSDP.
Step 3	**ip pim evpn-border-leaf** **Example:**	Configure VXLAN VTEP as TRM border leaf node,

	Command or Action	Purpose
	`switch(config)# ip pim evpn-border-leaf`	
Step 4	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 11`	Configure the loopback interface on all VXLAN VTEP devices.
Step 5	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 6	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.200.1/32`	Specify IP address.
Step 7	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.
Step 8	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 12`	Configure the PIM Anycast set RP loopback interface.
Step 9	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 10	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.200.11/32`	Specify IP address.
Step 11	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.
Step 12	**vrf context** *vrf-name* **Example:** `switch(config-if)# vrf context vrf100`	Create a VXLAN tenant VRF.
Step 13	**ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix* **Example:** `switch(config-vrf)# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4`	The value of the *ip-address-of-router* parameter is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.

	Command or Action	Purpose
Step 14	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.11`	Configure PIM Anycast RP set.
Step 15	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.12`	Configure PIM Anycast RP set.
Step 16	**ip msdp originator-id** *loopback* **Example:** `switch(config-vrf)# ip msdp originator-id loopback12`	Configure MSDP originator ID.
Step 17	**ip msdp peer** *ip-address* **connect-source** *loopback* **Example:** `switch(config-vrf)# ip msdp peer 209.165.201.11 connect-source loopback12`	Configure MSDP peering between border node and external RP router.

Configuring an External Router for RP Everywhere with MSDP Peering

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.
Step 2	**feature msdp** **Example:** `switch(config)# feature msdp`	Enable feature MSDP.
Step 3	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 11`	Configure the loopback interface on all VXLAN VTEP devices.
Step 4	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.

	Command or Action	Purpose
Step 5	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.201.1/32`	Specify IP address.
Step 6	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.
Step 7	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 12`	Configure the PIM Anycast set RP loopback interface.
Step 8	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 9	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 209.165.201.11/32`	Specify IP address.
Step 10	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on an interface.
Step 11	**vrf context** *vrf-name* **Example:** `switch(config-if)# vrf context vrf100`	Create a VXLAN tenant VRF.
Step 12	**ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix* **Example:** `switch(config-vrf)# ip pim rp-address 209.165.201.1 group-list 224.0.0.0/4`	The value of the *ip-address-of-router* parameters is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.
Step 13	**ip msdp originator-id loopback12** **Example:** `switch(config-vrf)# ip msdp originator-id loopback12`	Configure MSDP originator ID.
Step 14	**ip msdp peer** *ip-address* **connect-source loopback12** **Example:**	Configure MSDP peering between external RP router and all TRM border nodes.

	Command or Action	Purpose
	`switch(config-vrf)# ip msdp peer 209.165.200.11 connect-source loopback12`	

Configuring Layer 3 Tenant Routed Multicast

This procedure enables the Tenant Routed Multicast (TRM) feature. TRM operates primarily in the Layer 3 forwarding mode for IP multicast by using BGP MVPN signaling. TRM in Layer 3 mode is the main feature and the only requirement for TRM enabled VXLAN BGP EVPN fabrics. If non-TRM capable edge devices (VTEPs) are present, the Layer 2/Layer 3 mode and Layer 2 mode have to be considered for interop.

To forward multicast between senders and receivers on the Layer 3 cloud and the VXLAN fabric on TRM vPC border leafs, the VIP/PIP configuration must be enabled. For more information, see Configuring VIP/PIP.

Note TRM follows an always-route approach and hence decrements the Time to Live (TTL) of the transported IP multicast traffic.

Before you begin

VXLAN EVPN **feature nv overlay** and **nv overlay evpn** must be configured.

The rendezvous point (RP) must be configured.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.
Step 2	**feature ngmvpn** **Example:** `switch(config)# feature ngmvpn`	Enables the Next-Generation Multicast VPN (ngMVPN) control plane. New address family commands become available in BGP.
Step 3	**ip igmp snooping vxlan** **Example:** `switch(config)# ip igmp snooping vxlan`	Configure IGMP snooping for VXLAN VLANs.
Step 4	**interface nve1** **Example:** `switch(config)# interface nve 1`	Configure the NVE interface.

	Command or Action	Purpose
Step 5	**member vni** *vni-range* **associate-vrf** **Example:** `switch(config-if-nve)# member vni 200100 associate-vrf`	Configure the Layer 3 virtual network identifier. The range of *vni-range* is from 1 to 16,777,214.
Step 6	**mcast-group** *ip-prefix* **Example:** `switch(config-if-nve-vni)# mcast-group 225.3.3.3`	Builds the default multicast distribution tree for the VRF VNI (Layer 3 VNI). The multicast group is used in the underlay (core) for all multicast routing within the associated Layer 3 VNI (VRF). **Note** We recommend that underlay multicast groups for Layer 2 VNI, default MDT, and data MDT not be shared. Use separate, non-overlapping groups.
Step 7	**exit** **Example:** `switch(config-if-nve-vni)# exit`	Exits command mode.
Step 8	**exit** **Example:** `switch(config-if)# exit`	Exits command mode.
Step 9	**router bgp 100** **Example:** `switch(config)# router bgp 100`	Set autonomous system number.
Step 10	**exit** **Example:** `switch(config-router)# exit`	Exits command mode.
Step 11	**neighbor** *ip-addr* **Example:** `switch(config-router)# neighbor 1.1.1.1`	Configure IP address of the neighbor.
Step 12	**address-family ipv4 mvpn** **Example:** `switch(config-router-neighbor)# address-family ipv4 mvpn`	Configure multicast VPN.
Step 13	**send-community extended** **Example:**	Enables ngMVPN for address family signalization. The **send community extended** command ensures that extended communities are exchanged for this address family.

	Command or Action	Purpose
	`switch(config-router-neighbor-af)# send-community extended`	
Step 14	**exit** **Example:** `switch(config-router-neighbor-af)# exit`	Exits command mode.
Step 15	**exit** **Example:** `switch(config-router)# exit`	Exits command mode.
Step 16	**vrf context** *vrf_name* **Example:** `switch(config-router)#vrf context vrf100`	Configure VRF name.
Step 17	**ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix* **Example:** `switch(config-vrf)# ip pim rp-address 209.165.201.1 group-list 226.0.0.0/8`	The value of the *ip-address-of-router* parameter is that of the RP. The same IP address must be on all of the edge devices (VTEPs) for a fully distributed RP. For overlay RP placement options, see the Configuring a Rendezvous Point for Tenant Routed Multicast, on page 81 section.
Step 18	**address-family ipv4 unicast** **Example:** `switch(config-vrf)# address-family ipv4 unicast`	Configure unicast address family.
Step 19	**route-target both auto mvpn** **Example:** `switch(config-vrf-af-ipv4)# route-target both auto mvpn`	Defines the BGP route target that is added as an extended community attribute to the customer multicast (C_Multicast) routes (ngMVPN route type 6 and 7). Auto route targets are constructed by the 2-byte Autonomous System Number (ASN) and Layer 3 VNI.
Step 20	**ip multicast overlay-spt-only** **Example:** `switch(config)# ip multicast overlay-spt-only`	Gratuitously originate (S,A) route when the source is locally connected. The **ip multicast overlay-spt-only** command is enabled by default on all MVPN-enabled Cisco Nexus 3000 Series switches (typically leaf node).
Step 21	**interface***vlan_id* **Example:** `switch(config)# interface vlan11`	Configures the first-hop gateway (distributed anycast gateway for the Layer 2 VNI. No router PIM peering must ever happen with this interface.

	Command or Action	Purpose
Step 22	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Disables an interface.
Step 23	**vrf member** *vrf-num* **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 24	**ip address** *ip_address* **Example:** `switch(config-if)# ip address 11.1.1.1/24`	Configure IP address.
Step 25	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Enables IGMP and PIM on the SVI. This is required is multicast sources and/or receivers exist in this VLAN.
Step 26	**fabric forwarding mode anycast-gateway** **Example:** `switch(config-if)# fabric forwarding mode anycast-gateway`	Configure Anycast Gateway Forwarding Mode.
Step 27	**ip pim neighbor-policy NONE*** **Example:** `switch(config-if)# ip pim neighbor-policy NONE*`	Creates an IP PIM neighbor policy to avoid PIM neighborship with PIM routers within the VLAN. The **none** keyword is a configured route map to deny any ipv4 addresses to avoid establishing PIM neighborship policy using anycase IP. **Note** Do not use Distributed Anycast Gateway for PIM Peerings.
Step 28	**exit** **Example:** `switch(config-if)# exit`	Exits command mode.
Step 29	**interface** *vlan_id* **Example:** `switch(config)# interface vlan100`	Configure Layer 3 VNI.
Step 30	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Disable an interface.
Step 31	**vrf member vrf100** **Example:**	Configure VRF name.

	Command or Action	Purpose
	`switch(config-if)# vrf member vrf100`	
Step 32	**ip forward** **Example:** `switch(config-if)# ip forward`	Enable IP forwarding on interface.
Step 33	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on interface. There is no PIM peering happening in the Layer-3 VNI, but this command must be present for forwarding.

Configuring TRM on the VXLAN EVPN Spine

This procedure enables Tenant Routed Multicast (TRM) on a VXLAN EVPN spine switch.

Before you begin

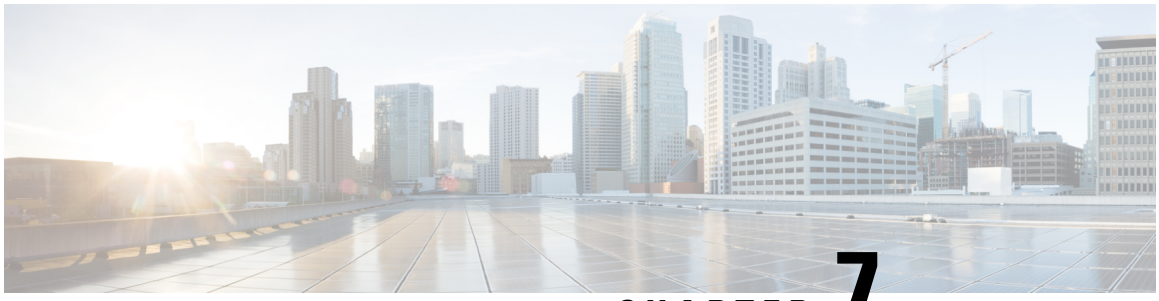
The VXLAN BGP EVPN spine must be configured.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.
Step 2	**route-map permitall permit 10** **Example:** `switch(config)# route-map permitall permit 10`	Configure the route-map. **Note** The route-map keeps the next-hop unchanged for EVPN routes • Required for eBGP • Options for iBGP
Step 3	**set ip next-hop unchanged** **Example:** `switch(config-route-map)# set ip next-hop unchanged`	Set next hop address. **Note** The route-map keeps the next-hop unchanged for EVPN routes • Required for eBGP • Options for iBGP
Step 4	**exit** **Example:** `switch(config-route-map)# exit`	Return to exec mode.

	Command or Action	Purpose
Step 5	**router bgp [autonomous system]** *number* **Example:** `switch(config)# router bgp 65002`	Specify BGP.
Step 6	**address-family ipv4 mvpn** **Example:** `switch(config-router)# address-family ipv4 mvpn`	Configure the address family IPv4 MVPN under the BGP.
Step 7	**retain route-target all** **Example:** `switch(config-router-af)# retain route-target all`	Configure retain route-target all under address-family IPv4 MVPN [global]. **Note** Required for eBGP. Allows the spine to retain and advertise all MVPN routes when there are no local VNIs configured with matching import route targets.
Step 8	**neighbor** *ip-address* [**remote-as** *number*] **Example:** `switch(config-router-af)# neighbor 100.100.100.1`	Define neighbor.
Step 9	**address-family ipv4 mvpn** **Example:** `switch(config-router-neighbor)# address-family ipv4 mvpn`	Configure address family IPv4 MVPN under the BGP neighbor.
Step 10	**disable-peer-as-check** **Example:** `switch(config-router-neighbor-af)# disable-peer-as-check`	Disables checking the peer AS number during route advertisement. Configure this parameter on the spine for eBGP when all leafs are using the same AS but the spines have a different AS than leafs. **Note** Required for eBGP.
Step 11	**rewrite-rt-asn** **Example:** `switch(config-router-neighbor-af)# rewrite-rt-asn`	Normalizes the outgoing route target's AS number to match the remote AS number. Uses the BGP configured neighbors remote AS. The **rewrite-rt-asn** command is required if the route target auto feature is being used to configure EVPN route targets.
Step 12	**send-community extended** **Example:** `switch(config-router-neighbor-af)# send-community extended`	Configures community for BGP neighbors.
Step 13	**route-reflector-client**	Configure route reflector.

	Command or Action	Purpose
	Example: `switch(config-router-neighbor-af)# route-reflector-client`	**Note** Required for iBGP with route-reflector.
Step 14	**route-map permitall out** **Example:** `switch(config-router-neighbor-af)# route-map permitall out`	Applies route-map to keep the next-hop unchanged. **Note** Required for eBGP.



CHAPTER 7

Configuring VXLAN Multihoming

This chapter contains the following sections:

VXLAN EVPN Multihoming Overview

Introduction to Multihoming

Cisco Nexus platforms support vPC-based multihoming, where a pair of switches act as a single device for redundancy and both switches function in an active mode. With Cisco Nexus 31128PQ switch and 3100-V platform switches in VXLAN BGP EVPN environment, there are two solutions to support Layer 2 multihoming; the solutions are based on the Traditional vPC (emulated or virtual IP address) and the BGP EVPN techniques.

Traditional vPC utilizes a consistency check that is a mechanism used by the two switches that are configured as a vPC pair to exchange and verify their configuration compatibility. The BGP EVPN technique does not have the consistency check mechanism, but it uses LACP to detect the misconfigurations. It also eliminates the MCT link that is traditionally used by vPC and it offers more flexibility as each VTEP can be a part of one or more redundancy groups. It can potentially support many VTEPs in a given group.

BGP EVPN Multihoming Terminology

See this section for the terminology used in BGP EVPN multihoming:

- EVI: EVPN instance represented by the VNI.
- MAC-VRF: A container to house virtual forwarding table for MAC addresses. A unique route distinguisher and import/export target can be configured per MAC-VRF.
- ES: Ethernet Segment that can constitute a set of bundled links.
- ESI: Ethernet Segment Identifier to represent each ES uniquely across the network.

EVPN Multihoming Implementation

The EVPN overlay draft specifies adaptations to the BGP MPLS based EVPN solution to enable it to be applied as a network virtualization overlay with VXLAN encapsulation. The Provider Edge (PE) node role in BGP MPLS EVPN is equivalent to VTEP/Network Virtualization Edge device (NVE), where VTEPs use control plane learning and distribution via BGP for remote addresses instead of data plane learning.

There are 5 different route types currently defined:

- Ethernet Auto-Discovery (EAD) Route
- MAC advertisement Route
- Inclusive Multicast Route
- Ethernet Segment Route
- IP Prefix Route

BGP EVPN running on Cisco NX-OS uses route type-2 to advertise MAC and IP (host) information, route type-3 to carry VTEP information (specifically for ingress replication), and the EVPN route type-5 allows advertisements of IPv4 or IPv6 prefixes in an Network Layer Reachability Information (NLRI) with no MAC addresses in the route key.

With the introduction of EVPN multihoming, Cisco NX-OS software utilizes Ethernet Auto-discovery (EAD) route, where Ethernet Segment Identifier and the Ethernet Tag ID are considered to be part of the prefix in the NLRI. Since the end points reachability is learned via the BGP control plane, the network convergence time is a function of the number of MAC/IP routes that must be withdrawn by the VTEP in case of a failure scenario. To deal with such condition, each VTEP advertises a set of one or more Ethernet Auto-Discovery per ES routes for each locally attached Ethernet Segment and upon a failure condition to the attached segment, the VTEP withdraws the corresponding set of Ethernet Auto-Discovery per ES routes.

Ethernet Segment Route is the other route type that is being used by Cisco NX-OS software with EVPN multihoming, mainly for Designated Forwarder (DF) election for the BUM traffic. If the Ethernet Segment is multihomed, the presence of multiple DFs could result in forwarding the loops in addition to the potential packet duplication. Therefore, the Ethernet Segment Route (Type 4) is used to elect the Designated Forwarder and to apply Split Horizon Filtering. All VTEPs/PEs that are configured with an Ethernet Segment originate this route.

To summarize the new implementation concepts for the EVPN multihoming:

- EAD/ES: Ethernet Auto Discovery Route per ES that is also referred to as type-1 route. This route is used to converge the traffic faster during access failure scenarios. This route has Ethernet Tag of 0xFFFFFFFF.
- EAD/EVI: Ethernet Auto Discovery Route per EVI that is also referred to as type-1 route. This route is used for aliasing and load balancing when the traffic only hashes to one of the switches. This route cannot have Ethernet Tag value of 0xFFFFFF to differentiate it from the EAD/ES route.
- ES: Ethernet Segment route that is also referred to as type-4 route. This route is used for DF election for BUM traffic.
- Aliasing: It is used for load balancing the traffic to all the connected switches for a given Ethernet Segment using the type-1 EAD/EVI route. This is done irrespective of the switch where the hosts are actually learned.

- Mass Withdrawal: It is used for fast convergence during the access failure scenarios using the type-1 EAD/ES route.
- DF Election: It is used to prevent forwarding of the loops and the duplicates as only a single switch is allowed to decap and forward the traffic for a given Ethernet Segment.
- Split Horizon: It is used to prevent forwarding of the loops and the duplicates for the BUM traffic. Only the BUM traffic that originates from a remote site is allowed to be forwarded to a local site.

EVPN Multihoming Redundancy Group

Consider the dually homed topology, where switches L1 and L2 are distributed anycast VXLAN gateways that perform Integrated Routing and Bridging (IRB). Host H2 is connected to an access switch that is dually homed to both L1 and L2.

The access switch is connected to L1 and L2 via a bundled pair of physical links. The switch is not aware that the bundle is configured on two different devices on the other side. However, both L1 and L2 must be aware that they are a part of the same bundle.

Note that there is no Multichassis EtherChannel Trunk (MCT) link between L1 and L2 switches and each switch can have similar multiple bundle links that are shared with the same set of neighbors.

To make the switches L1 and L2 aware that they are a part of the same bundle link, the NX-OS software utilizes the Ethernet Segment Identifier (ESI) and the system MAC address (system-mac) that is configured under the interface (PO).

Ethernet Segment Identifier

EVPN introduces the concept of Ethernet Segment Identifier (ESI). Each switch is configured with a 10 byte ESI value under the bundled link that they share with the multihomed neighbor. The ESI value can be manually configured or auto-derived.

LACP Bundling

LACP can be turned ON for detecting ESI misconfigurations on the multihomed port channel bundle as LACP sends the ESI configured MAC address value to the access switch. LACP is not mandated along with ESI. A given ESI interface (PO) shares the same ESI ID across the VTEPs in the group.

The access switch receives the same configured MAC value from both switches (L1 and L2). Therefore, it puts the bundled link in the UP state. Since the ES MAC can be shared across all the Ethernet-segments on the switch, LACP PDUs use ES MAC as system MAC address and the admin_key carries the ES ID.

Cisco recommends running LACP between the switches and the access devices since LACP PDUs have a mechanism to detect and act on the misconfigured ES IDs. In case there is mismatch on the configured ES ID under the same PO, LACP brings down one of the links (first link that comes online stays up). By default, on most Cisco Nexus platforms, LACP sets a port to the suspended state if it does not receive an LACP PDU from the peer. This is based on the **lacp suspend-individual** command that is enabled by default. This command helps in preventing loops that are created due to the ESI configuration mismatch. Therefore, it is recommended to enable this command on the port-channels on the access switches and the servers.

In some scenarios (for example, POAP or NetBoot), it can cause the servers to fail to boot up because they require LACP to logically bring up the port. In case you are using static port channel and you have mismatched ES IDs, the MAC address gets learned from both L1 and L2 switches. Therefore, both the switches advertise

the same MAC address belonging to different ES IDs that triggers the MAC address move scenario. Eventually, no traffic is forwarded to that node for the MAC addresses that are learned on both L1 and L2 switches.

Guidelines and Limitations for VXLAN EVPN Multihoming

See the following limitations for configuring VXLAN EVPN multihoming:

- EVPN multihoming is supported on the Cisco Nexus 3100-V and 3132-Z platform switches only.
- ARP suppression is supported with EVPN multihoming.
- EVPN multihoming is supported with multihoming to two switches only.
- Switchport trunk native VLAN is not supported on the trunk interfaces.
- Cisco recommends enabling LACP on ES PO.
- IPV6 is currently not supported.

Configuring VXLAN EVPN Multihoming

Enabling EVPN Multihoming

Cisco NX-OS allows either vPC based EVPN multihoming or ESI based EVPN multihoming. Both features should not be enabled together. ESI based multihoming is enabled using **evpn esi multihoming** CLI command. It is important to note that the command for ESI multihoming enables the Ethernet-segment configurations and the generation of Ethernet-segment routes on the switches.

The receipt of type-1 and type-2 routes with valid ESI and the path-list resolution are not tied to the **evpn esi multihoming** command. If the switch receives MAC/MAC-IP routes with valid ESI and the command is not enabled, the ES based path resolution logic still applies to these remote routes. This is required for interoperability between the vPC enabled switches and the ESI enabled switches.

Complete the following steps to configure EVPN multihoming:

Before you begin

VXLAN should be configured with BGP-EVPN before enabling EVPN ESI multihoming.

Procedure

	Command or Action	Purpose
Step 1	**evpn esi multihoming**	Enables EVPN multihoming globally.
Step 2	**ethernet-segment delay-restore time 30**	The ESI Port Channel remains down for 30 seconds after the core facing interfaces are up.
Step 3	**vlan-consistency-check**	Enables VLAN consistency check.
Step 4	**address-family l2vpn evpn maximum-paths <>maximum-paths ibgp <>**	Enables BGP maximum-path to enable ECMP for the MAC routes. Otherwise, the MAC routes

	Command or Action	Purpose

	Example:
	`address-family l2vpn evpn`
	`maximum-paths 64`
	`maximum-paths ibgp 64`
		have only 1 VTEP as the next-hop. This configuration is needed under BGP in Global level.

Step 5	**evpn multihoming core-tracking**	Enables EVPN multihoming core-links. It tracks the uplink interfaces towards the core. If all uplinks are down, the local ES based the POs is shut down/suspended. This is mainly used to avoid black-holing South-to-North traffic when no uplinks are available.

Step 6	**interface port-channel Ethernet-segment <>System-mac <>**
	Example:
	`ethernet-segment 11`
	`system-mac 0000.0000.0011`
		Configures the local Ethernet Segment ID. The ES ID has to match on VTEPs where the PO is multihomed. The Ethernet Segment ID should be unique per PO.
		Configures the local system-mac ID that has to match on the VTEPs where the PO is multihomed. The system-mac address can be shared across multiple POs.

Step 7	**hardware access-list tcam region vpc-convergence 256** **Example:** `hardware access-list tcam region vpc-convergence 256`	Configures the TCAM. This command is used to configure the split horizon ACLs in the hardware. This command avoids BUM traffic duplication on the shared ES POs.

VXLAN EVPN Multihoming Configuration Examples

See the sample VXLAN EVPN multihoming configuration on the switches:

```
Switch 1 (L1)

evpn esi multihoming

ethernet-segment delay-restore time 180
vlan-consistency-check
router bgp 1001
   address-family l2vpn evpn
   maximum-paths ibgp 2

interface Ethernet2/1
  no switchport
  evpn multihoming core-tracking
  mtu 9216
  ip address 10.1.1.1/30
  ip pim sparse-mode
  no shutdown

interface Ethernet2/2
```

```
  no switchport
  evpn multihoming core-tracking
  mtu 9216
  ip address 10.1.1.5/30
  ip pim sparse-mode
  no shutdown

interface port-channel11
  switchport mode trunk
  switchport trunk allowed vlan 901-902,1001-1050
  ethernet-segment 2011
    system-mac 0000.0000.2011
  mtu 9216


Switch 2 (L2)

evpn esi multihoming

ethernet-segment delay-restore time 180
vlan-consistency-check
router bgp 1001
   address-family l2vpn evpn
   maximum-paths ibgp 2

interface Ethernet2/1
  no switchport
  evpn multihoming core-tracking
  mtu 9216
  ip address 10.1.1.2/30
  ip pim sparse-mode
  no shutdown

interface Ethernet2/2
  no switchport
  evpn multihoming core-tracking
  mtu 9216
  ip address 10.1.1.6/30
  ip pim sparse-mode
  no shutdown

interface port-channel11
  switchport mode trunk
  switchport access vlan 1001
  switchport trunk allowed vlan 901-902,1001-1050
  ethernet-segment 2011
    system-mac 0000.0000.2011
  mtu 9216
```

Configuring Layer 2 Gateway STP

Layer 2 Gateway STP Overview

EVPN multihoming is supported with the Layer 2 Gateway Spanning Tree Protocol (L2G-STP). The Layer 2 Gateway Spanning Tree Protocol (L2G-STP) builds a loop-free tree topology. However, the Spanning Tree Protocol root must always be in the VXLAN fabric. A bridge ID for the Spanning Tree Protocol consists of

a MAC address and the bridge priority. When the system is running in the VXLAN fabric, the system automatically assigns the VTEPs with the MAC address c84c.75fa.6000 from a pool of reserved MAC addresses. As a result, each switch uses the same MAC address for the bridge ID emulating a single logical pseudo root.

The Layer 2 Gateway Spanning Tree Protocol (L2G-STP) is disabled by default on EVPN ESI multihoming VLANs. Use the **spanning-tree domain enable** CLI command to enable L2G-STP on all VTEPs. With L2G-STP enabled, the VXLAN fabric (all VTEPs) emulates a single pseudo root switch for the customer access switches. The L2G-STP is initiated to run on all VXLAN VLANs by default on boot up and the root is fixed on the overlay. With L2G-STP, the root-guard gets enabled by default on all the access ports. Use **spanning-tree domain** <*id*> to additionally enable Spanning Tree Topology Change Notification(STP-TCN), to be tunneled across the fabric.

All the access ports from VTEPs connecting to the customer access switches are in a *desg* forwarding state by default. All ports on the customer access switches connecting to VTEPs are either in root-port forwarding or alt-port blocking state. The root-guard kicks in if better or superior STP information is received from the customer access switches and it puts the ports in the *blk l2g_inc* state to secure the root on the overlay-fabric and to prevent a loop.

Guidelines for Moving to Layer 2 Gateway STP

Complete the following steps to move to Layer 2 gateway STP:

- With Layer 2 Gateway STP, root guard is enabled by default on all the access ports.
- With Layer 2 Gateway STP enabled, the VXLAN fabric (all VTEPs) emulates a single pseudo-root switch for the customer access switches.
- All access ports from VTEPs connecting to the customer access switches are in the **Desg FWD** state by default.
- All ports on customer access switches connecting to VTEPs are either in the root-port FWD or Altn BLK state.
- Root guard is activated if superior spanning-tree information is received from the customer access switches. This process puts the ports in **BLK L2GW_Inc** state to secure the root on the VXLAN fabric and prevent a loop.
- Explicit domain ID configuration is needed to enable spanning-tree BPDU tunneling across the fabric.
- As a best practice, you should configure all VTEPs with the lowest spanning-tree priority of all switches in the spanning-tree domain to which they are attached. By setting all the VTEPs as the root bridge, the entire VXLAN fabric appears to be one virtual bridge.
- ESI interfaces should not be enabled in spanning-tree edge mode to allow Layer 2 Gateway STP to run across the VTEP and access layer.
- You can continue to use ESIs or orphans (single-homed hosts) in spanning-tree edge mode if they directly connect to hosts or servers that do not run Spanning Tree Protocol and are end hosts.
- Configure all VTEPs that are connected by a common customer access layer in the same Layer 2 Gateway STP domain. Ideally, all VTEPs on the fabric on which the hosts reside and to which the hosts can move.
- The Layer 2 Gateway STP domain scope is global, and all ESIs on a given VTEP can participate in only one domain.

- Mappings between Multiple Spanning Tree (MST) instances and VLANs must be consistent across the VTEPs in a given Layer 2 Gateway STP domain.
- Non-Layer 2 Gateway STP enabled VTEPs cannot be directly connected to Layer 2 Gateway STP-enabled VTEPs. Performing this action results in conflicts and disputes because the non-Layer 2 Gateway STP VTEP keeps sending BPDUs and it can steer the root outside.
- Keep the current edge and the BPDU filter configurations on both the Cisco Nexus switches and the access switches after upgrading to the latest build.
- Enable Layer 2 Gateway STP on all the switches with a recommended priority and the *mst* instance mapping as needed. Use the commands **spanning-tree domain enable** and **spanning-tree mst <instance-id's> priority 8192**.
- Remove the BPDU filter configurations on the switch side first.
- Remove the BPDU filter configurations and the edge on the customer access switch.

  Now the topology converges with Layer 2 Gateway STP and any blocking of the redundant connections is pushed to the access switch layer.

Enabling Layer 2 Gateway STP on a Switch

Complete the following steps to enable Layer 2 Gateway STP on a switch.

Procedure

	Command or Action	Purpose
Step 1	**spanning-tree mode <rapid-pvst, mst>**	Enables Spanning Tree Protocol mode.
Step 2	**spanning-tree domain enable**	Enables Layer 2 Gateway STP on a switch. It disables Layer 2 Gateway STP on all EVPN ESI multihoming VLANs.
Step 3	**spanning-tree domain 1**	Explicit domain ID is needed to tunnel encoded BPDUs to the core and processes received from the core.
Step 4	**spanning-tree mst <id> priority 8192**	Configures Spanning Tree Protocol priority.
Step 5	**spanning-tree vlan <id> priority 8192**	Configures Spanning Tree Protocol priority.
Step 6	**spanning-tree domain disable**	Disables Layer 2 Gateway STP on a VTEP.

Example

All Layer 2 Gateway STP VLANs should be set to a lower spanning-tree priority than the customer-edge (CE) topology to help ensure that the VTEP is the spanning-tree root for this VLAN. If the access switches have a higher priority, you can set the Layer 2 Gateway STP priority to 0 to retain the Layer 2 Gateway STP root in the VXLAN fabric. See the following configuration example:

```
switch# show spanning-tree summary
Switch is in mst mode (IEEE Standard)
Root bridge for: MST0000
L2 Gateway STP bridge for: MST0000
L2 Gateway Domain ID: 1
Port Type Default                        is disable
Edge Port [PortFast] BPDU Guard Default  is disabled
Edge Port [PortFast] BPDU Filter Default is disabled
Bridge Assurance                         is enabled
Loopguard Default                        is disabled
Pathcost method used                     is long
PVST Simulation                          is enabled
STP-Lite                                 is disabled

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
MST0000                       0         0        0         12         12
---------------------- -------- --------- -------- ---------- ----------
1 mst                         0         0        0         12         12


switch# show spanning-tree vlan 1001

MST0000
  Spanning tree enabled protocol mstp

  Root ID    Priority    8192
             Address     c84c.75fa.6001   L2G-STP reserved mac+ domain id
             This bridge is the root
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    8192   (priority 8192 sys-id-ext 0)
             Address     c84c.75fa.6001
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
```

The output displays that the spanning-tree priority is set to 8192 (the default is 32768). Spanning-tree priority is set in multiples of 4096. The priority for individual instances is calculated as the priority and the Instance_ID. In this case, the priority is calculated as 8192 + 0 = 8192. With Layer 2 Gateway STP, access ports (VTEP ports connected to the access switches) have root guard enabled. If a superior BPDU is received on an edge port of a VTEP, the port is placed in the Layer 2 Gateway inconsistent state until the condition is cleared as displayed in the following example:

```
2016 Aug 29 19:14:19 TOR9-leaf4 %$ VDC-1 %$ %STP-2-L2GW_BACKBONE_BLOCK: L2 Gateway Backbone
 port inconsistency blocking port Ethernet1/1 on MST0000.
2016 Aug 29 19:14:19 TOR9-leaf4 %$ VDC-1 %$ %STP-2-L2GW_BACKBONE_BLOCK: L2 Gateway Backbone
 port inconsistency blocking port port-channel13 on MST0000.

switch# show spanning-tree

MST0000
  Spanning tree enabled protocol mstp
  Root ID    Priority    8192
             Address     c84c.75fa.6001
             This bridge is the root
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    8192   (priority 8192 sys-id-ext 0)
             Address     c84c.75fa.6001
```

```
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Po1              Desg FWD 20000     128.4096 Edge P2p
Po2              Desg FWD 20000     128.4097 Edge P2p
Po3              Desg FWD 20000     128.4098 Edge P2p
Po12            Desg BKN*2000       128.4107 P2p *L2GW_Inc
Po13            Desg BKN*1000       128.4108 P2p *L2GW_Inc
Eth1/1        Desg BKN*2000       128.1    P2p *L2GW_Inc
```

To disable Layer 2 Gateway STP on a VTEP, enter the **spanning-tree domain disable** CLI command. This command disables Layer 2 Gateway STP on all EVPN ESI multihomed VLANs. The bridge MAC address is restored to the system MAC address, and the VTEP may not necessarily be the root. In the following case, the access switch has assumed the root role because Layer 2 Gateway STP is disabled:

```
switch(config)# spanning-tree domain disable

switch# show spanning-tree summary
Switch is in mst mode (IEEE Standard)
Root bridge for: none
L2 Gateway STP                          is disabled
Port Type Default                       is disable
Edge Port [PortFast] BPDU Guard Default  is disabled
Edge Port [PortFast] BPDU Filter Default is disabled
Bridge Assurance                        is enabled
Loopguard Default                       is disabled
Pathcost method used                    is long
PVST Simulation                         is enabled
STP-Lite                                is disabled

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
MST0000                       4         0        0          8         12
---------------------- -------- --------- -------- ---------- ----------
1 mst                         4         0        0          8         12


switch# show spanning-tree vlan 1001

MST0000
  Spanning tree enabled protocol mstp
  Root ID    Priority    4096
             Address     00c8.8ba6.5073
             Cost        0
             Port        4108 (port-channel13)
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    8192   (priority 8192 sys-id-ext 0)
             Address     5897.bd1d.db95
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
```

With Layer 2 Gateway STP, the access ports on VTEPs cannot be in an edge port, because they behave like normal spanning-tree ports, receiving BPDUs from the access switches. In that case, the access ports on VTEPs lose the advantage of rapid transmission, instead forwarding on Ethernet segment link flap. (They have to go through a proposal and agreement handshake before assuming the FWD-Desg role).

Configuring VXLAN EVPN Multihoming Traffic Flows

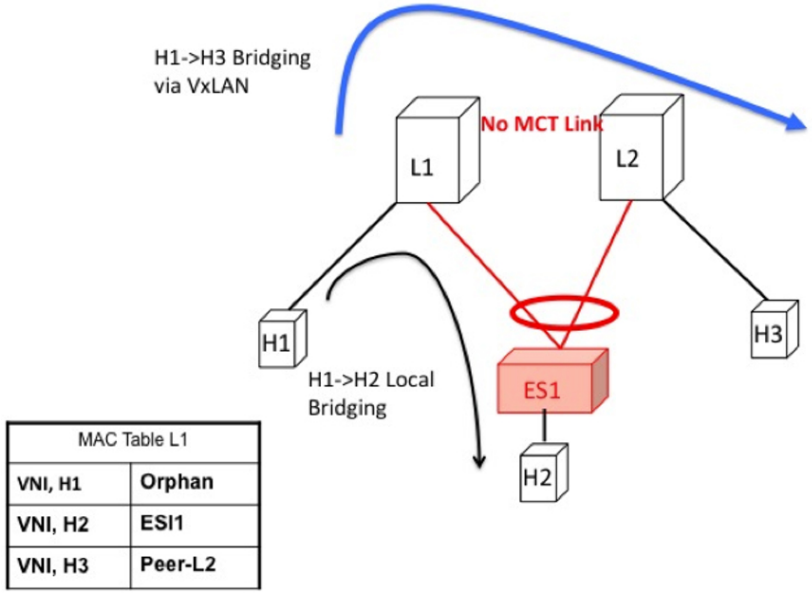
EVPN Multihoming Local Traffic Flows

All switches that are a part of the same redundancy group (as defined by the ESI) act as a single virtual switch with respect to the access switch/host. However, there is no MCT link present to bridge and route the traffic for local access.

Locally Bridged Traffic

Host H2 is dually homed whereas hosts H1 and H3 are single-homed (also known as orphans). The traffic is bridged locally from H1 to H2 via L1. However, if the packet needs to be bridged between the orphans H1 and H3, the packet must be bridged via the VXLAN overlay.

Figure 9: Local Bridging at L1. H1->H3 bridging via VXLAN. In vPC, H1->H3 will be via MCT link.



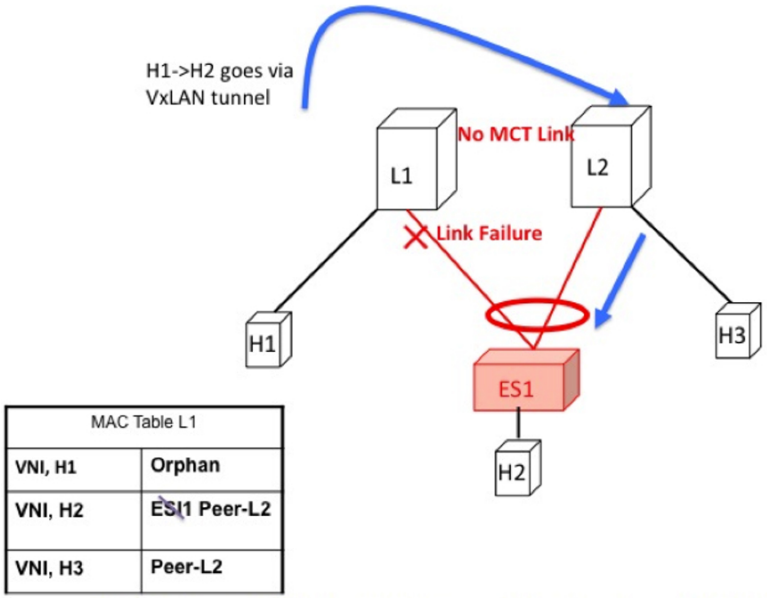
MAC Table L1	
VNI, H1	Orphan
VNI, H2	ESI1
VNI, H3	Peer-L2

Access Failure for Locally Bridged Traffic

If the ESI link at L1 fails, there is no path for the bridged traffic to reach from H1 to H2 except via the overlay. Therefore, the local bridged traffic takes the sub-optimal path, similar to the H1 to H3 orphan flow.

Note When such condition occurs, the MAC table entry for H2 changes from a local route pointing to a port channel interface to a remote overlay route pointing to peer-ID of L2. The change gets percolated in the system from BGP.

Figure 10: ES1 failure on L1. H1->H2 is now bridged over VXLAN tunnel.

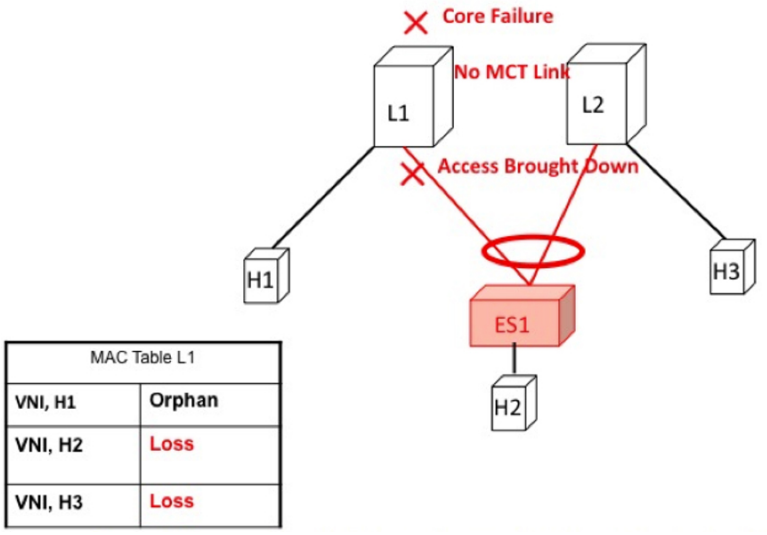


MAC Table L1	
VNI, H1	Orphan
VNI, H2	ESI1 Peer-L2
VNI, H3	Peer-L2

Core Failure for Locally Bridged Traffic

If switch L1 gets isolated from the core, it must not continue to attract access traffic, as it will not be able to encapsulate and send it on the overlay. This means that the access links must be brought down at L1 if L1 loses core reachability. In this scenario, orphan H1 loses all connectivity to both remote and locally attached hosts since there is no dedicated MCT link.

Figure 11: Core failure on L1. H1->H2 loses all connectivity as there is no MCT.



MAC Table L1	
VNI, H1	Orphan
VNI, H2	Loss
VNI, H3	Loss

Locally Routed Traffic

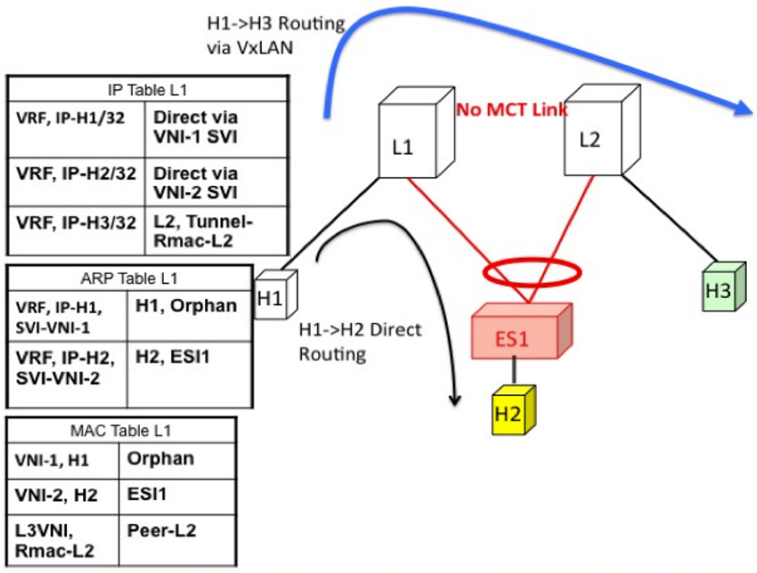
Consider H1, H2, and H3 being in different subnets and L1/L2 being distributed anycast gateways.

Any packet that is routed from H1 to H2 is directly sent from L1 via native routing.

However, host H3 is not a locally attached adjacency, unlike in vPC case where the ARP entry syncs to L1 as a locally attached adjacency. Instead, H3 shows up as a remote host in the IP table at L1, installed in the context of L3 VNI. This packet must be encapsulated in the router-MAC of L2 and routed to L2 via VXLAN overlay.

Therefore, routed traffic from H1 to H3 takes place exactly in the same fashion as routed traffic between truly remote hosts in different subnets.

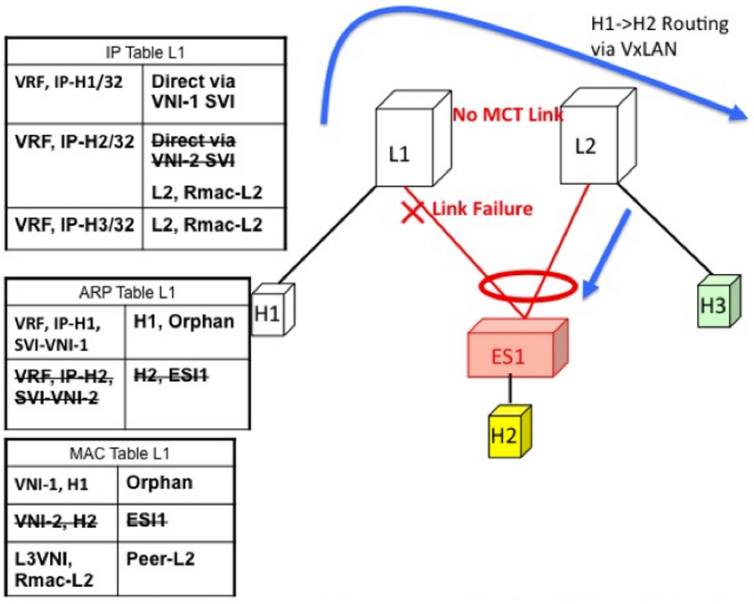
Figure 12: L1 is Distributed Anycast Gateway. H1, H2, and H3 are in different VLANs. H1->H3 routing happens via VXLAN tunnel encapsulation. In VPC, H3 ARP would have been synced via MCT and direct routing.



Access Failure for Locally Routed Traffic

In case the ESI link at switch L1 fails, there is no path for the routed traffic to reach from H1 to H2 except via the overlay. Therefore, the local routed traffic takes the sub-optimal path, similar to the H1 to H3 orphan flow.

Figure 13: H1, H2, and H3 are in different VLANs. ESI fails on L1. H1->H2 routing happens via VXLAN tunnel encapsulation.

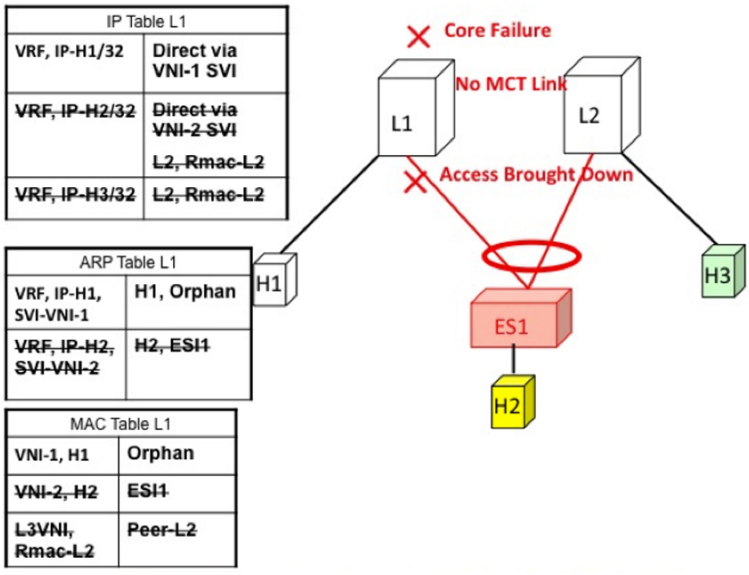


Core Failure for Locally Routed Traffic

If switch L1 gets isolated from the core, it must not continue to attract access traffic, as it will not be able to encapsulate and send it on the overlay. It means that the access links must be brought down at L1 if L1 loses core reachability.

In this scenario, orphan H1 loses all connectivity to both remote and locally attached hosts as there is no dedicated MCT link.

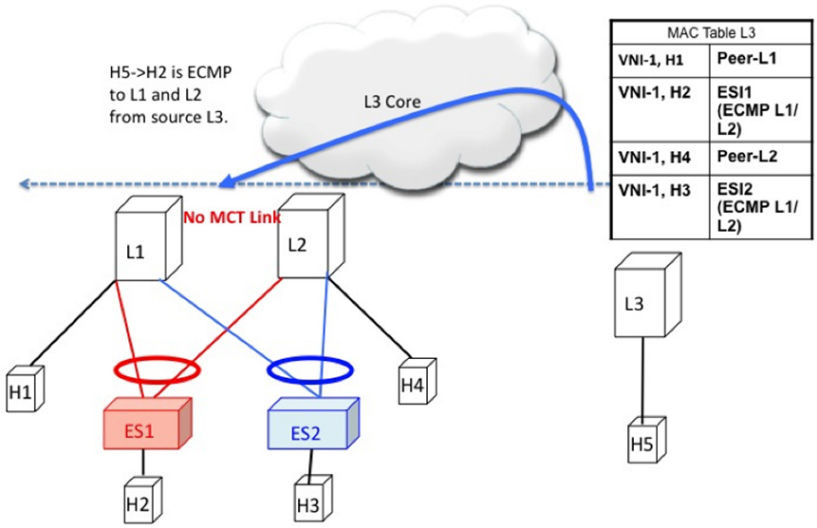
Figure 14: H1, H2, and H3 are in different VLANs. Core fails on L1. Access is brought down. H1 loses all connectivity.



EVPN Multihoming Remote Traffic Flows

Consider a remote switch L3 that sends bridged and routed traffic to the multihomed complex comprising of switches L1 and L2. As there is no virtual or emulated IP representing this MH complex, L3 must do ECMP at the source for both bridged and routed traffic. This section describes how the ECMP is achieved at switch L3 for both bridged and routed cases and how the system interacts with core and access failures.

Figure 15: Layer 2 VXLAN Gateway. L3 performs MAC ECMP to L1/L2.



Remote Bridged Traffic

Consider a remote host H5 that wants to bridge traffic to host H2 that is positioned behind the EVPN MH Complex (L1, L2). Host H2 builds an ECMP list in accordance to the rules defined in RFC 7432. The MAC table at switch L3displays that the MAC entry for H2 points to an ECMP PathList comprising of IP-L1 and IP-L2. Any bridged traffic going from H5 to H2 is VXLAN encapsulated and load balanced to switches L1 and L2. When making the ECMP list, the following constructs need to be kept in mind:

- Mass Withdrawal: Failures causing PathList correction should be independent of the scale of MACs.
- Aliasing: PathList Insertions may be independent of the scale of MACs (based on support of optional routes).

Below are the main constructs needed to create this MAC ECMP PathList:

Ethernet Auto Discovery Route (Type 1) per ES

EVPN defines a mechanism to efficiently and quickly signal the need to update their forwarding tables upon the occurrence of a failure in connectivity to an Ethernet Segment. Having each PE advertise a set of one or more Ethernet A-D per ES route for each locally attached Ethernet Segment does this.

Ethernet Auto Discovery Route (Route Type 1) per ES		
NLRI	Route Type	Ethernet Segment (Type 1)
	Route Distinguisher	Router-ID: Segment-ID (VNID << 8)
	ESI	<Type: 1B><MAC: 6B><LD: 3B>
	Ethernet Tag	MAX-ET
	MPLS Label	0
ATTRS	ESI Label Extended Community ESI Label = 0	Single Active = False
	Next-Hop	NVE Loopback IP
	Route Target	Subset of List of RTs of MAC-VRFs associated to all the EVIs active on the ES

MAC-IP Route (Type 2)

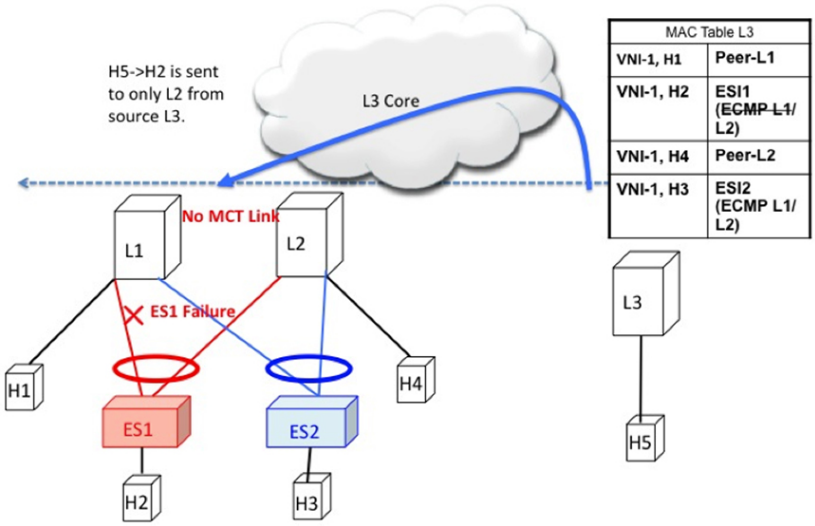
MAC-IP Route remains the same as used in the current vPC multihoming and standalone single-homing solutions. However, now it has a non-zero ESI field that indicates that this is a multihomed host and it is a candidate for ECMP Path Resolution.

MAC IP Route (Route Type 2)		
NLRI	Route Type	MAC IP Route (Type 2)
	Route Distinguisher	RD of MAC-VRF associated to the Host
	ESI	<Type : 1B><MAC : 6B><LD : 3B>
	Ethernet Tag	MAX-ET
	MAC Addr	MAC Address of the Host
	IP Addr	IP Address of the Host
	Labels	L2VNI associated to the MAC-VRF L3VNI associated to the L3-VRF
ATTRS	Next-Hop	Loopback of NVE
	RT Export	RT configured under MAC-VRF (AND/OR) L3-VRF associated to the host

Access Failure for Remote Bridged Traffic

In the condition of a failure of ESI links, it results in mass withdrawal. The EAD/ES route is withdrawn leading the remote device to remote the switch from the ECMP list for the given ES.

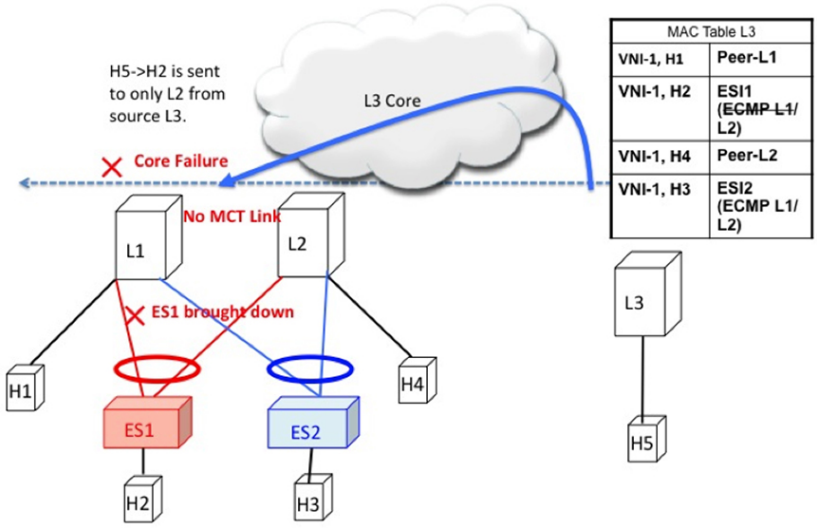
Figure 16: Layer 2 VXLAN Gateway. ESI failure on L1. L3 withdraws L1 from MAC ECMP list. This will happen due to EAD/ES mass withdrawal from L1.



Core Failure for Remote Bridged Traffic

If switch L1 gets isolated from the core, it must not continue to attract access traffic, as it is not able to encapsulate and send it on the overlay. It means that the access links must be brought down at L1 if L1 loses core reachability.

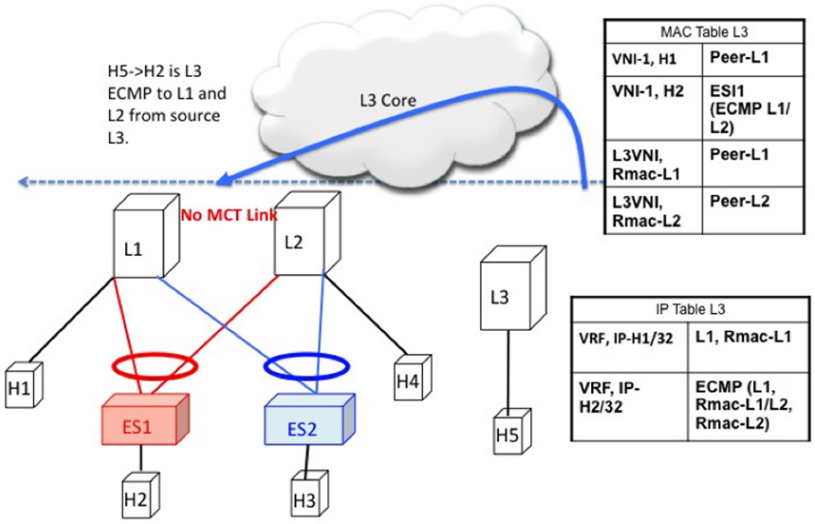
Figure 17: Layer 2 VXLAN Gateway. Core failure at L1. L3 withdraws L1 from MAC ECMP list. This will happen due to route reachability to L1 going away at L3.



Remote Routed Traffic

Consider L3 being a Layer 3 VXLAN Gateway and H5 and H2 belonging to different subnets. In that case, any inter-subnet traffic going from L3 to L1/L2 is routed at L3, that is a distributed anycast gateway. Both L1 and L2 advertise the MAC-IP route for Host H2. Due to the receipt of these routes, L3 builds an L3 ECMP list comprising of L1 and L2.

Figure 18: Layer 3 VXLAN Gateway. L3 does IP ECMP to L1/L2 for inter subnet traffic.



Access Failure for Remote Routed Traffic

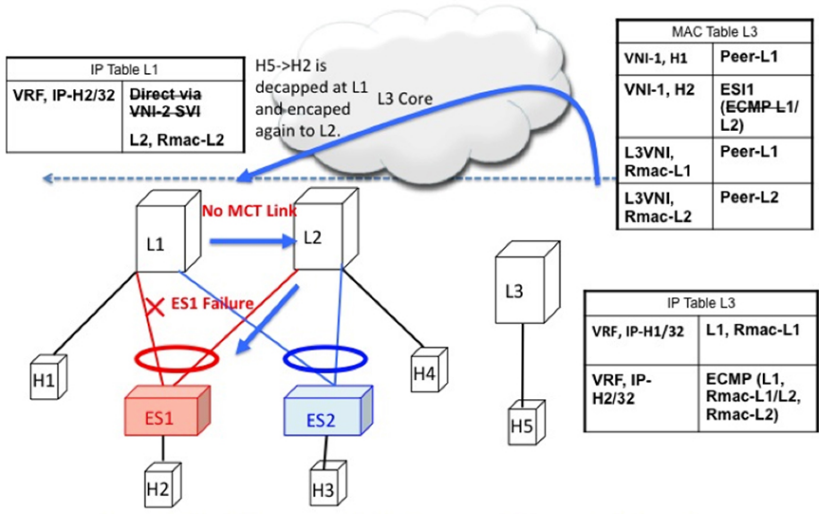
If the access link pointing to ES1 goes down on L1, the mass withdrawal route is sent in the form of EAD/ES and that causes L3 to remove L1 from the MAC ECMP PathList, leading the intra-subnet (L2) traffic to converge quickly. L1 now treats H2 as a remote route reachable via VxLAN Overlay as it is no longer directly connected through the ESI link. This causes the traffic destined to H2 to take the suboptimal path L3->L1->L2.

Inter-Subnet traffic H5->H2 will follow the following path:

- Packet are sent by H5 to gateway at L3.
- L3 performs symmetric IRB and routes the packet to L1 via VXLAN overlay.
- L1 decaps the packet and performs inner IP lookup for H2.
- H2 is a remote route. Therefore, L1 routes the packet to L2 via VXLAN overlay.
- L2 decaps the packet and performs an IP lookup and routes it to directly attached SVI.

Hence the routing happens 3 times, once each at L3, L1, and L2. This sub-optimal behavior continues until Type-2 route is withdrawn by L1 by BGP.

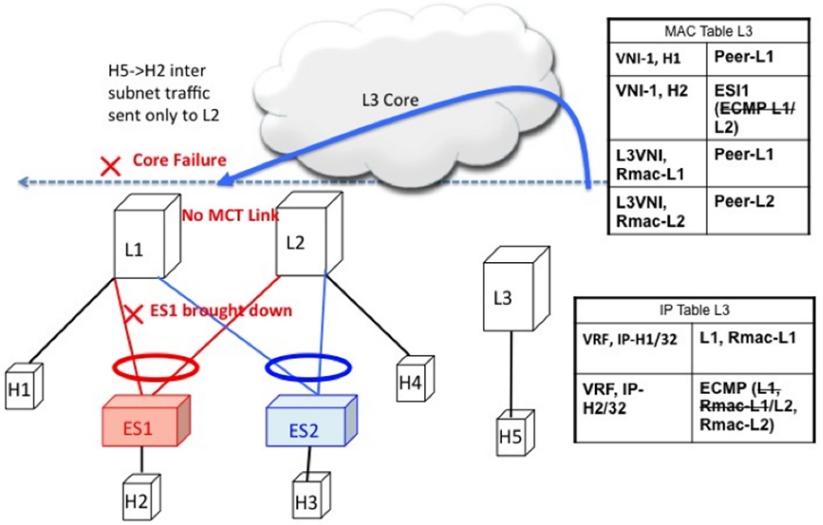
Figure 19: Layer 3 VXLAN Gateway. ESI failure causes ES mass withdrawal that only impacts L2 ECMP. L3 ECMP continues until Type2 is withdrawn. L3 traffic reaches H2 via suboptimal path L3->L1->L2 until then.



Core Failure for Remote Routed Traffic

Core Failure for Remote Routed Traffic behaves the same as core failure for remote bridged traffic. As the underlay routing protocol withdraws L1's loopback reachability from all remote switches, L1 is removed from both MAC ECMP and IP ECMP lists everywhere.

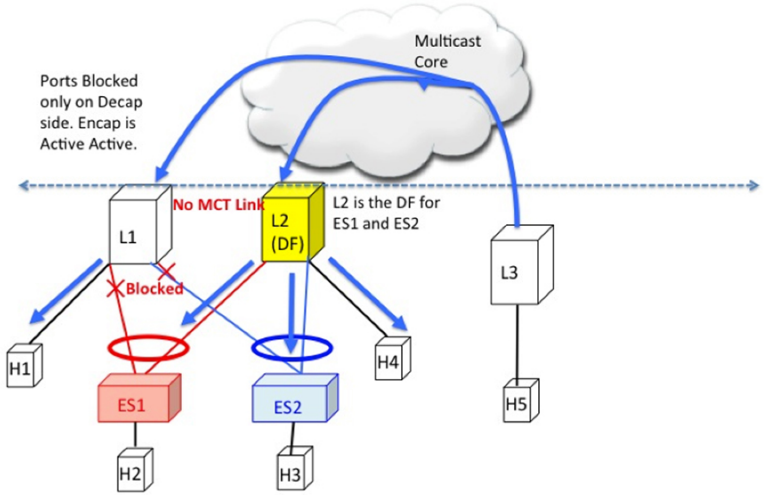
Figure 20: Layer 3 VXLAN Gateway. Core failure. All L3 ECMP paths to L1 are withdrawn at L3 due to route reachability going away.



EVPN Multihoming BUM Flows

Cisco NX-OS supports multicast core in the underlay with ESI. Consider BUM traffic originating from H5. The BUM packets are encapsulated in the multicast group mapped to the VNI. Because both L1 and L2 have joined the shared tree (*, G) for the underlay group based on the L2VNI mapping, both receive a copy of the BUM traffic.

Figure 21: BUM traffic originating at L3. L2 is the DF for ES1 and ES2. L2 decapsulates and forwards to ES1, ES2 and orphan. L1 decapsulates and only forwards to orphan.



Designated Forwarder

It is important that only one of the switches in the redundancy group decaps and forwards BUM traffic over the ESI links. For this purpose, a unique Designated Forwarder (DF) is elected on a per Ethernet Segment basis. The role of the DF is to decap and forward BUM traffic originating from the remote segments to the destination local segment for which the device is the DF. The main aspects of DF election are:

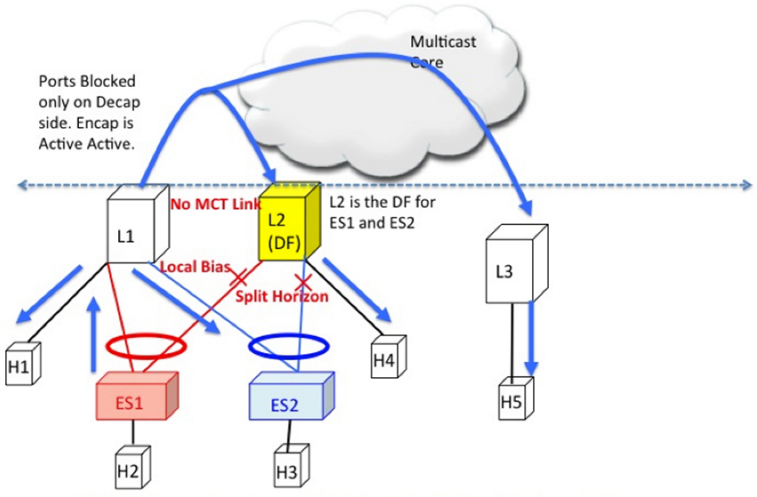
- DF Election is per (ES, VLAN) basis. There can be a different DF for ES1 and ES2 for a given VLAN.
- DF election result only applies to BUM traffic on the RX side for decap.
- Every switch must decap BUM traffic to forward it to singly homed or orphan links.
- Duplication of DF role leads to duplicate packets or loops in a DHN. Therefore, there must be a unique DF on per (ES, VLAN) basis.

Split Horizon and Local Bias

Consider BUM traffic originating from H2. Consider that this traffic is hashed at L1. L1 encapsulates this traffic in Overlay Multicast Group and sends the packet out to the core. All switches that have joined this multicast group with same L2VNI receive this packet. Additionally, L1 also locally replicates the BUM packet on all directly connected orphan and ESI ports. For example, if the BUM packet originated from ES1, L1 locally replicates it to ES2 and the orphan ports. This technique to replicate to all the locally attached links is termed as local-bias.

Remote switches decap and forward it to their ESI and orphan links based on the DF state. However, this packet is also received at L2 that belongs to the same redundancy group as the originating switch L1. L2 must decap the packet to send it to orphan ports. However, even through L2 is the DF for ES1, L2 must not forward this packet to ES1 link. This packet was received from a peer that shares ES1 with L1 as L1 would have done local-bias and duplicate copies should not be received on ES2. Therefore L2 (DF) applies a split-horizon filter for L1-IP on ES1 and ES2 that it shares with L1. This filter is applied in the context of a VLAN.

Figure 22: BUM traffic originating at L1. L2 is the DF for ES1 and ES2. However, L2 must perform split horizon check here as it shares ES1 and ES2 with L1. L2 however



Ethernet Segment Route (Type 4)

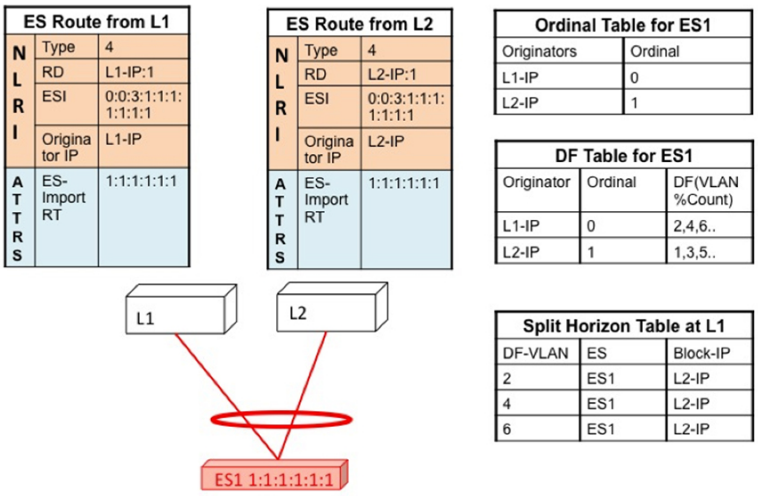
The Ethernet Segment Route is used to elect the Designated Forwarder and to apply Split Horizon Filtering. All the switches that are configured with an Ethernet Segment originate from this route. Ethernet Segment Route is exported and imported when ESI is locally configured under the PC.

Ethernet Segment Route (Route Type 4)		
NLRI	Route Type	Ethernet Segment (Type 4)
	RD	Router-ID: Base + Port Channel Number
	ESI	<Type : 1B><MAC : 6B><LD : 3B>
	Originator IP	NVE loopback IP
ATTRS	ES-Import RT	6 Byte MAC derived from ESI

DF Election and VLAN Carving

Upon configuration of the ESI, both L1 and L2 advertises the ES route. The ESI MAC is common between L1 and L2 and unique in the network. Therefore, only L1 and L2 import each other's ES routes.

Figure 23: If VLAN % count equals to ordinal, take up DF role.



Core and Site Failures for BUM Traffic

If the access link pertaining to ES1 fails at L1, L1 withdraws the ES route for ES1. This leads to a change triggering re-compute the DF. Since L2 is the only TOR left in the Ordinal Table, it takes over DF role for all VLANs.

BGP EVPN multihoming on Cisco Nexus 3100 Series switches provides minimum operational and cabling expenditure, provisioning simplicity, flow based load balancing, multi pathing, and fail-safe redundancy.

Configuring VLAN Consistency Checking

Overview of VLAN Consistency Checking

In a typical multihoming deployment scenario, host 1 belonging to VLAN X sends traffic to the access switch and then the access switch sends the traffic to both the uplinks towards VTEP1 and VTEP2. The access switch does not have the information about VLAN X configuration on VTEP1 and VTEP2. VLAN X configuration mismatch on VTEP1 or VTEP2 results in a partial traffic loss for host 1. VLAN consistency checking helps to detect such configuration mismatch.

For VLAN consistency checking, CFSoIP is used. Cisco Fabric Services (CFS) provides a common infrastructure to exchange the data across the switches in the same network. CFS has the ability to discover CFS capable switches in the network and to discover the feature capabilities in all the CFS capable switches. You can use CFS over IP (CFSoIP) to distribute and synchronize a configuration on one Cisco device or with all other Cisco devices in your network.

CFSoIP uses multicast to discover all the peers in the management IP network. For EVPN multihoming VLAN consistency checking, it is recommended to override the default CFS multicast address with the **cfs ipv4 mcast-address** <*mcast address*> CLI command. To enable CFSoIP, the **cfs ipv4 distribute** CLI command should be used.

When a trigger (for example, device booting up, VLAN configuration change, VLANs administrative state change on the ethernet-segment port-channel) is issued on one of the multihoming peers, a broadcast request

with a snapshot of configured and administratively up VLANs for the ethernet-segment (ES) is sent to all the CFS peers.

When a broadcast request is received, all CFS peers sharing the same ES as the requestor respond with their VLAN list (configured and administratively up VLAN list per ES). The VLAN consistency checking is run upon receiving a broadcast request or a response.

A 15 seconds timer is kicked off before sending a broadcast request. On receiving the broadcast request or response, the local VLAN list is compared with that of the ES peer. The VLANs that do not match are suspended. Newly matched VLANs are no longer suspended.

VLAN consistency checking runs for the following events:

- Global VLAN configuration: Add, delete, shut, or no shut events.

  Port channel VLAN configuration: Trunk allowed VLANs added or removed or access VLAN changed.

- CFS events: CFS peer added or deleted or CFSoIP configuration is removed.

- ES Peer Events: ES peer added or deleted.

The broadcast request is retransmitted if a response is not received. VLAN consistency checking fails to run if a response is not received after 3 retransmissions.

VLAN Consistency Checking Guidelines and Limitations

See the following guidelines and limitations for VLAN consistency checking:

- The VLAN consistency checking uses CFSoIP. Out-of-band access through a management interface is mandatory on all multihoming switches in the network.
- It is recommended to override the default CFS multicast address with the CLI **cfs ipv4 mcast-address** <*mcast address*> command.
- The VLAN consistency check cannot detect a mismatch in **switchport trunk native vlan** configuration.
- CFSoIP and CFSoE should not be used in the same device.
- CFSoIP should not be used in devices that are not used for VLAN consistency checking.
- If CFSoIP is required in devices that do not participate in VLAN consistency checking, a different multicast group should be configured for devices that participate in VLAN consistency with the CLI **cfs ipv4 mcast-address** <*mcast address*> command.

Displaying Show command Output for VLAN Consistency Checking

See the following show commands output for VLAN consistency checking.

To list the CFS peers, use the **sh cfs peers name nve** CLI command.

```
switch# sh cfs peers name nve

Scope      : Physical-ip
--------------------------------------------------------------------------
 Switch WWN              IP Address
--------------------------------------------------------------------------
 20:00:f8:c2:88:23:19:47 172.31.202.228                      [Local]
```

```
                          Switch
 20:00:f8:c2:88:90:c6:21 172.31.201.172                          [Not Merged]
 20:00:f8:c2:88:23:22:8f 172.31.203.38                           [Not Merged]
 20:00:f8:c2:88:23:1d:e1 172.31.150.132                          [Not Merged]
 20:00:f8:c2:88:23:1b:37 172.31.202.233                          [Not Merged]
 20:00:f8:c2:88:23:05:1d 172.31.150.134                          [Not Merged]
```

The **show nve ethernet-segment** command now displays the following details:

- The list of VLANs for which consistency check is failed.
- Remaining value (in seconds) of the global VLAN CC timer.

```
switch# sh nve ethernet-segment
ESI Database
----------------------------------------
ESI: 03aa.aaaa.aaaa.aa00.0001,
   Parent interface: port-channel2,
  ES State: Up
  Port-channel state: Up
  NVE Interface: nve1
   NVE State: Up
   Host Learning Mode: control-plane
  Active Vlans: 3001-3002
   DF Vlans: 3002
   Active VNIs: 30001-30002
  CC failed VLANs: 0-3000,3003-4095
  CC timer status: 10 seconds left
  Number of ES  members: 2
  My ordinal: 0
  DF timer start time: 00:00:00
  Config State: config-applied
  DF List: 201.1.1.1 202.1.1.1
  ES route added to L2RIB: True
  EAD routes added to L2RIB: True
```

See the following Syslog output:

```
switch(config)# 2017 Jan ?7 19:44:35 Switch %ETHPORT-3-IF_ERROR_VLANS_SUSPENDED: VLANs
2999-3000 on Interface port-channel40 are being suspended.
(Reason: SUCCESS)

After Fixing configuration
2017 Jan ?7 19:50:55 Switch %ETHPORT-3-IF_ERROR_VLANS_REMOVED: VLANs 2999-3000 on Interface
 port-channel40 are removed from suspended state.
```

Configuring ESI ARP Suppression

Overview of ESI ARP Suppression

ESI ARP suppression is an extension of already available ARP suppression solution in VXLAN-EVPN. This feature is supported on top of ESI multihoming solution, that is on top of VXLAN-EVPN solution. ARP

suppression is an optimization on top of BGP-EVPN multihoming solution. ARP broadcast is one of the most significant part of broadcast traffic in data centers. ARP suppression significantly cuts down on ARP broadcast in the data center.

ARP request from host is normally flooded in the VLAN. You can optimize flooding by maintaining an ARP cache locally on the access switch. ARP cache is maintained by the ARP module. ARP cache is populated by snooping all the ARP packets from the access or server side. Initial ARP requests are broadcasted to all the sites. Subsequent ARP requests are suppressed at the first hop leaf and they are answered locally. In this way, the ARP traffic across overlay can be significantly reduced.

ARP suppression is only supported with BGP-EVPN (distributed gateway).

ESI ARP suppression is a per-VNI (L2-VNI) feature. ESI ARP suppression is supported in both L2 (no SVI) and L3 modes. Beginning with Cisco NX-OS Release 7.0(3)I7(1), only L3 mode is supported.

The ESI ARP suppression cache is built by:

- Snooping all ARP packets and populating ARP cache with the source IP and MAC bindings from the request.
- Learning IP-host or MAC-address information through BGP EVPN MAC-IP route advertisement.

Upon receiving the ARP request, the local cache is checked to see if the response can be locally generated. If the cache lookup fails, the ARP request can be flooded. This helps with the detection of the silent hosts.

Limitations for ESI ARP Suppression

See the following limitations for ESI ARP suppression:

- ESI multihoming solution is supported only on Cisco Nexus 3100 platform switches.
- ESI ARP suppression is only supported in L3 (SVI) mode.
- ESI ARP suppression cache limit is 64K that includes both local and remote entries.

Configuring ESI ARP Suppression

For ARP suppression VACLs to work, configure the TCAM carving using the **hardware access-list tcam region arp-ether 256** CLI command.

```
Interface nve1
   no shutdown
   source-interface loopback1
   host-reachability protocol bgp
   member vni 10000
       suppress-arp
   mcast-group 224.1.1.10
```

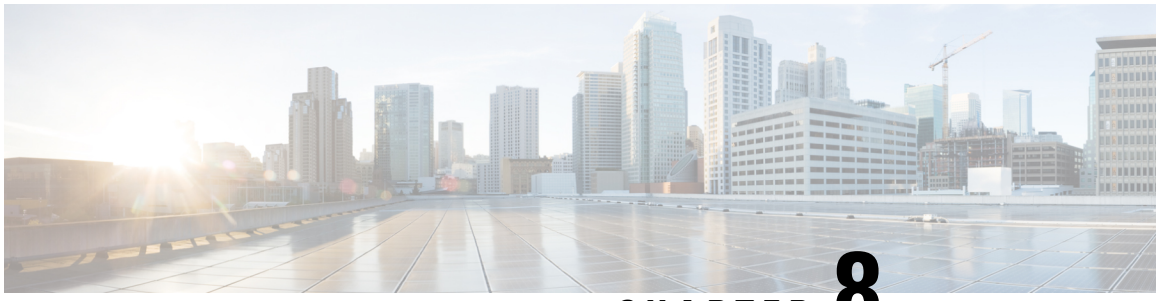
Displaying Show Commands for ESI ARP Suppression

See the following Show commands output for ESI ARP suppression:

```
switch# show ip arp suppression-cache ?
 detail        Show details
```

```
   local          Show local entries
   remote     Show remote entries
   statistics  Show statistics
   summary Show summary
   vlan           L2vlan

switch# show ip arp suppression-cache local

Flags: + - Adjacencies synced via CFSoE
       L - Local Adjacency
       R - Remote Adjacency
       L2 - Learnt over L2 interface
      PS - Added via L2RIB, Peer Sync
      RO - Dervied from L2RIB Peer Sync Entry

Ip Address      Age      Mac Address    Vlan Physical-ifindex    Flags              Remote
 Vtep Addrs

61.1.1.20       00:07:54 0000.0610.0020  610 port-channel20     L
61.1.1.30       00:07:54 0000.0610.0030  610 port-channel2       L[PS RO]
61.1.1.10       00:07:54 0000.0610.0010  610 Ethernet1/96        L


switch# show ip arp suppression-cache remote
 Flags: + - Adjacencies synced via CFSoE
        L - Local Adjacency
        R - Remote Adjacency
        L2 - Learnt over L2 interface
          PS - Added via L2RIB, Peer Sync
          RO - Dervied from L2RIB Peer Sync Entry
          Ip Address      Age      Mac Address         Vlan    Physical-ifindex    Flags
    Remote Vtep Addrs
 61.1.1.40       00:48:37 0000.0610.0040   610     (null)                      R
 VTEP1, VTEP2.. VTEPn


switch# show ip arp suppression-cache detail
       Flags: + - Adjacencies synced via CFSoE
       L - Local Adjacency
       R - Remote Adjacency
       L2 - Learnt over L2 interface
       PS - Added via L2RIB, Peer Sync
       RO - Derived from L2RIB Peer Sync Entry

             Ip Address      Age      Mac Address    Vlan Physical-ifindex    Flags
     Remote Vtep Addrs
 61.1.1.20       00:00:07 0000.0610.0020  610 port-channel20     L
 61.1.1.30       00:00:07 0000.0610.0030  610 port-channel2       L[PS RO]
 61.1.1.10       00:00:07 0000.0610.0010  610 Ethernet1/96        L
 61.1.1.40       00:00:07 0000.0610.0040  610 (null)                    R
 VTEP1, VTEP2.. VTEPn


switch# show ip arp suppression-cache summary
 IP ARP suppression-cache Summary
 Remote          :1
 Local              :3
 Total               :4
switch# show ip arp suppression-cache statistics
 ARP packet statistics for suppression-cache
 Suppressed:
 Total 0, Requests 0, Requests on L2 0, Gratuitous 0, Gratuitous on L2 0
 Forwarded :
 Total: 364
```

```
  L3 mode :      Requests 364, Replies 0
  Request on core port 364, Reply on core port 0
                 Dropped 0
  L2 mode :      Requests 0, Replies 0
                 Request on core port 0, Reply on core port 0
                 Dropped 0
 Received:
 Total: 3016
 L3 mode:       Requests 376, Replies 2640
  Local Request 12, Local Responses 2640
                 Gratuitous 0, Dropped 0
  L2 mode :      Requests 0, Replies 0
                 Gratuitous 0, Dropped 0


switch# sh ip arp multihoming-statistics vrf all
ARP Multihoming statistics for all contexts
Route Stats
============
 Receieved ADD from L2RIB        :1756 | 1756:Processed ADD from L2RIB Receieved DEL from
L2RIB        :88 | 87:Processed DEL from L2RIB Receieved PC shut from L2RIB    :0 |
1755:Processed PC shut from L2RIB Receieved remote UPD from L2RIB :5004 | 0:Processed remote
 UPD from L2RIB
ERRORS
=======
Multihoming ADD error invalid flag         :0
Multihoming DEL error invalid flag         :0
Multihoming ADD error invalid current state:0
Multihoming DEL error invalid current state:0
Peer sync DEL error MAC mismatch           :0
Peer sync DEL error second delete          :0
Peer sync DEL error deleteing TL route     :0
True local DEL error deleteing PS RO route :0

switch#
```



CHAPTER 8

Configuring IPv6 Across a VXLAN EVPN Fabric

This chapter contains the following sections:

Overview of IPv6 Across a VXLAN EVPN Fabric

This section provides an example configuration that enables IPv6 in the overlay of a VXLAN EVPN fabric.

Cisco Nexus 3500 Series switches do not support IPv6 Across VXLAN EVPN on Cisco NX-OS Release 7.0(3)I7(2) and the previous releases.

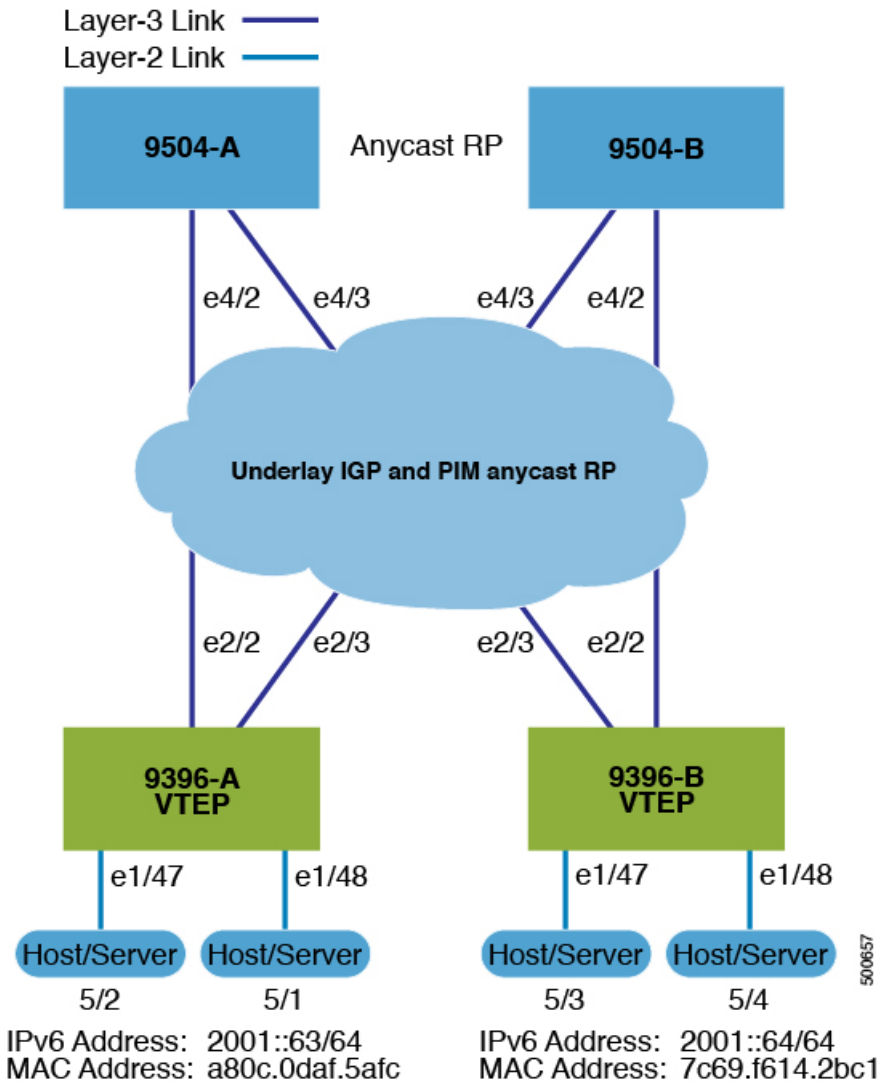
The VXLAN encapsulation mechanism encapsulates the IPv6 packets in the overlay as IPv4 UDP packets and uses IPv4 routing to transport the VXLAN encapsulated traffic.

To enable IPv6 across a VXLAN EVPN fabric, the IPv6 address family is included in VRF, BGP, and EVPN. IPv6 routes are initiated in the tenant VRF IPv6 unicast address-family on a VTEP and are advertised in the VXLAN fabric through the L2VPN EVPN address family as EVPN route-type 2 or 5.

Note These routes are advertised as EVPN routes on the spine.

Configuring IPv6 Across a VXLAN EVPN Fabric Example

Topology for the example:



Note In the example:

- Configuration for hosts in VLAN 10 is mapped to vn-segment 10010.
- VRF RED is the VRF associated with this VLAN.
- 20010 is the L3 VNI for VRF RED.
- VLAN 100 is mapped to L3 VNI 20010.

- Configure the Layer 2 VLAN.

```
vlan 10
  name RED
  vn-segment 10010
```

- Configure the VLAN for L3 VNI .

```
vlan 100
 name RED_L3_VNI_VLAN
 vn-segment 20010
```

- Define the anycast gateway MAC.

```
fabric forwarding anycast-gateway-mac 0000.2222.3333
```

- 

Note You can choose either of the following two command procedures for creating the NVE interfaces. Use the first one for a small number of VNIs. Use the second procedure to configure a large number of VNIs.

Define the NVE interface.

Option 1

```
interface nve1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  member vni 20010 associate-vrf
  member vni 10010
    suppress-arp
    mcast-group 225.4.0.1
```

Option 2

```
interface nve1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  global mcast-group 255.4.0.1
  member vni 20010 associate-vrf
  member vni 10010
    suppress-arp


evpn
  vni 10010 l2
```

Note The following commands are optional, but may be entered as overrides.

```
rd auto
    route-target import auto
    route-target export auto
```

- Add configuration the to SVI definition on VLAN 10 and on L3 VNI VLAN 100.

```
interface Vlan10
  description RED
```

```
  no shutdown
  vrf member RED
  no ip redirects
  ip address 10.1.1.1/24
  ipv6 address 2001::1/64
  fabric forwarding mode anycast-gateway
```

- Configure SVI definition for VLAN 100.

```
interface Vlan100
 description RED_L3_VNI_VLAN
 no shutdown
 vrf member RED
 ip forward
 ipv6 address use-link-local-only
```

Note The IPv6 address use-link-local-only serves the same purpose as IP FORWARD for IPv4. It enables the switch to perform an IP based lookup even when the interface VLAN has no IP address defined under it.

- Add configuration to the VRF definition.

```
vrf context RED
  vni 20010
  rd auto
```

Note The following commands are automatically configured unless one or more are entered as overrides.

```
rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
  address-family ipv6 unicast
    route-target both auto
    route-target both auto evpn


evpn
  vni 10010 l2
```

Note The following commands are automatically configured unless one or more are entered as overrides.

```
rd auto
    route-target import auto
    route-target export auto
```

- Add configuration to the VRF definition under BGP.

```
router bgp 65000
  vrf RED
    address-family ipv4 unicast
      advertise l2vpn evpn
   address-family ipv6 unicast
      advertise l2vpn evpn
```

Note If VTEPs are configured to operate as VPC peers, the following configuration is a best practice that should be included under the VPC domain on both switches.

```
vpc domain 1
   ipv6 nd synchronize
```

Show Command Examples

The following are examples of verifying IPv6 advertisement over VXLAN EVPN:

- Display ND information for the connected server.

```
9396-B_VTEP# show ipv6 neighbor vrf RED

Flags: # - Adjacencies Throttled for Glean
       G - Adjacencies of vPC peer with G/W bit
       R - Adjacencies learnt remotely

IPv6 Adjacency Table for VRF RED
Total number of entries: 2
Address         Age       MAC Address     Pref Source     Interface
2001::64        00:00:26  7c69.f614.2bc1  50   icmpv6     Vlan10
fe80::7e69:f6ff:fe14:2bc1
                00:01:13  7c69.f614.2bc1  50   icmpv6     Vlan10
```

- Check the L2ROUTE and ensure the MAC-IP was learned.

```
9396-B_VTEP# show l2route evpn mac-ip evi 10 host-ip 2001::64
Mac Address    Prod Host IP                                 Next Hop (s)
-------------- ---- --------------------------------------- --------------
7c69.f614.2bc1 HMM  2001::64                                N/A
```

Note MAC-IP table is populated only when the end server sends a neighbor solicitation message (ARP in case of IPv4).

- Verify the route is present locally in the BGP table.

```
9396-B_VTEP# show bgp l2vpn evpn 2001::64
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 198.19.0.15:34180    (L2VNI 10010)
BGP routing table entry for [2]:[0]:[0]:[48]:[7c69.f614.2bc1]:[128]:[2001::64]/368,
version 678
```

```
Paths: (1 available, best #1)
Flags: (0x00010a) on xmit-list, is not in l2rib/evpn

  Advertised path-id 1
  Path type: local, path is valid, is best path, no labeled nexthop
  AS-Path: NONE, path locally originated
    198.19.0.15 (metric 0) from 0.0.0.0 (198.19.0.15)
      Origin IGP, MED not set, localpref 100, weight 32768
      Received label 10010 20010
      Extcommunity:  RT:64567:10010 RT:64567:20010

    Path-id 1 advertised to peers:
    198.19.0.3
    198.19.0.4
```

- Verify the route is present in the remote VTEP 9396-A-VTEP BGP table.

```
9396-A-VTEP# show bgp l2vpn evpn 2001::64
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 198.19.0.14:34180    (L2VNI 10010)
BGP routing table entry for [2]:[0]:[0]:[48]:[7c69.f614.2bc1]:[128]:[2001::64]/368,
version 305
Paths: (1 available, best #1)
Flags: (0x00021a) on xmit-list, is in l2rib/evpn, is not in HW,

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
             Imported from
198.19.0.15:34180:[2]:[0]:[0]:[48]:[7c69.f614.2bc1]:[128]:[2001::64]/240
  AS-Path: NONE, path sourced internal to AS
    198.19.0.15 (metric 81) from 198.19.0.3 (198.19.0.3)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10010 20010
      Extcommunity:  RT:64567:10010 RT:64567:20010 ENCAP:8 Router MAC:5087.89a1.a52f
      Originator: 198.19.0.15 Cluster list: 198.19.0.3
```

- Check the L2ROUTE and ensure that the MAC-IP was learned on the remote VTEP - 9396-A-VTEP.

```
rswV1leaf14# show l2route evpn mac-ip evi 1413 host-ip 2001::64
Mac Address    Prod Host IP                                 Next Hop (s)
-------------- ---- --------------------------------------- --------------
7c69.f614.2bc1 BGP  2001::64                                198.19.0.15
```

INDEX

V